LREC 2026

The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026

Workshop Proceedings

Editors
Jin Zhao, Claire Benet Post, Elizabeth Hoefer

11 May 2026

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026

ISBN 978-2-493814-54-8

Preface

Welcome to the Proceedings of the Seventh International Workshop on Designing Meaning Representations (DMR 2026), held on May 11, 2026, in Palma de Mallorca, Spain, in conjunction with the 15th edition of the Language Resources and Evaluation Conference (LREC 2026).

The DMR workshop series brings together researchers working on the design, annotation, and use of meaning representations across a variety of linguistic frameworks and applications. This year's workshop features contributions spanning Abstract Meaning Representation (AMR), Uniform Meaning Representation (UMR), frame semantics, compositional semantics, and other formalisms for capturing linguistic meaning.

We received 22 submissions and accepted 14 papers for presentation: 6 as oral presentations and 8 as posters. In addition, the workshop includes the First Shared Task on UMR Parsing, with an overview paper and two system papers from the shared task track. The program also features a panel discussion on "The Role of Symbolic Representations in the Era of LLMs", moderated by James Pustejovsky, with panelists Louise McNally, German Rigau, Jan Hajič, Susan Windisch Brown, and Claire Bonial.

We would like to thank the members of the program committee for their thorough and timely reviews, the authors for their contributions, and the LREC 2026 organizers for their support. We also gratefully acknowledge the shared task organizers, Daniel Zeman and Jan Štěpánek, for their efforts in designing and coordinating the first UMR parsing shared task.

Jin Zhao, Claire Benet Post, and Elizabeth Hoefer
DMR 2026 Workshop Chairs

Organizing Committee

Workshop Chairs

- Jin Zhao, Brandeis University
- Claire Benet Post, University of Colorado Boulder
- Elizabeth Hoefer, University of St. Thomas

Organizing Committee

- Jan Hajič, Charles University
- Daniel Zeman, Charles University
- Jan Štěpánek, Charles University
- Alexis Palmer, University of Colorado Boulder
- James Pustejovsky, Brandeis University
- Nianwen Xue, Brandeis University

Shared Task Organizers

- Daniel Zeman, Charles University
- Jan Štěpánek, Charles University

Program Committee

- Omri Abend, The Hebrew University of Jerusalem
- Maxime Amblard, Université de Lorraine
- Ayoub Bagheri, Utrecht University
- Jorge Baptista, University of Algarve & INESC-ID Lisboa
- Abhidip Bhattacharyya, University of Massachusetts Amherst
- Claire Bonial, ARL
- Johan Bos, University of Groningen
- Richard Brutti, Brandeis University
- Alastair Butler, Hirosaki University
- Kilian Evang, Heinrich Heine University Düsseldorf
- Jan Hajič, Charles University
- Johannes Heinecke, Orange Innovation
- Bin Li, Nanjing Normal University
- Juri Opitz, University of Zurich
- Alexis Palmer, University of Colorado Boulder
- Martha Palmer, University of Colorado Boulder
- Nathan Schneider, Georgetown University
- Djamé Seddah, Inria Paris
- Mollie Shichman, University of Maryland College Park
- Haibo Sun, Brandeis University
- Jingxuan Tu, Brandeis University
- Zdeněka Urešová, Charles University
- Kristin Wright-Bettner, University of Colorado Boulder
- Nianwen Xue, Brandeis University
- Zhixing Xu, Nanjing Normal University
- Annie Zaenen, Stanford
- Deniz Zeyrek, Middle East Technical University

Invited Panelists

- Louise McNally, Universitat Pompeu Fabra
- German Rigau, University of the Basque Country (UPV/EHU)
- Jan Hajič, Charles University
- Susan Windisch Brown, University of Colorado Boulder
- Claire Bonial, U.S. Army Research Laboratory

Table of Contents

Workshop Program

Monday, May 11, 2026

09:00–09:10 ***Opening Remarks***

09:10–10:25 **Oral Session 1**
Room: Room #12
Chair: Jim Martin

09:10–09:35 *CxGr-AMR: Extending Abstract Meaning Representation Beyond Lexically Anchored Relations with Constructional Rolesets*
Claire Bonial, Claire Benet Post, Paul Van Eecke, Katrien Beuls and Harish Tayyar Madabushi

09:35–10:00 *Adding Aspectual Information to Structured Meaning Representations*
Claire Benet Post, Paul Bontempo, August Ulfelder Milliken, Alvin Po-Chun Chen, Nicholas Derby, Saksham Khatwani, Sumeyye Nabieva, Karthik Sairam and Alexis Palmer

10:00–10:25 *Modelling Idiomatic Expressions in Abstract Meaning Representation*
Venera Gareeva and Johannes Heinecke

Monday, May 11, 2026

10:30–11:30 **Coffee Break & Poster Session**
Room: Room #12
Chair: Claire Benet Post

TrAinMR: an Annotator Training Website for Abstract Meaning Representation
Mina Yang and Shira Wein

Named Entity Recognition for Persian Literary Text: A Case Study on The Little Prince
Minoo Nassajian, Joakim Nivre and Daniel Zeman

Towards Consistent UMR Annotation of Deverbal Nouns: Evidence from Czech and Latin
Hana Hledíková, Federica Gamba, Marketa Lopatkova and Jan Štěpánek

SAVI: Web-based Multilayered Semantic Annotation Validation Interface
Sashank Tatavolu, Soma Soma, Pratibha Rani and Sukhada Sukhada

Finding Meaning in Embeddings: Concept Separation Curves
Paul Keuren, Marc Ponsen and Robert Ayoub Bagheri

Monday, May 11, 2026 (continued)

Extracting First Order Logic formulas from graphical semantic representations
Rémi DE VERGNETTE, Vincent Tourneur and Maxime Amblard

Meaning Representations as Variational Quantum Circuits
Tilen Gaetano Limbäck-Stokin, Tanishka A. Birdavade, Kin Ian Lo and Mehrnoosh Sadrzadeh

Superframes: A Schema for Lexicon-free Frame-semantic Annotation
Kilian Evang

Monday, May 11, 2026

11:30–12:45 **Shared Task Session: UMR Parsing**
Room: Room #12
Chair: Dan Zeman

11:30–11:55 *First Shared Task on UMR Parsing*
Jan Štěpánek, Daniel Zeman, Marketa Lopatkova, Federica Gamba, Hana Hledíková and Nianwen Xue

11:55–12:15 *Orange @ UMR Parsing Shared Task*
Johannes Heinecke and Munshi Asadullah

12:15–12:35 *Sema System for the DMR 2026 Shared Task: Multistage UMR Parsing with Qwen3-4B*
Rémi DE VERGNETTE and Maxime Amblard

12:35–12:45 ***Q&A / Discussion***

Monday, May 11, 2026

13:00–14:00 ***Lunch Break***

14:00–15:30 *Panel: The Role of Symbolic Representations in the Era of LLMs*
Moderator: James Pustejovsky; Panelists: Louise McNally, German Rigau, Jan Hajic, Susan Windisch Brown, Claire Bonial

15:30–15:55 **Oral Session 2**
Room: Room #12
Chair: Jan Stepanek

15:30–15:55 *Meaning Annotation Experience. A Tribute to Petr Sgall*
Marie Mikulová, Jan Štěpánek, Barbora Štěpánková, Jarmila Panevova and Eva Hajicova

16:00–16:30 ***Coffee Break***

Monday, May 11, 2026

16:30–17:20 **Oral Session 3**
Room: Room #12
Chair: Claire Benet Post

16:30–16:55 *Extending Uniform Meaning Representation to Persian: The First Corpus Resource*
Minoo Nassajian and Daniel Zeman

16:55–17:20 *Regression-Tested Compositional Semantics: A Graphical Development Environment for Glue and Description-by-Analysis*
Mark-Matthias Zymla and Kascha Kruschwitz

17:20–17:45 ***Open Discussion***

17:45–18:00 ***Closing Remarks***

CxGr-AMR: Extending Abstract Meaning Representation Beyond Lexically Anchored Relations with Constructional Rolesets

Claire Bonial[1], **Claire Benet Post**[2], **Paul Van Eecke**[3], **Katrien Beuls**[4], **Harish Tayyar Madabushi**[5]

[1]DEVCOM U.S. Army Research Laboratory, [2]University of Colorado,
[3]Vrije Universiteit Brussel, [4]Université de Namur, [5]University of Bath
claire.n.bonial.civ@army.mil

Abstract

Current Abstract Meaning Representation (AMR) annotation guidelines, which largely tie argument structure to lexical rolesets, systematically misrepresent cases in which key semantic roles stem from clause-level structure rather than the verb, leaving these meanings either unnaturally attached, incorrect, or unexpressed. To address this limitation, we present CxGr-AMR, a novel extension of AMR that captures the semantics of various types of phrasal constructions, including argument structure constructions. We first examine how such cases are handled under current Standard-AMR guidelines and show that these analyses are often inadequate when constructionally contributed roles clash with those assigned by the verb. We then provide a theoretical grounding for our CxGr-AMR rolesets that lay out the relationship between the syntactic signatures of constructional slots and particular semantic roles associated with them. Finally, we develop an annotation-expert-in-the-loop pipeline for the semi-automatic annotation of sentences, and release a dataset containing 355 instances of phrasal constructions annotated with both Standard and CxGr-AMR.

Keywords: CxGr-AMR, Abstract Meaning Representation (AMR), constructional semantics, argument structure constructions

1. Introduction

When hearing or reading sentences such as "*Firefighters cut a three-year-old free*," or "*The coach shouted their players into a queue*," English speakers immediately grasp that there is more to the meaning of these sentences than the mere cutting or shouting events they evoke. One can indeed understand from the first sentence that it was a cutting action performed by firefighters that *resulted in* the freedom of a three-year-old, while the second sentence evokes a scenario where players *moved* into a queue as a result of the coach's shouting. The observation that there exists meaning that is contributed by larger syntactic patterns, which is not reducible to the meanings of the lexical items that instantiate them, forms a basic tenet of the theory of Construction Grammar (CxG) (Fillmore, 1988; Goldberg, 1995; Kay and Fillmore, 1999; Fried and Östman, 2004).

Abstract Meaning Representation (AMR) is a widely adopted, graph-based meaning representation with an established annotation system and broad downstream use (Banarescu et al., 2013), so failures of coverage have direct consequences for both annotation quality and semantic parsing. Although AMR has been used to represent the meaning of constructions in computational construction grammar implementations (e.g., Schmalz and Cornillie, 2022; Beuls and Van Eecke, 2025), AMR primarily anchors argument structure in lexical verbal "rolesets", and therefore struggles to represent the clause-level meaning in sentences such as those presented above without mis-attaching roles or leaving key meaning implicit.

This paper presents an extension of AMR that facilitates the annotation of semantic roles that are evoked by linguistic structures above the level of lexical items.[1] Crucially, AMR's inventory of senses and rolesets is designed to be extensible, so the missing clause-level relations can be added in a way that remains compatible with existing AMR graphs and tools. Extending AMR therefore lets us address this coverage gap without abandoning the standard formalism or its resources (see §3). Concretely, the extension we present enables accurate semantic parsing of phrasal constructions such as the Resultative: "*The man shrieked himself unconscious*." Examples (1) and (2) below illustrate the current Standard-AMR parse and the new CxGr-AMR representation for this sentence:

(1)
```
(s / shriek-01
      :ARG0 (m / man
                :ARG1-of s)
      :mod (u / unconscious))
```

Parsed as: *The man shrieked, the thing-shrieked was the man, and the shrieking was unconscious*.

[1]We release the CxGr-AMR dataset, the constructional roleset specifications, and the annotation guidelines: https://github.com/H-TayyarMadabushi/cxgr-amr-construction-grammar

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 1–19
11 May 2026.

(2)
```
(r / resultative-91
      :ARG0 (m / man)
      :ARG1 m
      :ARG2 (u / unconscious)
      :ARG3 (s / shriek-01
          :ARG0 m))
```

Parsed as: *The man's shrieking caused him to become unconscious*.

Although many sentences can be accurately parsed according to verbal semantics alone, a comprehensive meaning representation should have the machinery to represent all meaning-bearing elements of a language. Furthermore, argument structure constructions constitute some of the most frequent constructions of English, and they are cross-linguistically common as well (Goldberg, 1995; Perek and Lemmens, 2010), yet any instantiation with a relatively infrequent, creative verb that does not share the semantics of the construction will not be represented accurately under current guidelines.

To address this gap, we present CxGr-AMR,[2] which extends the AMR formalism according to the theoretical tenets of CxG. After providing background information on AMR and CxG (§2), as well as our choice of the AMR formalism (§3), we detail the shortcomings of the current Standard-AMR representation of four English Argument Structure Constructions (ASCs) (§4): (1) Resultative, (2) Caused Motion, (3) Intransitive Motion, and (4) Ditransitive. In each subsection of §4, we then provide the proposed CxGr-AMR roleset. We describe how we develop a semi-automatic, expert-annotator-in-the-loop pipeline for the creation of a large corpus of CxGr-AMR (§5).

2. Background & Related Work

2.1. Standard-AMR

AMR represents the meaning of a sentence in the form of a graph (Banarescu et al., 2013). The "abstract" nature of AMR is intended to capture concepts and the relations between them devoid of idiosyncratic syntactic differences. Thus, for example, the realizations of "*fear*" as a verb or noun, and the adjective "*afraid*" are all annotated identically in AMR. This makes AMR an appealing formalism for representing meaning agnostic to the language and linguistic realization (Xue et al., 2014).

The edges of the graph represent semantic relations. AMR leverages the PropBank (Palmer et al., 2005) lexicon of "rolesets": for a given relation, the set of semantic roles licensed by that relation onto its syntactic arguments. The most frequent, and often "core" arguments of a relation are given argument numbers: ARG0-5 (Bonial et al., 2012).

AMR has an extensive set of relations that can be flexibly combined with any relation, such as the basic `:mod` or "modifier" role used in (1) to represent the meaning of "*unconscious*" with respect to the shrieking event.[3]

2.2. Lexical Bias in Standard-AMR

Although neither PropBank nor AMR explicitly embrace any theoretical viewpoint, the fact that most relations in the shared PropBank/AMR lexicon are individual lexical items leads to the implicit assumption that a single lexical head projects or licenses the semantic roles and argument structure of a clause. This approach is adequate for many sentences, wherein the semantics and argument structure of the verb are compatible with, and align with, the semantics and argument structure of the broader syntactic environment the verb is found in.

However, such an approach becomes problematic when facing cases wherein the lexical verb's senses are not commonly observed to license the arguments of the surrounding syntactic environment (first pointed out with respect to light verb annotation (Bonial and Palmer, 2016)). For example, the verb "*shout*," in its communication sense, commonly licenses a shouter and an utterance shouted, as well as an addressee shouted at or to. In the sentence "*The coach shouted their players into a queue*", the shouting event itself shares semantics with the communication sense, and yet the direct object is "*their players*" (not the utterance) and the prepositional phrase is the final goal location "*into a queue*" (not the addressee).

From a CxG theoretical approach, patterns of syntactic slots can assign semantic roles to the lexical items situated within them (Goldberg, 1995). Thus, the shouting example can be understood as an instantiation of the Caused Motion construction, which carries the motion semantics and assigns the thing-moved role as well as the goal location. If one instead posits that constructions can assign the semantic roles of a pattern of syntactic slots, then this vastly reduces the requisite size of the database of relations (Bonial et al., 2017). CxGr-AMR operationalises this idea within AMR by introducing rolesets that encode these clause-level roles *directly*, while preserving the verb's semantics as a concomitant event.

2.3. Data: The CoGS corpus

To evaluate the Standard-AMR treatment of constructions as well as motivate the requirements of

[2] Pronounced "Construction Grammar." The intention of this title is not to suggest that a grammar is encoded in AMR, but rather that we extend AMR in a manner following CxG approaches.

[3] AMR guidelines: https://github.com/amrisi/amr-guidelines

Construction	Meaning Description	Form Description	Example
Caused Motion	Agent of the action denoted by the verb causes theme to move along or towards a goal.	PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$	[[Workers]$_1$ dumped$_2$ [large burlap sacks of the imported material]$_3$ [into a huge bin. . .]$_4$]$_5$
Intransitive Motion	A theme carries out an event that causes or accompanies movement.	PHONOLOGY: /A_1 B_2 $C_{3/4}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBL_3]$_{VP}$]$_4$	[[The cyclone]$_1$ was sweeping$_2$ [across the state . . .]$_3$]$_4$
Ditransitive	Agent of the action denoted by the verb is construed as (intending to) cause a recipient to receive a theme.	PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBJ_4]$_{VP}$]$_5$	[[My ex]$_1$ feeds$_2$ [my kids]$_3$ [cheese whiz and R.C. Cola.]$_4$]$_5$
Resultative	Agent of the action denoted by the verb causes a patient to change / become a resulting state.	PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/ MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$	[[It]$_1$ jerks$_2$ you$_3$ awake$_4$ with the first sentence. . .]$_5$

Table 1: Excerpt of the argument structure construction descriptions that we embrace to develop our CxGr-AMR rolesets, drawn from Bonial and Tayyar Madabushi (2025).

the CxGr-AMR formalism, we leverage the Construction Grammar Schematicity (CoGS) corpus (Bonial and Tayyar Madabushi, 2025). The CoGS corpus is made up of about 600 instances of phrasal constructions, roughly equally distributed across 10 construction types (the Resultative construction is more frequent, given corpus expansion described in Scivetti et al. (2025)), including the four ASCs of focus in the present research. See Table 1 for CoGS ASC definitions and examples. The CoGS corpus is unique and particularly useful for this research because it is specifically made up of instances of constructions wherein the verbal semantics alone cannot account for the broader semantics.

This makes CoGS an ideal testbed for exploring a meaning representation's coverage of semantics. In other corpora, Standard-AMR may seem perfectly adequate because of the frequency with which ASCs are instantiated by a single, compatible lexical verb (such as "give" in the case of the Ditransitive, "put" in the case of Caused Motion, "make" in the case of the Resultative, and "go" in the case of Intransitive Motion). In CoGS, the same constructions are instantiated creatively with verbs that do not normally license the arguments and respective semantic roles surrounding it. CoGS also abstracts away from the separate problem of construction detection in raw text. Without a resource that makes constructionally contributed roles explicit (and models trained on it), we should not expect reliable identification of these instances at scale, which is part of the motivation for CxGr-AMR.

There are different schools of CxG, which differ slightly in details of how constructions are defined, enumerated, related to one another, and how they interact (see Hoffmann and Trousdale (2013) for an overview). We embrace the definition of constructions as pairings of form and meaning, where the form may be at the morphological, lexical, or phrasal level (Hoffmann, 2022). While the present research is largely compatible with all schools of CxG, we draw most heavily on Goldberg (1995). That work lays out the syntactic slots and associated semantic roles of ASCs (see Table 2). Our constructional rolesets are directly inspired by these associations between constructional slots and semantic roles.

Syntactic Slots	Subject	Verb	Object	Oblique
Lexical Items	*Firefighters*	*cut*	*the child*	*free*
ASC Roles	Cause	—	Patient	Result State
Verb Roles	Agent	—	—	—

Table 2: Resultative construction diagram demonstrating how both ASCs and verb semantic roles fuse in *Firefighters cut the child free*.

3. Rationale for an AMR Extension

Resources like Standard-AMR and PropBank demonstrate that annotating the mapping between words (and subgraph structures or syntactic trees) and particular semantic roles enables systems trained on such annotations to identify who is doing what to whom, when, where, and why (Palmer et al., 2022). Our analyses of Standard-AMR parses of CoGS argument structure constructions show that, when semantic roles are treated as stemming only from lexical relations, parsers trained on such data misparse cases where the verb's roleset does not align with the clause pattern.

Nonetheless, AMR is flexible enough to be ex-

tended to ASCs when we simply introduce rolesets for phrasal constructions.[4] These rolesets allow us to define the basic semantics of the construction, as well as the semantic role of each slot of the construction. Furthermore, although the present research is restricted to English, because AMR eschews annotating with respect to the part of speech of a given concept, it is not tied to an English-centric syntactic realization of a construction.

Finally, although the present research is the first exploration of Standard-AMR treatment of ASCs, there is a tradition of positing some rolesets for a limited set of English constructions in AMR. Past research has established rolesets for phrasal constructions including the Comparative Correlative (Bonial et al., 2018). However, past research assumes that verbal lexical relations are adequate for ASCs, whereas we provide evidence challenging this assumption.

4. ASCs: Problems & Solutions

In each section to follow, we analyze an individual ASC and show where the current Standard-AMR formalism cannot adequately capture the semantics of these constructions. We then outline the new CxGr-AMR roleset.

4.1. Resultative

The Resultative construction involves an Agent's participation in one event that causes a change of state in a Patient. As shown in Table 1, the Resultative construction is made up of 4 fully flexible (i.e. there are no fixed words or phonological forms) constructional slots:

PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$

The subject (A) is the agent of some event expressed by the verb (B), wherein carrying out that event causes the object (C) to change to the resulting state expressed by the oblique (D).

4.1.1. Standard-AMR Treatment: Resultative

Resultatives are particularly problematic within a Standard-AMR treatment first because AMR generally assesses arguments according to the projection of roles by a lexical verb relation, and second because Standard-AMR lacks a modifier role for results or secondary predication often expressed by adjectives in the Resultative. Thus, our assessment of Standard-AMR parses of CoGS Resultatives demonstrates a common pattern of misrepresenting the object of the lexical verb as a theme or patient of that verb and the expression of the resulting state as a modifier of the lexical verb. For example, the instance *The man shrieked himself unconscious* is parsed as seen in Example (1) (§1) according to the semantics of the shriek-01 roleset:[5]

ARG0-PAG: shrieker
ARG1-PPT: shriek itself, or utterance
ARG2-GOL: unfortunate listener

What the parse in (1) represents with respect to the meaning of this sentence is that the man is both the shrieker (ARG0) and the utterance (ARG1), and that the shrieking event is modified by *unconscious*. Thus, the parse incorrectly analyzes the false object as the patient and does not provide an informative analysis of the role of the resulting state.

While the automatically parsed data consistently exhibits this pattern of misrepresentation, which is rooted in Standard-AMR's anchoring of argument structure in lexical rolesets, we also consider whether Standard-AMR has in principle a way to represent Resultative semantics even when current parsers fail to recover it from existing corpora. At best, one can introduce a generic causal relation (e.g., cause-01) to relate the concomitant event (shriek-01) to the ensuing state (unconscious-01):

```
(c / cause-01
    :ARG0 (s / shriek-01
              :ARG0 (m / man))
    :ARG1 (u / unconscious-01
              :ARG1 m))
```

However, this representation collapses a distinction that English overtly encodes in form: Resultatives package the causing event and the resultant state as a single constructional causal complex, whereas "because" clauses encode a much broader class of causal/explanatory relations. Importantly, the contrast is not that "because" cannot be temporally overlapping (indeed it can), but that cause-01 alone does not mark (i) the event-internal/means-like construal characteristic of Resultatives (the shrieking is construed as the mechanism by which the change of state comes about), (ii) the constructional linking that tightly couples a participant in the concomitant event with the undergoer of the resultant state, and (iii) the result entailment/culmination that motivates the Resultative form. As a consequence, a graph using only cause-01 does not preserve why a speaker chose a Resultative rather than an explanatory "because" relation, and it provides no dedicated locus for construction-specific constraints that downstream parsing or generation would need to recover the intended construal.

4.1.2. CxGr-AMR: Resultative

To capture these constructional slots and their semantics, we develop the constructional roleset:

[4] See Appendix A for a discussion of other relevant NLP resources and AMR extensions.

[5] All rolesets are copied verbatim from https://propbank.github.io/v3.4.0/frames/.

Resultative-91

Arg0-Cause (SBJ slot (A))
Arg1-Patient (OBJ slot (C))
Arg2-Result (OBL slot (D))
Arg3-Concomitant_Event (V slot (B))

This roleset enables mapping the constructional slots (expressed as subgraphs) to the semantic roles of the construction. It leaves underspecified the precise nature of the verbal semantics within the construction, as this can vary. Additionally, we note that the Arg0 can vary in its agentivity depending upon the verb in the structure. For example, in *He knocked himself unconscious*, while *he* is the prototypical agent of the sentence (in the sense of Dowty (1991)), he is likely not intentional in this act. Example (2) (§1) shows the CxGr-AMR analysis of the example utterance *The man shrieked himself unconscious* discussed above.

4.2. Caused Motion

The Caused Motion construction involves an agent doing something that causes a theme to move along a path or towards some goal. The Caused Motion construction also involves four fully flexible or schematic slots:

PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBL_4]$_{VP}$]$_5$

The subject (A) is the agent of an event expressed by the verb (B), wherein carrying out that event causes the object (C) to move along some path or to a destination/goal expressed as an oblique (D).

4.2.1. Standard-AMR: Caused Motion

Standard-AMR has a fairly comprehensive inventory of roles that capture motion semantic roles including `:source`, `:path`, `:direction`, and `:destination`. Thus, if the lexical verb involved in the Caused Motion construction does involve any motion semantics, then the Standard-AMR treatment is potentially adequate. However, if the lexical verb of the Caused Motion construction does not typically involve any motion semantics, then Caused Motion semantics are often misanalyzed because the syntactic object of the lexical verb is construed as a prototypical argument of that verb in its typical sense. For example, the sentence *They laughed the actor off the stage* is automatically parsed according to the laugh-01 roleset:

ARG0-PAG: laugher
ARG1-PPT: cognate object
ARG2-CAU: source of joy
ARG3-PRD: end state of Arg0, as result of laughing

```
(z0 / laugh-01
    :ARG0 (z1 / they)
    :ARG2 (z2 / person
                :ARG0-of (z3 / act-01))
    :location (z4 / off
                :op1 (z5 / stage)))
```

The above parse denotes that they (ARG0) are laughing at the actor (ARG2), and that this laughter is happening off-stage (:location). Thus, this parse fails to capture any motion semantics at all.

One could capture Caused Motion instances with non-motion verbs using the introduction of causation and potentially introducing the implicit element of motion (i.e. they laughed, causing the actor to move off-stage):

```
(c / cause-01
  :ARG0 (l / laugh-01
          :ARG0 (t / they))
  :ARG1 (m / move-01
         :ARG1 (p / person
                 :ARG0-of (a / act-01))
         :ARG2 (o / off
                 :op1 (s / stage))))
```

Like the Resultative treatment, however, we note that this fails to capture the tightly coupled, complex-event nature of the causal and temporal relations between the motion and the concomitant event of laughter.

4.2.2. CxGr-AMR: Caused Motion

To capture the constructional slots and their semantics, we develop the following roleset:

Caused-Motion-91

Arg0-Cause (SBJ slot (A))
Arg1-Theme (OBJ slot (C))
Arg2-Goal (OBL slot (D))
Arg3-Concomitant_Event (V slot (B))

This roleset similarly supports mapping the constructional slots to the semantic roles of the construction, as exemplified by the newly annotated example *They laughed the actor off the stage*:

```
(c / caused-motion-91
    :ARG0 (t / they)
    :ARG1 (p / person
        :ARG0-of (a / act-01))
    :ARG2 (o / off
        :op1 (s / stage))
    :ARG3 (l / laugh-01
        :ARG0 t))
```

This represents the meaning: Their laughing caused the actor to move off the stage.

4.3. Intransitive Motion

The Intransitive motion construction involves a Theme in an event that causes or accompanies movement in space. The Intransitive Motion construction consists of three schematic constructional slots:

PHONOLOGY: /A_1 B_2 $C_{3/4}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBL_3]$_{VP}$]$_4$
The subject (A) is a theme carrying out an event (B) that causes or accompanies movement along a path expressed as an oblique (C).

4.3.1. Standard-AMR: Intransitive Motion

As described for the Caused Motion construction, the inventory of motion-related modifier roles within the Standard-AMR inventory enables reasonable annotation of Intransitive Motion construction in cases where the lexical verb can felicitously be construed as compatible with such motion arguments. In these cases, the lexical relation roleset also reflects motion semantics, so that the argument structure of the lexical verb aligns for one or more arguments with the expected arguments of the Intransitive Motion construction.

In cases where the lexical verb is not semantically compatible with such motion modifiers, such as sound emission verbs, the Standard-AMR parser output misrepresents the Intransitive Motion meaning. For example, *...troops rumbled along the main road...* is annotated with the following roleset for the sound emission verb rumble-01 and assigned the following parse:

ARG0-PPT: entity rumbling
ARG1-PPT: sound/utterance
ARG2-GOL: hearer

```
(z0 / rumble-01
    :ARG0 (z5 / troop
    :location (z8 / along
            :op1 (z9 / road
                    :mod (z11 / main))))
```

This parse fails to capture that the troops are in motion at all. However, this sentence is assigned a gold standard parse in the AMR corpus and is exemplified in Bonial et al. (2018), where the `:location` above is simply swapped for a `:path` modifier. Thus, the Standard-AMR machinery can posit the path argument, but fails to account for the fact that the sound emission lexical verb would not generally license motion arguments.

4.3.2. CxGr-AMR: Intransitive Motion

We represent the constructional slots and semantics of the Intransitive Motion construction with the following roleset:

Intransitive-Motion-91
Arg1-Theme (SBJ slot (A))
Arg2-Path/Goal (OBL slot (C))
Arg3-Concomitant_Event (V slot (B))

This roleset can be applied to the instance *Troops rumbled along the main road* to give the following CxGr-AMR graph:

```
(i / intransitive-motion-91
    :ARG1 (t / troop
    :ARG2 (a / along
            :op1 (r / road
                    :mod (m / main)))
    :ARG3 (r2 / rumble-01
            :ARG0 t))
```

This precisely captures the motion semantics of the construction: the troops move, rumbling, along the main road. Notice that the Intransitive Motion construction is a more parsimonious way of expressing this precise framing of the event, as opposed to the preceding rephrasing with "move."

4.4. Ditransitive

The Ditransitive construction involves an agent carrying out an event that is construed as causing a recipient to receive a theme. The construction consists of four fully schematic slots:

PHONOLOGY: /A_1 B_2 C_3 $D_{4/5}$/
MORPHOSYNTAX: [SBJ_1 [V_2 OBJ_3 OBJ_4]$_{VP}$]$_5$
The subject (A) carries out an action denoted by the verb (B), which is construed as (intending to) cause a recipient, expressed as the object (C), to receive a theme, expressed as a second object (D).

4.4.1. Standard-AMR: Ditransitive

When analyzing Standard-AMR parses of CoGS Ditransitives, we see again the pattern emerge wherein if the lexical verb within the Ditransitive is semantically compatible with the meaning of the Ditransitive, and therefore one or more arguments of the lexical verb align with expected arguments of the Ditransitive, then the Standard-AMR parse is generally adequate, although perhaps lacking the precise semantics. Standard-AMR also includes a `:beneficiary` that can be used in cases where the lexical verb roleset does not include a recipient or beneficiary argument.

However, we again see the pattern that verbs that are not typically found in a Ditransitive structure and would not generally license a recipient are given problematic parses in Standard-AMR. For example, *They're going to kill Reagan a commie* is assigned the kill-01 roleset and parsed in the following way:

ARG0-PAG: killer
ARG1-PPT: corpse
ARG2-MNR: instrument

```
(z0 / kill-01
    :ARG0 (z1 / they)
    :ARG1 (z2 / person
              :wiki "Ronald_Reagan"
              :name (z3 / name
```

```
                    :op1 "Reagan")
            :mod (z4 / commie)))
```

This parse denotes that they (ARG0) kill Reagan (ARG1), who is modified as communist. This demonstrates how automatic parsers trained without regard to constructional semantics can wholly misrepresent the semantics of a sentence.

4.4.2. CxGr-AMR: Ditransitive

To represent the meaning of the Ditransitive, we develop the following roleset:

Ditransitive-91

Arg0-Cause (SBJ slot (A))
Arg1-Theme (OBL slot (D))
Arg2-Recipient (OBJ slot (C))
Arg3-Concomitant_Event (V slot (B))

When applied to our earlier, problematic example, we now derive the following parse:[6]

```
(d / ditransitive-91
   :ARG0 (t / they)
   :ARG1 (c / commie)
   :ARG2 (p / person
            :wiki "Ronald_Reagan"
            :name (n / name
                     :op1 "Reagan"))
   :ARG3 (k / kill-01
            :ARG0 t
            :ARG1 c))
```

This correctly captures the relations among arguments as well as the (intended) Ditransitive transfer/beneficiary semantics.

4.5. Constructional Roleset Summary

The assignment of numbered arguments is closely paralleled for each constructional roleset. This is intentional, and the argument number assignment itself is guided by the PropBank guideline for Arg0 to be a prototypical agent, Arg1 to be a prototypical patient, and Arg2 to be another core argument, but what this argument is depends upon the class of the relation. For example, relations that lexically entail motion would be a goal, while those that entail transfer have a recipient Arg2. Consistent numbering facilitates combining data for different argument types across relations, especially given that data might be sparse for any single relation. Thus, CxGr-AMR provides another source of data enabling understanding of how agents, patients, and motion-related arguments are realized in English because Arg0s, Arg1s and Arg2s across motion relations, now including phrasal constructions, can be combined.

To maintain symmetry and consistency with the argument structures of existing relations, we shift the concomitant event slot to Arg3. Thus, within CxGr-AMR, consistency in role numbers allows one to generalize over, for example, concomitant events for a particular set of constructions. As we expand to additional constructions, this also facilitates potentially combining or splitting constructions. For example, it is possible to annotate Intransitive Resultatives with the Resultative-91 roleset (while simply omitting the Arg0), but alternatively, the intransitive cases could easily be detected automatically given their unique argument structure signature and assigned to a finer-grained roleset if desired.

Finally, we emphasize that the CxGr-AMR rolesets are not intended to encode English syntax or to elevate a particular surface pattern to the level of meaning representation. The syntactic "signatures" we reference throughout §4 are used only as a practical cue for where English reliably packages certain meanings; what the rolesets make explicit is the constructionally contributed semantics (e.g., result entailment, caused motion, caused transfer) that would otherwise be implicit or mis-attached under a purely lexically anchored analysis. In this sense, the constructional node functions as a typed semantic operator that remains compatible with Standard-AMR (lexical predicates are retained as concomitant events) while providing a uniform locus for clause-level meaning that can generalize beyond any single verb and, in principle, beyond English-specific realizations.

5. CxGr-AMR Corpus Development

Writing AMR graphs has been established as a notoriously time-consuming and expensive task (Martin et al., 2020). Banarescu et al. (2013) have asserted that, using the AMR annotator, annotation time averages around 7-10 minutes per sentence.[7] For our targeted CoGS corpus of more than 350 sentences, a large portion of which are lengthy (i.e. 20+ words), and would require additional annotation time, we sought to shorten the annotation process so that we could focus on the new constructions proposed in this paper.

To this end, we formulated a semi-automated annotation process composed of four parts: 1) Standard-AMR parsing for the initial graphs, 2) LLM-based editing to incorporate the updated rolesets, 3) rule-based logical fixes, and 4) manual post-editing of the resulting graphs. Using this system allowed us to bypass the expensive, start-from-scratch annotation step and move directly to the faster process of editing the generated graphs. We

[6] To reduce the size of the graph for this paper, we have simplified the structure of "commie", which calls for representation using the `:have-org-role-91` roleset.

[7] This estimate reflects the time for *trained* annotators. Annotators not extensively trained easily require 30 minutes to an hour.

note that while CoGS provides construction labels for our initial study, the longer-term goal is for CxGr-AMR to supply the supervision needed to learn these constructional meanings from raw text rather than assuming gold construction identification.

5.1. Automated Parsing

For the automated AMR parsing portion of this project, we utilized the SPRING parser (Bevilacqua et al., 2021), a state-of-the-art AMR parser with a SMATCH score (Cai and Knight, 2013) of 83.0 (Blloshmi et al., 2021) on the LDC2020T0 dataset (Knight et al., 2020).[8]

Next, we manually edited from the AMR parser graphs a small subsection of the data that included 12 sentences divided equally by construction type: Resultative, Caused Motion, Intransitive Motion, and Ditransitive. Each of the three sentences chosen included one simple sentence (wherein the head node that would become -91 roleset was not embedded within another clause), a complex sentence (wherein the -91 roleset was embedded within another clause), and a passive variant. This was done so that we would have a few in-context learning examples for each sentence type, thereby demonstrating how the -91 roleset should be translated from the Standard-AMR.[9]

Once these graphs were produced, we checked the graphs for common errors, including if the model mistakenly added two CxGr-AMR rolesets to the graph, if any of the variable names were duplicated, if a graph was not produced, or if there were not the same number of open parentheses to match closed parentheses in the graph.[10] The last error type (of parentheses issues) was the most common; a simple script fix was used to delete or add parentheses.

On the remaining set of graphs that still had errors after rule-based corrections, we ran these using the more expensive GPT 5.2 model. Most of the remaining problematic graphs were longer on average than those without obvious errors,[11] and we saw an improvement in graph quality with the larger model. Once these were run, the rule-based corrections script was run once more and parentheses were checked and fixed again.

[8]For a general overview of AMR parsers, please see the NLP progress leader-board that is available here: https://nlpprogress.com/english/semantic_parsing.html#amr-parsing.

[9]The exact prompts with these examples are in subsection 8.1 in Appendix B. The first run of corrections included these instructions and were run on `GPT-5-mini`.

[10]See Figure 1 in Appendix B for details on the number of these errors in the AMR `GPT-5-mini` graphs.

[11]See Figure 2 in Appendix B.

5.2. Manual Edits

After automatic processing, the graphs were manually reviewed. We divided the data by construction type so each of our four annotators could specialize in a single construction. Graphs for manual correction were presented in a spreadsheet that included a binary correctness judgment, a column for corrected graphs, and a notes column for cases requiring adjudication. Annotators were instructed to flag uncertain cases for group discussion, which we resolved through adjudication sessions. Annotators followed roleset specifications when adjusting concomitant events and avoided introducing roles not licensed by the frame index.[12]

Manual review revealed that while many graphs were structurally well-formed, semantic and roleset-level issues remained common. In one annotator's subset of Ditransitives, for example, 33/50 graphs contained at least one error. Frequent error types included incorrect argument layouts within rolesets (such as ARG2 instead of ARG3), selection of an incorrect roleset sense (such as *run-01* instead of *run-02*), hallucinated roles and rolesets, misplaced scope (such as temporal modifiers attached at the wrong level), and occasional spurious modifiers or wiki links. Pronoun normalization errors, like *I* vs. *me*, also appeared.

5.3. Corpus Statistics

In Table 3, we have summarized the current corpus. We note the correlation between average sentence length to graphs that required manual fixes after the automated annotation process. There is a clear trend that the auto-generated graphs suffer in quality given longer sentences, thereby requiring more manual fixes.[13] Additionally, given the high percentages of graphs that were fixed, manual checking and correction is not a step that can be bypassed: when 69.2% of Resultative graphs were not generated correctly, it is clear that a human still needs to be in the annotation loop.

6. Future Work

Our annotation experiments indicate that fully automatic conversion remains unreliable. This motivates future work aimed at improving annotation support and automatic processing for these constructions. Therefore, we plan to develop dedicated annotation tools for CxGr-AMR and to extend it to other constructions beginning with those in the CoGS corpus (see Figure 4 in Appendix B). As we develop these tools and conduct manual

[12]Rolesets found here: https://propbank.github.io/v3.4.0/frames/.

[13]A visual is available in Figure 3 in Appendix B.

CxN Type	Count	% Fix	LenGPT	LenFix	LenAll
Resultative	192	69.2	8.92	12.56	17.27
Intransitive	58	35.3	12.09	19.17	22.05
Caused Motion	55	50	16.14	21.86	24.11
Ditransitive	50	66	7.29	19.67	15.46
Total	355	60.8	10.15	17.21	18.86

Table 3: Construction type, count, percentage of graphs needing a manual fix, avg sentence length of graphs that did *not* need a manual fix, avg sentence length of graphs needing manual fix, and overall avg sentence length.

edits more broadly, we will also conduct broader inter-annotator-agreement measures of CxGr-AMR. We will also evaluate whether CxGr-AMR improves generalization in downstream tasks that depend on explicit argument structure.

In the future, we will explore how these methods can support cross-linguistic meaning representation, particularly for languages where morphosyntax and clause patterns contribute important semantic roles that are not well captured by lexical rolesets alone (see Appendix C for an initial exploration within Quechua).

One natural extension is to Uniform Meaning Representation (UMR) (Van Gysel et al., 2021), which builds on AMR's PropBank backbone while adding aspect, modality, discourse structure, and a more abstract inventory of participant roles in order to be more cross-linguistically applicable. While UMR introduces generalized roles[14] such as *actor*, *undergoer*, and *theme*, these remain largely tied to predicate-level argument structure and do not fully address meaning contributed by constructions. Cross-linguistically, many semantic roles arise from morphosyntactic patterns (e.g., caused motion, applicatives, valency-changing morphology) rather than from the lexical predicate itself.

Our approach complements UMR by introducing constructional rolesets that explicitly encode these patterns independently of any single predicate. Importantly, this also yields a more efficient and interpretable representation, where constructional meaning is made explicit rather than implicitly distributed across predicate-specific roles. This is particularly useful in languages where PropBank-style rolesets are incomplete or unavailable, where the UMR roleset assignment system would be utilized, since these defined structures may be used if the concept exists in another language (see Appendix C for examples).

Our solution extends easily to English UMR, and the example below illustrates this integration in English while Appendix C provides further examples for Quechua: while the standard UMR graph encodes the event via *drive-01*, the constructional extension introduces an *intransitive-motion-91* node that captures motion semantics contributed by the construction, making the decomposition of meaning more transparent by separating path from manner.

(1) Standard AMR: *He drove through the tunnel.*

```
(d / drive-01
    :ARG0 (h / he)
    :path (t / tunnel))
```

(2) UMR (Bonn et al., 2024):

```
(d/ drive-01
    :ARG0 (p/ person
        :ref-person 3rd
        :ref-number Singular)
    :path (t/ tunnel)
    :aspect Performance)
```

(3) UMR + CxGr-AMR:

```
(m / intransitive-motion-91
    :ARG1 (p / person
        :ref-person 3rd
        :ref-number Singular)
    :ARG2 (r / through
        :op1 (t / tunnel))
    :ARG3 (d / drive-01
        :ARG0 p
        :aspect Performance))
```

7. Conclusions

AMR remains important even in the current landscape of LLMs, particularly in settings where meaning representations must be explicit and auditable. Yet Standard-AMR largely links core semantic roles to lexical rolesets, which leads to workarounds for clauses whose argument structure is not naturally licensed by the verb alone. This coverage gap matters because it affects very common and productive patterns of English, and it therefore limits the reliability of AMR-based semantic parsing.

We addressed this limitation by introducing CxGr-AMR, an extension of AMR that adds explicit rolesets for clause patterns that contribute semantic roles *beyond* those projected by the lexical predicate. We grounded these rolesets in existing linguistic analyses, and we released a dataset of 355 instances annotated in both Standard-AMR and CxGr-AMR, together with the roleset specifications and annotation guidelines. Compared to Standard-AMR workarounds that rely on generic causation or motion paraphrases, CxGr-AMR provides a more direct and transparent representation of meaning.

[14] UMR guidelines: https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md

Limitations

Although the framework is designed with cross-linguistic applicability in mind (see Appendix C), the present study evaluates the approach on English data. Extending the framework to additional languages will require further investigation beyond the scope of this initial paper on how constructional semantics interact with language-specific patterns.

Another limitation concerns the annotation process. While part of the annotation pipeline was automated, a substantial portion (about 60%) of graphs required at least one manual correction; thus, it is clear that human annotators must remain in the loop to accurately capture these constructions, particularly when introducing new rolesets and structural conventions. As well, the semi-automatic pipeline relies on large language models for graph editing. While this approach substantially reduces valuable annotation time, it could introduce potential concerns related to reproducibility, model drift, and hallucinated structures. This fact should be taken into account by researchers interested in utilizing, reproducing, or extending this work.

8. Bibliographical References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Katrien Beuls and Paul Van Eecke. 2025. Construction grammar and artificial intelligence. In Mirjam Fried and Kiki Nikiforidou, editors, *The Cambridge Handbook of Construction Grammar*, pages 543–571. Cambridge University Press, Cambridge, United Kingdom.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12564–12573.

Rexhina Blloshmi, Michele Bevilacqua, Edoardo Fabiano, Valentina Caruso, and Roberto Navigli. 2021. Spring goes online: end-to-end amr parsing and generation. In *Proceedings of the 2021 conference on empirical methods in natural language processing: system demonstrations*, pages 134–142.

Hans C Boas. 2021. Construction grammar and frame semantics. In *The Routledge handbook of cognitive linguistics*, pages 43–77. Routledge.

Claire Bonial, Bianca Badarau, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Tim O'Gorman, Martha Palmer, and Nathan Schneider. 2018. Abstract meaning representation of constructions: The more we include, the better the representation. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Claire Bonial, Kathryn Conger, Jena D Hwang, Aous Mansouri, Yahya Aseri, Julia Bonn, Timothy O'Gorman, and Martha Palmer. 2017. Current directions in english and arabic propbank. In *Handbook of linguistic annotation*, pages 737–769. Springer.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-amr: abstract meaning representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695.

Claire Bonial, Jena Hwang, Julia Bonn, Kathryn Conger, Olga Babko-Malaya, and Martha Palmer. 2012. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*, 48:14.

Claire Bonial and Martha Palmer. 2016. Comprehensive and consistent propbank light verb annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3980–3985.

Claire Bonial and Harish Tayyar Madabushi. 2025. Constructing understanding: on the constructional information encoded in large language models. *Language Resources and Evaluation*, 59(4):4559–4598.

Julia Bonn, Matthew J Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajic, Kenneth Lai, James H Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benét Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E L Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint*

International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 2537–2547.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752.

David Dowty. 1991. Thematic proto-roles and argument selection. *language*, 67(3):547–619.

Charles J Fillmore. 1967. The case for case.

Charles J. Fillmore. 1988. The mechanisms of "construction grammar". In *Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 35–55.

Mirjam Fried and Jan-Ola Östman. 2004. Construction grammar: A thumbnail sketch. In Jan-Ola Östman and Mirjam Fried, editors, *Construction grammar in a cross-language perspective*, pages 1–86. John Benjamins, Amsterdam, Netherlands.

Adele E. Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, Chicago, IL, USA.

Thomas Hoffmann. 2022. *Construction grammar*. Cambridge University Press.

Thomas Hoffmann and Graeme Trousdale. 2013. *The Oxford handbook of construction grammar*. Oxford University Press.

Paul Kay and Charles Fillmore. 1999. Grammatical constructions and linguistic generalizations: The *What's X Doing Y?* construction. *Language*, 75(1):1–33.

Kevin Knight, Bianca Badarau, and Laura Banarescu. 2020. *Abstract meaning representation (amr) annotation release 3.0*. Lead Discovery Center LDC.

Mary Martin, Cecilia Mauceri, Martha Palmer, and Christoffer Heckman. 2020. Leveraging non-specialists for accurate and time efficient amr annotation. In *Proceedings of the LREC 2020 Workshop on" Citizen Linguistics in Language Resource Development"*, pages 35–39.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2022. Machine learning for semantic role labeling. In *Semantic Role Labeling*, pages 31–52. Springer.

Florent Perek and Maarten Lemmens. 2010. Getting at the meaning of the english at-construction: the case of a constructional split. *CogniTextes. Revue de l'Association française de linguistique cognitive*, 5(Volume 5).

Veronica Juliana Schmalz and Frederik Cornillie. 2022. Towards truly intelligent and personalized icall systems using fluid construction grammar. In *Colpaert, J., Wang, Y., & Stockwell, G.(Eds.)(2022). Proceedings of the XXIst International CALL Research Conference. London: Castledown Publishers. https://doi.org/10.29140/9781914291050*, pages 169–179. Castledown Publishers.

Wesley Scivetti, Melissa Torgbi, Mollie Shichman, Taylor Pellegrin, Austin Blodgett, Claire Bonial, and Harish Tayyar Madabushi. 2025. Beyond memorization: Assessing semantic generalization in large language models using phrasal constructions. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1184–1201.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Nianwen Xue, Ondrej Bojar, Jan Hajic, Martha Palmer, Zdenka Uresova, and Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of english amrs to chinese and czech. In *LREC*, volume 14, pages 1765–1772. Reykjavik, Iceland.

Appendix A: Related NLP Resources

There has been a close relationship between construction grammar and frame semantics as both are rooted in Fillmore's case grammar (Fillmore, 1967) (see also Boas (2021).) A basic principle of CxG is that constructions carry meaning, not only lexical items: something that is typically not reflected in semantic formalisms. Specifically within NLP resources, corpora that are exhaustively annotated with semantics, such as PropBank, tie roleset instances to lexical units. Even FrameNet, a repository of frames based directly on Fillmore's work, is

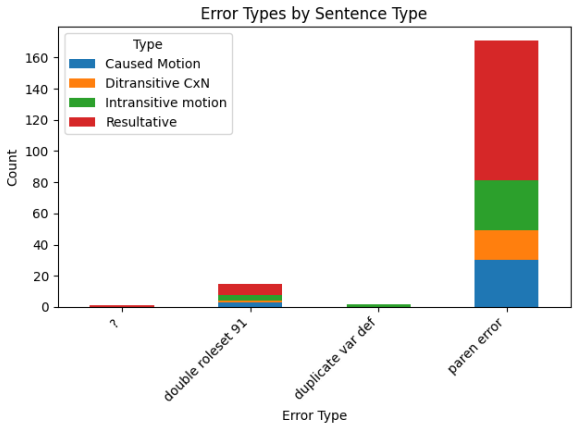


Figure 1: Initial errors found when looking at the GPT-5-mini auto-generated graphs that were made from the input of a sentence, a construction specific prompt, and the AMR SPRING parser graph.

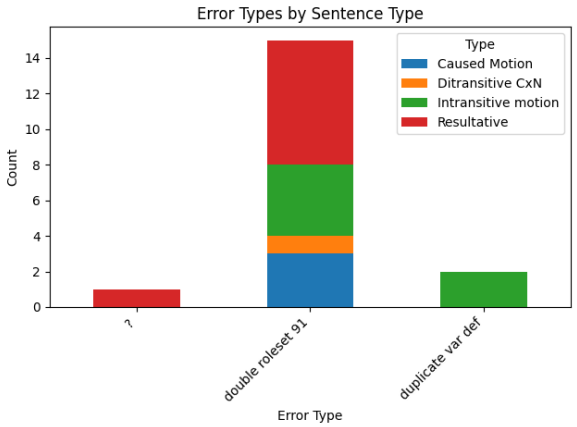


Figure 2: The errors that remained in the auto-generated graphs once the parentheses issues were removed with a simple script.

modeled in such a way where frames are evoked by lexical units (Baker et al., 1998).

As we show throughout Section 4, Standard-AMR solutions are rather complex and lack generality. Thus, we vastly extend the initial efforts of encoding select constructional semantics into AMR Bonial et al. (2018) by leveraging dedicated rolesets on a more abstract level. CxGr-AMR is compatible with other AMR extensions, as each makes use of the Standard-AMR inventory of relations, but then adds other elements, such as speech acts in the case of Dialogue-AMR (Bonial et al., 2020), and cross-lingual meaning-bearing elements in the case of Uniform Meaning Representation (Van Gysel et al., 2021).

Appendix B: LLM Experiment Details

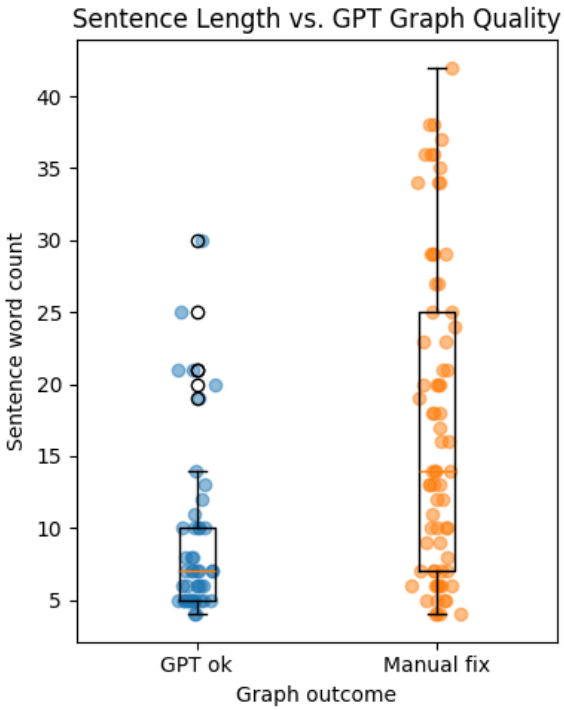


Figure 3: A scatter plot overlaid with a bar plot showing sentence counts for graphs generated by GPT that annotators decided did not need fixes, shown in blue, versus those that did need manual fixes, shown in orange.

8.1. GPT Prompt Examples

Below is each of the verbatim prompts used when prompting GPT models to edit SPRING parser graphs along with their sentences. The model was given a different prompt based on which construction type the sentence was classified under.

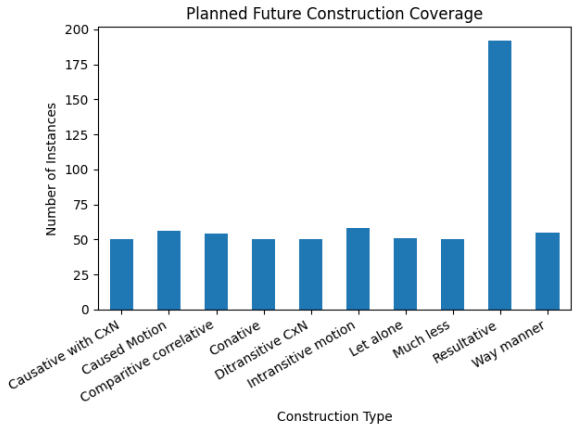


Figure 4: Counts of different construction types available in our corpus. Eventually, we would like to expand the dataset to all of these constructions.

8.1.1. Resultative

```
Update the AMR to use the constructional
roleset resultative-91 when the sentence
expresses a RESULTATIVE construction
(an event causes an entity to end up in
a resulting state/role/location).

I will give you a sentence and AMR
graph with a resultative construction
within it.

Example sentences  Resultative
Construction:
Could he shriek himself unconscious?
Firefighters cut the man free
He had often drunk himself silly

With various OBL types:
He wiped the table very clean. (ADJ)
Pop music drives him round the
bend. (PP)
They elected him president.  (NP)
A judge ordered the recordings to
be made public. (nonfinite clause)

In the auto-generated AMR for the
sentence "the man shrieked himself
unconcious" we get:
(z0 / shriek-01
    :ARG0 (z1 / man
              :ARG1-of z0)
    :mod (z2 / unconscious))

This is a Resultative Construction
wherein:
Sentence: The man shrieked himself
unconscious
Shriek.01 has the arguments:
ARG0-PAG: shrieker
ARG1-PPT: shriek itself, or utterance
ARG2-GOL: unfortunate listener

This can be read as: "man is treated
as shrieker and thing-shrieked,
shrieking event is modified as
unconscious". Which is problematic and
not the actual MEANING of the sentence.

We want to change this graph to instead
look like this:

(r / resultative-91
    :ARG0 (m / man)
    :ARG1 m
    :ARG2 (u / unconscious)
    :ARG3 (s / shriek-01
        :ARG0 m))

Wherein we have the Resultative.91
construction:

Resultative.91
Arg0-Cause: The man
Arg1-Patient: The man
Arg2-Result: unconscious
Arg3-Concomitant_Event: shriek.01
(Arg0: The man)


Here are some additional sentences
with the orginal auto-generated AMR
and the graph we transform them into:

Sentence: she laughed herself silly.

auto-amr:
(z0 / laugh-01
    :ARG0 (z1 / she)
    :manner (z2 / silly))

manually corrected AMR:
(r / resultative-91
    :ARG0 (s / she)
    :ARG1 s
    :ARG2 (s2 / silly)
    :ARG3 (l / laugh-01
        :ARG0 s))

Sentence: This nice man probably just
wanted Mother to KISS him unconscious.

auto AMR:
(z0 / probable
    :domain (z1 / want-01
        :ARG0 (z2 / man
            :ARG1-of (z3 / nice-01)
            :mod (z4 / this))
        :ARG1 (z5 / kiss-01
            :ARG0 (z6 / person
                :ARG0-of (z7 /
                have-rel-role-91
                    :ARG1 z2
                    :ARG2
```

```
                    (z8 / mother)))
            :ARG1 z2
            :manner (z9 / unconscious))
        :mod (z10 / just)))

manually corrected AMR:
(p / probable-01
    :ARG1 (w / want-01
        :ARG0 (m / man
            :mod (t / this)
            :ARG1-of (n / nice-01))
        :ARG1 (r / resultative-91
            :ARG0 (p / person
                :name (n2 / name
                :op1 "Mother"))
            :ARG1 m
            :ARG2 (u /
            unconscious-02
                ARG1 m)
            :ARG3 (k / kiss-01
                :ARG0 p
                :ARG1 m))
        :mod (j / just)))

Sentence: I was scared stupid by
what I saw.

auto AMR:
(z0 / scare-01
    :ARG0 (z1 / thing
        :ARG1-of (z2 / see-01
            :ARG0 (z3 / i
                :mod (z4 /
                stupid))))
    :ARG1 z3)"

manually corrected AMR:
(r / resultative-91
    :ARG0 (t / thing
        :ARG1-of (s / see-01
            :ARG0 (i / i)
            :ARG1 t))
    :ARG1 i
    :ARG2 (s2 / stupid)
    :ARG3 (s3 / scare-01
        :ARG0 t
        :ARG1 i))
```

8.1.2. Caused-motion

```
"I want you to update the graph to
include the new roleset caused-motion-91.

Caused motion constructions occur
when the Agent is doing something that
causes theme to move along or towards
a goal.

An example sentence with Caused Motion
Construction:
They laughed the actor off the stage

Here the caused motion construction
should be:

Caused-Motion.91:
 Arg0-cause: They
 Arg1-theme: the actor
 Arg2-goal: off the stage
 Arg3-concomitant-event: laugh.01
   (Arg0: They)

An example of where the auto-generated
AMR fails to capture this structure is
in the following instance:

The Caused Motion Construction is:
They laughed the actor off the stage

the auto generated graph is:
(z0 / laugh-01
    :ARG0 (z1 / they)
    :ARG2 (z2 / person
              :ARG0-of (z3 / act-01))
    :location (z4 / off
                  :op1 (z5 / stage)))

This graph has the head node as Laugh.01
(mismatch with caused-motion Arg structure)
 ARG0-PAG: laugher
 ARG1-PPT: cognate object
 ARG2-CAU: source of joy
 ARG3-PRD: end state of arg0, as result
 of laughing

Which semantically represents something
like: "They laughed at the actor off-stage"

If we were to fix this to represent the
actual meaning we would use caused-motion-91:
Caused-Motion.91
  Arg0-cause: They
  Arg1-theme: the actor
  Arg2-goal: off the stage
  Arg3-concomitant-event: laugh.01
    (Arg0: They)

Another Common Current AMR Treatment
(Not-As-Bad): Caused-Motion
Sentence: After the seven hundred
passengers on the train were stranded
for two hours , they were GUIDED
through the tunnels to a safe place .

auto AMR graph:
(z0 / guide-01
    :ARG1 (z1 / passenger
        :quant 700)
    :ARG2 (z2 / place
        :ARG1-of (z3 / safe-01))
    :path (z4 / tunnel)
    :time (z5 / after
        :op1 (z6 / strand-01
            :ARG1 z1
            :location (z7 / train)
            :duration (z8 /
            temporal-quantity
                :quant 2
```

```
                :unit (z9 /
                hour)))))

Here Guide.01 (aligns with
caused-motion Arg structure):
 ARG0-PAG: guide, agent
 ARG1-PPT: entity guided
 ARG2-GOL: guided in/through
 ARG3-PRD: signposts along the way

We could fix it with
Caused-Motion.91:
 Arg0-cause: -
 Arg1-theme: the seven hundred
 passengers
 Arg2-goal: to a safe place
 Arg3-concomitant-event: guide.01
  (Arg0: They)


Here are some additional sentences
with the orginal auto-generated AMR
and the graph we want to transform
them into:

Sentence: She wiggled her feet out
of the boots.

auto-amr:
""(z0 / wiggle-01
    :ARG0 (z1 / she)
    :ARG1 (z2 / foot
        :part-of z1)
    :direction (z3 / out-of
        :op1 (z4 / boot)))""

manually corrected AMR:
""(c / caused-motion-91
    :ARG0 (s / she)
    :ARG1 (f / foot
              :part-of s)
    :ARG2 (g / out-of
              :op1 (b / boot))
    :ARG3 (w / wiggle-01
              :ARG0 s
              :ARG1 f))""

Sentence: The stone was THROWN
across the river.

auto-amr:
""(z0 / throw-01
    :ARG1 (z1 / stone)
    :path (z2 / across
              :op1 (z3 / river)))""

manually corrected AMR:
""(c / caused-motion-91
    :ARG1 (s / stone)
    :ARG2 (a / across
        :op1 (r / river))
    :ARG3 (t / throw-01
        :ARG1 s))""

Sentence: Fundamentally, everyone is
entitled to a private life and - no
matter who they are - they have a
right not to have their personal life
DRAGGED through the mud for political
point scoring or for the general
consumption of the public.

auto-amr:
(z0 / and
    :op1 (z1 / entitle-01
        :ARG1 (z2 / life
            :ARG1-of (z3 /
            private-02))
        :ARG2 (z4 / everyone))
    :op2 (z5 / right-05
        :ARG1 z4
        :ARG2 (z6 / drag-01
            :polarity -
            :ARG1 (z7 / life
                :ARG1-of (z8 /
                personal-02
                    :ARG2 z4))
            :ARG2 (z9 / mud)
                :purpose (z10 / or
                    :op1 (z11 /
                    score-01
                        :ARG3 (z12 /
                        point)
                        :mod (z13 /
                        politics))
                    :op2 (z14 /
                    consume-01
                        :ARG0 (z15 /
                        public)
                        :ARG1 z7
                        :ARG1-of
                        (z16 /
                        general-02))))
        :ARG1-of (z17 / regardless-91
            :ARG2 z4))
    :mod (z18 / fundamental))

manually corrected AMR:
""(a / and
    :op1 (e / entitle-01
        :ARG1 (e2 / everyone)
        :ARG2 (l / life
            :ARG1-of (p2 / private-02)))
    :op2 (r / right-05
        :ARG1 e2
        :ARG2 (c / caused-motion-91
            :polarity -
            :ARG1 (l2 / life
                :ARG1-of (p3 / personal-02
                    :ARG2 e2))
            :ARG2 (t / through
                :op1 (m / mud))
            :ARG3 (d / drag-01
                :ARG1 l2)
            :purpose (o / or
                :op1 (s / score-01
```

```
                    :ARG3 (p4 / point)
                    :mod (p5 / politics))
                :op2 (c2 / consume-01
                    :ARG0 (p6 / public)
                    :ARG1 l2
                    :ARG1-of (g / general-02)))
        :ARG1-of (r2 / regardless-91
            :ARG2 e2))
    :mod (f / fundamental))
```

8.1.3. Intransitive

```
I want you to update the AMR graph to
include the new roleset intransitive-motion-91.

Intransitive Motion Construction have
a theme that carries out event that
causes or accompanies movement in space.

An example sentence with Intransitive
Motion Construction:
The fly buzzed into the room

the auto generated graph is:
""(z0 / fly-01
    :ARG1-of (z1 / fly-01)
    :destination (z2 / room))""

The issue with this is:
Fly.01?? Buzz not represented
It gives us an inaccurate meaning
like:"One flying event is the passenger
of another flying event into the room."

If we were to fix this to represent
the actual meaning we would use
Intransitive-Motion.91:
Arg1-theme: The fly
Arg2-goal: into the room
Arg3-concomitant-event: buzz.01
(Arg0: the fly)


Another Common Current AMR Treatment:
Intransitive Motion
Sentence: He ran out of the house.

auto AMR:
(z0 / run-02
    :ARG0 (z1 / he)
    :direction (z2 / out-of
        :op1 (z3 / house)))

Issue:
Run.02 (Motion semantics align
with Intransitive Motion)
ARG0-PPT: runner
ARG1-LOC: course, race, distance
ARG2-PPT: opponent

fix with Intransitive-Motion.91:
Arg1-theme: He
Arg2-goal: out of the house
Arg3-concomitant-event: ran-01
  (Arg0: he)


Additional examples:

Sentence: The Weweantic River
FLOWS through the pond.

auto-amr:
""(z0 / flow-01
    :ARG1 (z1 / river
        :wiki "Weweantic_River"
        :name (z2 / name
            :op1 "Weweantic"
            :op2 "River"))
    :path (z3 / pond))""

manually corrected AMR:
(i / intransitive-motion-91
    :ARG1 (r / river
        :wiki "Weweantic_River"
        :name (n / name
            :op1 "Weweantic"
            :op2 "River"))
    :ARG2 (p / pond)
    :ARG3 (f / flow-01
        :ARG1 r))


Sentence: The river Avon has been
strolled along by thousands of
famous people throughout history.

auto-amr:
(z0 / stroll-01
    :ARG0 (z1 / person
        :ARG1-of (z2 / fame-01)
        :quant (z3 / multiple
            :op1 1000))
    :ARG1 (z4 / river
        :wiki ""Avon_River""
        :name (z5 / name
            :op1 ""Avon""))
    :path (z6 / along)
    :duration (z7 / history))""

manually corrected AMR:
(i / intransitive-motion-91
    :ARG1 (p / person
        :ARG1-of (f / fame-01)
        :quant (q / multiple
            :op1 1000))
    :ARG2 (a / along
        :op1 (r / river
            :wiki ""Avon_River""
            :name (n / name
            :op1 ""Avon"")))
    :ARG3 (s / stroll-01
        :ARG0 p)
    :duration (h / history))
```

8.1.4. Ditransitive

```
I want you to update the AMR graph
```

```
to include the new roleset
ditransitive-91.

Ditransitive Construction is when
an Agent carries out an event that
is construed as causing a recipient
to receive a theme.

An example sentence with Ditransitive
Construction
They're going to kill Reagan a commie.

the auto generated graph is:
"(z0 / kill-01
    :ARG0 (z1 / they)
    :ARG1 (z2 / person
        :wiki "Ronald_Reagan"
        :name (z3 / name
            :op1 "Reagan")
        :mod (z4 / commie)))"

The issue with this is:
Kill.01
ARG0-PAG: killer
ARG1-PPT: corpse
ARG2-MNR: instrument
"They killed Ronald Reagan, who is a
commie" is the meaning we get,
which is incorrect.

If we were to fix this to represent
the actual meaning we would use
Ditransitive.91:
Arg0-Cause: They
Arg1-Theme: a commie
Arg2-Recipient: Reagan
Arg3-Concomitant-event: kill


Another Common Current AMR
Treatment (ok): Ditransitive

Sentence: Jack poured Jane an
arsenic-laced martini.

auto generated AMR:
"(z0 / pour-01
    :ARG0 (z1 / person
        :wiki -
        :name (z2 / name
            :op1 "Jack"))
    :ARG1 (z3 / martini
        :ARG1-of (z4 / lace-01
            :ARG2 (z5 / arsenic)))
    :ARG4 (z6 / person
        :wiki -
        :name (z7 / name
            :op1 "Jane")))"

The issue:
Pour.01
ARG0-PAG: agent, pourer
ARG1-PPT: liquid
ARG2-DIR: source
ARG3-GOL: destination
"Jack poured an arsenic laced
martini (?) Jane"  which is not
a great interpretation

we can fix this with Ditransitive.91:
Arg0-Cause: Jack
Arg1-Theme: an arsenic laced martini
Arg2-Recipient: Jane
Arg3-Concomitant-event: pour.01
(arg0: Jack, Arg1: martini)


Additional examples:

Sentence: Jack passed her the salt.

auto-amr:
"(z0 / pass-05
    :ARG0 (z1 / person
        :wiki -
        :name (z2 / name
            :op1 "Jack"))
    :ARG1 (z3 / salt)
    :ARG2 (z4 / she))"

manually corrected AMR:
(d / ditransitive-91
    :ARG0 (j / person
        :wiki -
        :name (n / name
            :op1 "Jack"))
    :ARG1 (s / salt)
    :ARG2 (s2 / she)
    :ARG3 (p / pass-05
        :ARG0 j
        :ARG1 s
        :ARG2 s2))


Sentence: I no longer think the
US Constitution AFFORDS me rights
as a citizen .

auto-amr:
"(z0 / think-01
    :ARG0 (z1 / i)
    :ARG1 (z2 / afford-02
        :ARG0 (z3 / law
        :wiki
        "United_States_Constitution"
        :name (z4 / name
            :op1 "US"
            :op2 "Constitution"))
        :ARG1 (z5 / right-05
            :ARG1 z1
            :prep-as (z6 / citizen))
        :ARG2 z1)
    :time (z7 / no-longer))"

manually corrected AMR:
"(t / think-01
```

```
    :ARG0 (i / i)
    :time(n2 / no-longer)
    :ARG1 (d / ditransitive-91
        :ARG0 (l / law
            :wiki
            "United_States_Constitution"
            :name (n / name
                :op1 "US"
                :op2 "Constitution"))
        :ARG1 (r / right-05
            :ARG1 i
            :prep-as (c / citizen))
        :ARG2 i
        :ARG3 (a / afford-02
            :ARG0 l
            :ARG1 r
            :ARG2 i)))"


Sentence: These skates were bought
for me by my mom.

auto-amr:
"(z0 / buy-01
    :ARG0 (z1 / person
        :ARG0-of (z2 /
        have-rel-role-91
            :ARG1 (z3 / i)
            :ARG2 (z4 / mom)))
    :ARG1 (z5 / skate
         :mod (z6 / this))
    :ARG4 z3)"

manually corrected AMR:
"(d / ditransitive-91
    :ARG0 (m / person
        :ARG0-of (h /
        have-rel-role-91
            :ARG1 (i / me)
            :ARG2 (m2 / mom)))
    :ARG1 (s / skates
        :mod (t / this))
    :ARG2 i
    :ARG3 (b / buy-01
        :ARG0 m
        :ARG1 s
        :ARG4 i))"
```

Appendix C: Cross-Linguistic Validity

Part of the motivation for our work was to capture constructions that occur cross-linguistically. This section will give a small taste of how this might be done for the Cuzco-Collao (QUZ) variety of Quechua, a language with complex, agglutinative morphology.

Quechua is particularly interesting because it has much more explicit Caused Motion constructions than English. In QUZ, the '-chi' affix indicates or delegates action and has an express causative function morphologically, and thus marks a causative construction. Combined in a sentence with the '-ta' object marker on the item being moved, it expresses the meaning that the action is causing something to happen to the patient of the sentence.

(1) *Chaymi rumita suchuchirqanku*
Chaymi rumi-ta suchu-chi-rqa-nku
therefore stone-ACC drag-CAUS-PST-3.PL
aya churasqanku
aya chura-sqa-nku
corpse place-Not.Experienced.PST-3PL
t'oqomanta.
t'oqo-manta
hole-ABL.from

'The stone was dragged from the hole where the corpse had been placed.'

We can see in the above example that the causative affix '-chi' attaches to the verb *suchuy* (to drag), while the accusative marker '-ta' attaches to the object *rumi* (stone). This sentence then more literally translates to, 'Therefore they caused the stone to be dragged from the hole where the corpse had been placed (I did not directly see this).'

The basic English representation of this AMR is shown in the following graph.

[1: English AMR] *The stone was dragged from the hole where the corpse had been placed.*

```
(c / caused-motion-91
    :ARG0 (t / they)
    :ARG1 (s / stone)
    :ARG2 (f / from
        :op1 (h / hole
            :location-of (p / place-01
                :ARG1 (c2 / corpse)))))
    :ARG3 (dr / drag-01
              :ARG0 t
              :ARG1 s))
```

Our approach provides a clear way of capturing the meaning in the causative '-chi' morpheme, as seen in the AMR drafted below. It provides a parallel in meaning with the English graph, as the graph structures can remain essentially the same, indicating that these two are semantically similar. The Caused Motion construction in particular would be easy to search for and capture in QUZ. The '-chi' morepheme maps naturally to caused-motion-91 construction. The subject affix on the verb containing '-chi' is then the :ARG0 (in this case *-nku* or 'they'). The :ARG1 is the object that comes before verb with the accusative marker '-ta' (*rumita*). The :ARG2 then can appear in a number of ways, but would generally be any ablative suffix such as '-manta' (*t'oqomanta*).

[2: Quechua AMR] *Chaymi rumita suchuchirqanku aya churasqanku t'oqomanta.*

```
(a / and
    :op1 (c / chi 'caused-motion-91'
        :ARG0 (t / nku 'they')
```

```
        :ARG1 (s / rumi 'stone')
        :ARG2 (f / manta 'from'
            :op1 (h / t'oqo 'hole'
                :location-of (p /
                place-01
                    :ARG1 (c2 /
                    corpse))))
        :ARG3 (d / suchu 'drag-01'
            :ARG0 t
            :ARG1 s))
    :op2 (s2 / sqa 'see-01'
        :polarity -
        :ARG0 (i / i)
        :ARG1 c2
        :manner (d2 / direct-02)))
```

The one addition made to the graph is the evidentiality, which is not present in the plain English graph. The '-sqa' affix is an evidential marker that lets one know that an event was not experienced by the speaker themselves and that the event happened in the past. English does not have grammaticalized evidentiality, and thus it was excluded from the plain English translation graph.

AMR was originally designed for English (and to some extent, Chinese) and therefore does not always directly capture semantic distinctions present in other languages. Thus, the next examples are intended to show how our non-lexically anchored rolesets could be applied beyond AMR to Uniform Meaning Representations (UMR), which is another graph-based semantic framework designed to represent meaning in a cross-linguistically applicable and computationally tractable way as an extension to AMR (Van Gysel et al., 2021).

(2) *Paytaq* *kawsay* *unuta*
Pay-taq kawsay unu-ta
3SG-FOC life water-ACC
qosunkiman.
qo-sunki-man
give-3SG>2SG-COND

'He could give you living water.'

The above sentence gives us an example of a Ditransitive sentence in QUZ. The simple English representation of the translation sentence is as follows:

[3: English AMR] *He could give you living water.*

```
(p / possible-01
    :ARG1 d / ditransitive-91
        :ARG0 (h / he)
        :ARG1 (w / water
            :mod living)
        :ARG2 (y / you)
        :ARG3 (g / give-01
                :ARG0 h
                :ARG1 w
                :ARG2 y))
```

Trying to extend this to Quechua, which has a more literal translation of 's/he could give you living water', results in the next graph:

[4: Quechua AMR] *Paytaq kawsay unuta qosunkiman.*

```
(p / possible-01
    :ARG1 d / ditransitive-91
        :ARG0 (p / pay 's/he')
        :ARG1 (u / unu 'water'
            :mod kawsay 'living')
        :ARG2 (n / nki 'you')
        :ARG3 (g / give-01
                :ARG0 p
                :ARG1 u
                :ARG2 n))
```

A notable issue with Graph 4, and Graph 2 for that matter, is that evidentiality is not a concept well captured in AMR representations, but is grammatically expressed in Quechua and, thus, is essential to any accurate semantic representation of its sentences. The `:modstr` value provides a solution to this issue in UMR. Additionally, the idea of genderless 3rd person singular is lost in the English AMR, but recaptured in Graph 5 within UMR. UMR also more accurately depicts verbs in languages that lack a standard PropBank roleset, as seen for the verb 'qoy-00'. For this reason, general CxGr-AMR rolesets would be especially helpful in languages that lack lexical resources or quality mappings to lexical resources in English, as the CxGr-AMR roleset can be used to capture meaning in lieu of costly lexicon development.

[5: Quechua UMR] *Paytaq kawsay unuta qosunkiman.*

```
(d / ditransitive-91
    :ARG0 (p / pay 'person'
        :ref-person 3rd
        :ref-number Singular)
  :ARG1 (u / unu 'water'
        :mod kawsay 'living')
  :ARG2 (s / sunki 'you'
        :ref-person 2nd
        :ref-number Singular)
  :ARG3 (q / qoy-00 'give'
        :actor p
        :theme u
        :recipient s
        :modstr NeutAff
        :aspect Performance))
```

Graph 5 demonstrates that the Ditransitive construction transcends English and can be utilized to accurately depict the semantic relations in QUZ. Our approach could be easily integrated into the UMR schema to more accurately capture cross-linguistic realizations of these types of non-lexically anchored constructions.

Adding Aspectual Information to Structured Meaning Representations

Claire Benét Post,[1] **Paul Bontempo,**[1] **August Milliken,**[1]
Nicholas Derby, Saksham Khatwani, Sumeyye Nabieva,
Alvin Po-Chun Chen, Karthik Sairam, Alexis Palmer

University of Colorado Boulder
{benet.post, paul.bontempo, august.milliken, alexis.palmer}@colorado.edu

Abstract

To fully capture the meaning of a sentence, semantic representations should encode *aspect*, which describes the internal temporal structure of events. In graph-based meaning representation frameworks such as Uniform Meaning Representations (UMR), aspect expresses how events unfold over time, including distinctions such as states, activities, and completed events. Despite its importance, aspect remains sparsely annotated across semantic meaning representation frameworks, hindering not only current manual annotation, but also the development of automatic systems capable of predicting aspectual information. In this paper, we introduce a new dataset of English sentences annotated with UMR aspect labels over Abstract Meaning Representation (AMR) graphs. We describe the annotation scheme and guidelines used to label eventive predicates according to the UMR aspect lattice, as well as the annotation pipeline used to ensure consistency and quality across annotators through a multi-step adjudication process. To demonstrate the utility of the dataset for future automation, we perform simple baseline experiments using three modeling approaches. Our results establish initial benchmarks for automatic UMR aspect prediction and provide a foundation for integrating aspect into semantic meaning representations more broadly.

Keywords: Aspect annotation, semantic meaning representations, aspectual generation benchmarks

1. Introduction

Semantic representations frequently center around capturing components of meaning related to the core *events* conveyed by individual natural language utterances. Nearly all meaning representation (MR) formats express the core predicates associated with those events, along with any arguments to those predicates. MRs differ quite substantially, though, when it comes to the expression of additional event information, such as tense, modality, aspect, or information structure. Languages also differ substantially in the degree to which they grammaticalize (or require the expression of) the same event-related information.

Aspect is a core component of Uniform Meaning Representation (UMR),[1] a graph-based semantic framework designed to represent meaning in a cross-linguistically applicable and computationally tractable way. UMR is an extension and modification of the Abstract Meaning Representation (AMR) framework (Banarescu et al., 2013), an MR which nicely suits the typological properties of English but which begins to strain when adapted for other languages. Like AMR, UMR is a graph-based framework encoded in Penman-style notation (Wein and Bonn, 2023). Unlike AMR, UMR was designed by and with linguistic typologists, in order to build an approach to annotation suitable for diverse languages (Van Gysel et al., 2021).

```
She is still writing her paper.
(w/ write-01
    :ARG0 (p/ person
        :ref-person 3rd
        :ref-number Singular)
    :ARG1 (p2/ paper
        :poss p
        :ref-number Singular)
    :mod (s/ still)
    :aspect Activity
    :modstr FullAff)
```

Figure 1: Example UMR graph with the eventive predicate `write`. This graph captures predicate argument structure, properties of the arguments, and event-related properties. We highlight parts of the graph relevant for the aspect label *Activity*.

Unlike tense, which encodes *when* an event occurs, aspect captures the *how*: the internal temporal structure, duration, and completedness of events (Comrie, 1976; Croft, 2012; Donatelli et al., 2018). It allows a semantic system to distinguish between, for example, habitual, ongoing processes, or completed achievements, enabling a more nuanced interpretation of event semantics.

In UMR, aspect is applied to all *eventive* elements (also known as *eventualities*) in a sentence. The central eventuality introduced by an utterance is typically the concept aligned with the main finite

[1] Equal contribution.

[1] https://umr4nlp.github.io/web/

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 20–36

verb, as seen in Figure 1. Eventualities in UMR refer to the full predication, encompassing the verb and its arguments (Donatelli et al., 2018; Kingsbury and Palmer, 2003). UMR defines a particular inventory of aspectual categories, following Croft (2022), aligned with other well-established event typologies, and including (among others) states, activities, accomplishments, achievements, and processes (Bach, 1986). In UMR the aspectual categories are organized into a lattice that supports both coarse- and fine-grained aspectual distinctions. Unlike surface grammatical cues, like those in auxiliaries or verb morphology, UMR aspect is a semantic feature. It abstracts away from morphosyntactic form to represent covert event structure and is intended to generalize across typologically diverse languages (Van Gysel et al., 2021).

Annotating aspect is no simple feat. Theoretical debates span decades, including disagreements about the universality of aspectual categories, the granularity of classifications, and their interaction with tense and modality (Reichenbach, 1947; Vendler, 1957; Comrie, 1976; Langacker, 2011; Dowty, 1986; Hinrichs, 1986; Moens and Steedman, 1988; Klein, 2013; Chang et al., 2022; Partee, 2011; Croft, 2012). While there are a number of corpora with aspect annotations, and several computational models (Friedrich et al., 2023), there is no unified approach to annotation or modeling (among others: Pustejovsky et al., 2003; Derczynski, 2017; Pustejovsky et al., 2017; **?**; Friedrich et al., 2016; Mostafazadeh et al., 2016; Laparra et al., 2018; O'Gorman et al., 2016; Gantt et al., 2022).

From a typological perspective, some languages encode aspect more saliently than others, further complicating annotation for multilingual or cross-linguistic frameworks. For example, American Sign Language and Mandarin Chinese prioritize aspectual distinctions over tense (Li and Thompson, 1989; McDonald, 1982), while Hindi includes a dedicated aspect morpheme separate from tense or mood (Van Olphen, 1975). In contrast, many Indo-European languages conflate aspect and tense morphologically, often obscuring the underlying semantic distinctions. The categories in and structure of the UMR aspect lattice are flexible enough to precisely encode aspectual distinctions as seen in each of these languages.

Given these complexities, manual aspect annotation is time-consuming, error-prone, and highly sensitive to annotator interpretation. Yet its inclusion in UMR is a cornerstone to achieving a more expressive, cross-linguistic meaning representation system. UMR builds on earlier formalisms such as Abstract Meaning Representation (AMR) (Banarescu et al., 2013), where aspect was initially introduced to support event-based reasoning but was never fully adopted into standard annotation guidelines. Donatelli et al. (2018) formalize aspect in AMR, laying out annotation principles and aligning event types with lexical frames.

Despite its importance for accurately representing event semantics, aspect remains underannotated in existing UMR resources. This scarcity limits both the scale and consistency of manual UMR annotation and hinders development of automatic parsers capable of reliably predicting aspect.

To address this bottleneck, we present a new dataset of English sentences manually annotated with UMR aspect labels. The study presented here is part of an ongoing effort to build a large-scale English UMR dataset by converting existing AMR graphs to UMR representations for the same sentences Bonn et al. (2023). The conversion is a multistep process, combining automated processing and manual intervention. The AMR graphs which are the foundation of that conversion effort do not include any aspect annotations, and the new dataset is intended to support development of automated aspect annotation to fill in aspect values for the remaining AMR graphs. Under our approach, annotators are provided with AMR-derived graphs for these same sentences sentence and asked to label eventualities according to the UMR aspect lattice (see Section 4.1 for corpus details).

Our annotation pipeline combines structured annotator training with multiple rounds of independent annotation, group adjudication sessions, and expert consultation to improve the guidelines and resolve difficult annotation decisions, resulting in a high-quality resource intended to support future UMR modeling efforts. To validate dataset quality, we additionally report baseline experiments spanning three modeling families: (1) a rule-based approach, (2) an embedding-based classifier, and (3) a large language model (LLM) prompting approach.

We target two complementary objectives:

- **Task 1: Data annotation.** We construct a new gold-standard dataset of 1473 carefully validated sentences labeled according to the UMR aspect annotation scheme.
- **Task 2: Baseline modeling.** We present standard data splits and initial performance benchmarks for automated UMR aspect labeling.

This is the first dataset designed to support supervised learning for UMR aspect labeling.[2]

2. Related Work

The semantics of aspect has been a long-standing topic of debate in linguistic theory. Seminal works

[2]All data and labels available at: https://github.com/clairepost/UMR_Aspect_Data.git

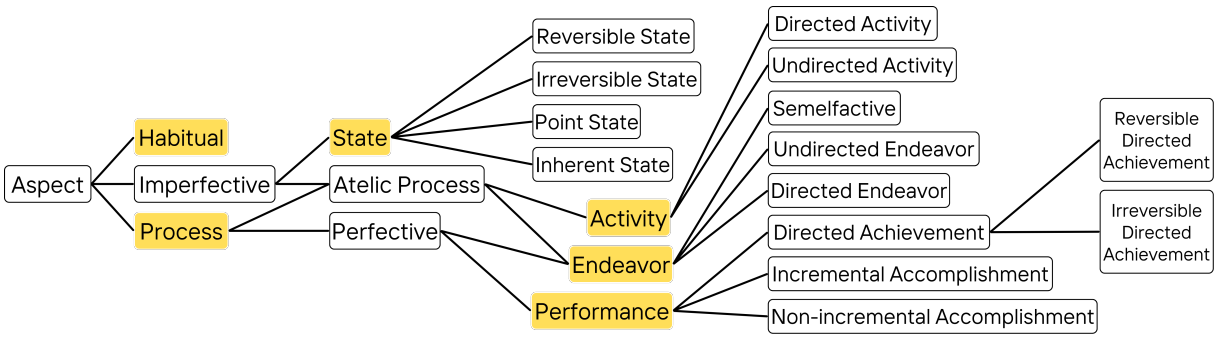


Figure 2: UMR aspect lattice with aspectual values utilized in our English annotation highlighted in yellow.

by Reichenbach (1947), Vendler (1957), and Comrie (1976) lay the foundation for distinguishing between types of eventualities–states, achievements, activities, accomplishments–based on their temporal and structural properties. Dowty (1986) and Langacker (2011) further explore the interaction between aspect, argument structure, and lexical semantics. These formalisms inform how events are modeled in UMR today.

Later developments such as Hinrichs' interval-based models (1986), Moens and Steedman's narrative structure theory (1988), and Klein's temporal logic (2013) introduce more formal ways to encode event structure and its temporal entailments. These insights highlight the need for meaning representation frameworks like UMR to go beyond grammatical tense and directly encode aspectual distinctions based on semantic content.

Aspect annotation. Aspect has been incorporated into previous semantic annotation and event modeling efforts, particularly in temporal information extraction. TimeML (Pustejovsky et al., 2003) and its follow-up projects such as the TempEval competitions (Derczynski, 2017) include annotation for aspect, typically via shallow textual cues. There are several datasets developed with robust manual aspect annotation that consider sentential context and event structure, such as DIASPORA (Kober et al., 2020) and the Universal Decompositional Semantics dataset (Gantt et al., 2022); these datasets employ coarse-grained aspect classes, rather than the more extensive and typologically-broad lattice provided by the UMR framework. Other recent work seeks to automate aspect classification using linguistic features (**?**), discourse roles (Friedrich et al., 2016), and LSTM-based models that integrate context (Mostafazadeh et al., 2016; Laparra et al., 2018).

While effective to some degree, these systems often operate over flat text or shallow syntactic representations. They do not handle the rich predicate-argument structures or graph-based semantics found in UMR and AMR. Moreover, they treat aspect as a kind of downstream feature, rather than an integral part of event structure representation.

Donatelli et al. (2018) develop an approach for adding feature-based tense and aspect information to AMR graphs. For aspect, the scheme encodes four crucial semantic features: `+/-stable`, `+/-ongoing`, `+/-complete`, and `+/-habitual`. Many aspectual categories can be derived from the combination of semantic feature values. Under the UMR scheme, however, aspectual categories are to be labeled directly rather than broken down into features.

Automatic aspect annotation for UMR. Due to the small amount of available UMR data, prior work has focused primarily on methods for generating UMR graphs without supervised training. Chun and Xue (2024) propose a multi-step strategy for converting AMR graphs into UMR graphs, including the addition of aspect. The approach derives aspect from `Tense` and `VerbForm` features output by UDPipe v2 (Straka, 2018). Similarly, Sun et al. (2024) experiment with few-shot and Think-Aloud prompting on LLMs to generate Chinese UMR graphs, including aspect labels.

AutoAspect, which directly targets UMR aspect, proposes a branching rule-based approach specifically for classifying UMR aspects in English UMR graphs (Chen et al., 2021). AutoAspect delivers high precision for two aspectual categories and variable results for others. We aim to build a larger aspect-labeled dataset to support a broader range of learning approaches. We further note that significant refinements have been made to the UMR dataset since this system was published, as seen in (Bonn et al., 2024a).

3. Annotation Scheme

3.1. UMR Aspect Lattice

This paper is concerned with the aspect annotation of English sentences, as highlighted in Figure 2. UMR organizes aspectual categories within an as-

pect lattice, a hierarchical structure that captures relationships between coarse and fine-grained labels. This design allows annotations to represent general distinctions while remaining compatible with more specific readings when additional linguistic information is available.

This structure is particularly useful for cross-linguistic semantic annotation. English often relies on relatively coarse-grained aspectual distinctions, while other languages encode finer aspectual contrasts directly in their grammar. The lattice enables UMR to support consistent representations across typologically diverse languages. This was a major motivation for our work, and we hope in the future to expand aspect annotation to more languages.

The aspectual categories chosen for English annotation include a set of base-level distinctions–*State*, *Performance*, *Endeavor*, *Activity*, and *Habitual*, and *Habitual*–as well as a more coarse-grained value for event nominals and other underspecified events, *Process*.

3.2. Aspect Types

State. This value corresponds to stative events, indicating that no change occurs during the event, as prescribed by (Vendler, 1967). It includes predicate nominals, predicate locations, and thetic (presentational) possession.

[1] *The cat loves milk*.

```
(l/ love-01
    :ARG0 (c/ cat
        :ref-number Singular)
    :ARG1 (m/ milk)
    :aspect State
    :modstr FullAff)
```

More specifically, in English, the *State* value encompasses modal verbs (e.g., "The cat **needs** to eat.") and events under the scope of ability modals (e.g., "The cat is **able** to eat."). UMR classifies *inactive actions*, as defined by (Croft, 2012), as stative. This includes posture verbs (e.g., "The cat **hangs** on the windowsill."), perception verbs (e.g., "The cat **sees** milk."), mental activities (e.g., "The cat **thinks** about jazz."), verbs of operation (e.g., "The cat is **working** on catching mice."). The *State* value, in English, is also an umbrella that covers inherent states (e.g., "The cat **is** black."), reversible states (e.g., "The cat is **hungry**."), irreversible states (e.g., "The glass **is shattered**."), and point states (e.g., "When it **is** 12:30pm, feed the cat.").

Performance. This category covers events that reach a result state, such as achievements that have some instantaneous binary change, accomplishments where there is a run-up process before the change, or when the event reaches a result state that has a natural endpoint.

[2] *The cat walked along the fence in 2 minutes.*

```
(w/ walk-01
    :ARG0 (c/ cat
        :ref-number Singular)
    :ARG2 (a/ along
        :op1 (f/ fence))
    :duration (t/ temporal-quantity
        :unit (m/ minute)
        :quant 2)
    :aspect Performance
    :modstr FullAff)
```

For instance, completive markers (e.g., "The cat finished **climbing** *up the tree*.") and container adverbials (e.g., "The cats **scampered** along the fence *in 10 seconds*.") are both indicators that an event has reached a distinct result state. Note that the temporal expression "in two minutes" appears in the graph under the attribute `:duration`.

Endeavor. *Endeavor* is used for processes that end within the time window in question, but do *not* reach a particular result state: e.g., compare graph [2] to graph [3] below.

[3] *The cat walked along the fence.*

```
(w/ walk-01
    :ARG0 (c/ cat
        :ref-number Singular)
    :ARG2 (a/ along
        :op1 (f/ fence))
    :aspect Endeavor
    :modstr FullAff)
```

The *Endeavor* value is often mistaken for *Performance* and vice versa. Often, in English predicates have explicit aspectual marking to be considered an *Endeavor*. Terminative aspectual markers, like "stop" in English, and durative adverbials (e.g., "The cat **ate** kibble *for thirty seconds*.") are both strong indicators for *Endeavor*.

Activity. The *Activity* aspect covers processes that do not start or end during the time window in question. They can be ongoing with respect to present or past time (e.g., “The cat **was playing** the piano.”) For an example graph, see Figure 1.

Identifying *Activity* is difficult because it is largely dependent upon context, document creation time, and real world knowledge. However, there are some grammatical clues that can help. For example, if the event is in the present progressive (e.g., "The cat **is playing** the piano."), it is typically annotated with *Activity*. Inceptive and continuative aspectual markers may also imply that an event has not ended (e.g., "The cat **started playing** the piano." and "The cat **kept on playing** the piano."). In UMR, iterative events are labeled *Undirected Activity*; they fit under the *Activity* umbrella for us.

Habitual. The *Habitual* aspectual sense is usually straightforward to identify. It covers things that happen repeatedly or regularly. In English, adverbials such as “used to” and “always” often (but not always) modify the verb in habitual events.

[4] *The cat eats kibble.*

```
(w/ eat-01
    :ARG0 (c/ cat
        :ref-number Singular)
    :ARG1 (k/ kibble)
    :aspect Habitual
    :modstr FullAff)
```

In English, *Habitual* is often expressed through simple present tense, while habituals in the past are often indicated with "used to" (e.g., "The cat **used to eat** kibble.").

Process. Of the labels we use for English, *Process* is the most coarse-grained. It describes an ongoing event where the beginning or end is uncertain or unspecified. The most common use of *Process* in our dataset is as the default label for event nominalizations (e.g., "The cat denied **wrongdoing**.").

[5] *After the game, the cat slept.*

```
(s/ sleep-01
    :ARG0 (c/ cat
        :ref-number Singular)
    :temporal (a/ after
        :op1 (g/ game
            :aspect Process))
    :aspect State
    :modstr FullAff)
```

Graph [5] shows another category of events typically annotated with *Process* in UMR. Here, the **game** event is packaged in a referring expression, and the prepositional phrase appears under the attribute *:temporal*. We take a similar approach for underived nominals, nominalizations, and gerunds.

None. For our annotations, we additionally allow annotators to apply the label *NONE* for predicates that they deem to be non-eventive, even though the appear in the graph as a semantic predicate. This happens frequently for adjectival and adverbial concepts, which do not participate in eventualities. These show up as predicates because they often receive automatic mappings into FrameNet predicates in AMR (Baker et al., 1998); these are then carried over into the UMR versions of the graphs.

3.3. Comparison to Other Aspect Annotation Schemata

Aspect annotation has been widely studied in linguistics, and UMR is a recent schema that builds on prior theoretical work. In particular, UMR follows approaches such as Croft (2012), which emphasize that aspectual interpretation depends on multiple factors and that a single event may admit multiple plausible aspectual readings depending on context. This perspective informed our adjudication process when resolving difficult cases.

Because our task is motivated by downstream NLP and machine learning applications, our annotation process follows principles outlined by Pustejovsky et al. (2017). We prioritize consistency in label assignment in order to maximize the learnable signal in the dataset, even when this means limiting the number of annotated examples. Nevertheless, as shown in Figure 2, the selected aspect labels differ in granularity, which introduces variation in specificity across the dataset.

Prior work such as the DIASPORA dataset (Kober et al., 2020) also explores aspect annotation but employs a more coarse-grained 3-label schema (*state*, *telic*, *atelic*) to reduce label overlap and maintain uniform granularity. Our dataset instead adopts the richer UMR aspect inventory while remaining compatible with prior work. To illustrate this compatibility, we developed a mapping between our schema and the DIASPORA labels and automatically applied it to a subset of our data, manually evaluating the resulting label assignments. We found that the two schemata are broadly compatible, excepting the case of event nominals, which take *process* labels in UMR but are without a consistent equivalent in DIASPORA, requiring manual adjudication rather than automated label mapping.

4. Building the Corpus

4.1. Data

Our dataset is sourced from the UMR 2.0 Dataset (Bonn et al., 2025) which contains roughly 30,000 UMR graphs in different stages of conversion from AMR graphs (Knight et al., 2020; Bonn et al., 2020).[3] Some UMR 2.0 graphs have aspect annotations from previous work; we only annotate graphs that do not yet have aspect labels. To ensure broad coverage for training and evaluation, we select four corpora from the dataset to annotate:

1. The Little Prince corpus, a set of sentences from the English translation of *The Little Prince* by Antoine de Saint-Expupéry.

2. The Minecraft corpus, a set of dialogues and corresponding grounding data from a collaborative structure-building task in Minecraft (Narayan-Chen et al., 2019).

[3]We keep all graphs in the original format for aspect labeling; we make no structural modifications or revisions to the data.

3. The BOLT DF corpus, which contains English language forum posts crawled as part of the DARPA BOLT project.

4. The Weblog corpus, comprised of weblog posts and online news articles.

A detailed summary of aspect label statistics, by corpus, for the existing UMR dataset can be found in Appendix A as Table 5.

4.2. Annotation

Annotation proceeded in two phases: in PHASE 1, a team of 8 annotators worked in pairs to label each event marked in the graphs. In PHASE 2, a smaller group focused on adjudicating decision ties with expert consultation, while ensuring consistency with previous annotations.

4.2.1. Phase 1: Initial Annotation

Given the complexity of aspect annotation and its theoretical underpinnings, we first focused on helping all members of the team develop a strong understanding of the UMR aspect schema, before proceeding to the bulk annotation of PHASE 1. Our goals were to ensure that each annotator contributed quality data and that rules were applied consistently between annotators. To this end, we conducted 8 weekly training sessions throughout PHASE 1.

Annotation guidelines and training materials were built from existing UMR resources and developed into task-specific training materials.[4] Each week, team members presented on different topics from these materials and discussed example annotations as a group to clarify issues.[5]

We conducted an initial practice task in which each team member annotated up to 50 predicates from the Pear Story corpus (Bonn et al., 2023). This dataset was selected for its short, visually grounded sentences, which aided learning and facilitated discussion. We reviewed inter-annotator agreement and recurring errors on the practice task before proceeding to bulk annotation.

Label-wise accuracy measured over the Pear Story annotations showed that the categories *State* and *Performance* were quite reliably and consistently identified, while minority classes like *Endeavor* and *Habitual* were less consistent. These findings prompted us to hold a focused error correction and continued training session, in which we reviewed common sources of confusion, such as distinctions between *State* vs. *Performance* and *Performance* vs. *Endeavor*.

We next moved into full-scale annotation. Each corpus was assigned to two annotators for independent labeling, resulting in two first-pass labels per event per corpus. Each numbered predicate within the AMR graphs was annotated with one of the six UMR aspect labels or marked with *NONE* if the predicate was deemed non-eventive. Table 1 shows the distribution of aspect labels for the completed dataset. The first-pass bulk annotation lasted approximately 6 weeks.

4.2.2. Phase 2: Tie-breaking and Adjudication

Following PHASE 1, all events with conflicting aspect labels were routed to a tie-breaking process. Each sentence and its annotations were reviewed by a third annotator to make a final determination. If the adjudicating annotator disagreed with both original labels, the sentence went to an additional adjudication step, up to a maximum of 5 total annotation rounds. All intermediate labels are preserved and ranked in our final dataset, along with the final adjudicated labels.

In the next step, a team of two annotators reviewed all data for consistency. Together, the tie-breaking and consistency adjudication lasted about 8 weeks. This review process ensured each adjudicated aspect label: (i) follows our annotation guidelines, and (ii) is consistent with other instances in the dataset. To confirm consistency of a given label, we compare difficult cases to sentences with the same event *and* to events with the same label.

The duration and complexity of this annotation process indicates the corresponding complexity of aspect itself; even with months of discussion, some instances in the dataset remained ambiguous. For particularly complex disagreements—such as differentiating *Endeavor* from *Performance*—we consulted directly with external experts to align the annotations with their interpretations.

4.3. Inter-Annotator Agreement

We report agreement metrics across the various phases of our annotation and adjudication process, compared against our finalized gold-level labels in Table 1. Overall, these metrics indicate sufficient consensus for further analysis using our dataset.

Per-class Krippendorff's alpha. These values, computed for each aspect label class against all other classes combined, measure the agreement across annotators from zero (random chance) to 1 (perfect agreement), with a standard significance threshold of 0.67 (Krippendorff, 2004).

[4] See Appendix B for more details.

[5] Resources from these meetings are on GitHub: https://github.com/clairepost/UMR_Aspect_Data.git.

Per-class		
Label	$K\alpha*$	**Count**
None	0.735	514
State	0.706	385
Performance	0.716	360
Process	**0.547**	100
Habitual	0.704	56
Activity	**0.502**	40
Endeavor	**0.559**	18
Total		**1,473**
First-pass		
Percent agreement		74.1%
Cohen's κ		0.656

Table 1: Inter-annotator agreement metrics; boldface numbers indicate low agreement. *Per-class Krippendorff's alpha computed as one-vs-rest.

Gold label	Error rate	Error Label
Activity	6.7%	Performance
Endeavor	12.2%	Performance
Habitual	6.8%	Performance
None	6.0%	State
Performance	3.9%	None
Process	9.1%	None
State	4.8%	None

Table 2: Final gold aspect labels with the error rate compared to the final adjudication. 'Error Label' is the most common incorrect annotation.

First-pass metrics. Percent agreement represents the proportion of events where both first-pass annotators provided the same label for an event, though this does not necessarily mean that such events were exempt from adjudication later. Cohen's κ measures agreement on the same scale as Krippendorff's α, but strictly between two annotators, so we report the κ score for the two first-pass annotators, averaged across all 4 corpora.

5. Annotation Challenges

Table 5 reports the most frequent erroneous labels applied during our annotation process, with the associated gold-standard label in the left column. The results illustrate that certain types of events are particularly difficult to disambiguate, even for experts. In Table 3 we present representative examples of particularly challenging annotation cases. Many of these issues stem from the limited contextual information available to annotators, as sentences were presented individually rather than as part of complete documents. In naturally occurring discourse, surrounding context typically resolves such ambiguities, and aspectual interpretation is no exception.

For instance, in Table 3 example (a), the event suggests a *Process* label in the sentence context, since the speaker specifies no explicit number of blocks, and the event lacks an inherent endpoint. However, the wider discourse context (blocksworld video game) includes a particular number of red blocks available to the players, and a finite grid on which to place them, which instead motivates a *Performance* label.

Similarly, in example (b), contextual cues could determine whether an event is an *Endeavor* or a *Performance*, depending on whether we understand "clap" as a process without change of state (hands end up as they started), or a complete event that reaches a natural conclusion (hands start apart and end together in a single motion). Without surrounding document context, we cannot say for sure which type of clap this sentence describes.

To account for multiple plausible interpretations, our adjudication schema allows for the identification of a secondary label that reflects a reasonable alternative reading, even when a primary label is selected based on the most likely interpretation. For events without such ambiguity such as example (c), we provide only one adjudicated label.

6. Automatic Annotation

One goal of this annotation effort is to build training and evaluation data for automatic aspect classification. Toward that end, we establish standard data splits and evaluate the performance of several simple baseline models, leaving development of more sophisticated models for near future work.

6.1. Data splits

We establish standardized splits of our final dataset in a 70/15/15 train/val/test ratio, as illustrated in Table 7, found in Appendix C. We split the data on sentence boundaries, rather than by event, to ensure that no contextual information is shared across splits. This is necessary because a single sentence may include multiple events, and sentential context seen in training for one event could unfairly inform test performance on another. We stratified each split to ensure consistent and balanced class distribution.[6] To accomplish this, we first identify a dominant aspect class for each sentence, by counting the most frequent label across all events per sentence. For sentences with just one event or whose events all have different labels, we consider the first event's label to be dominant. We perform an initial 70/30 split of the sentences, keeping the label distribution of both consistent with those of the overall dataset. We do the same for the secondary

[6] Exact data splits are available on our GitHub.

Sentence	Event	Main label	2nd label
(a) <Architect> now above that place red blocks on the grid.	Place	Performance	Process
(b) "Clap your hands, one against the other," the conceited man now directed him.	Clap	Endeavor	Performance
(c) Greed is when you are wealthy and lobby your representatives for special tax breaks because you are over 60 years of age.	Age	State	N/A

Table 3: Selected examples with ambiguous aspect; some annotated with secondary labels.

division on the 30% split, producing comparable validation and test splits.

6.2. Baseline models

To establish a performance baseline for this task on this data set, we test on two types of models: 1) LLMs (both closed and open-source) under a simple prompting paradigm; and 2) a simple feedforward neural architecture.

LLM Prompting. We experiment with multiple LLMs in a prompting paradigm to evaluate their capability for aspect classification without fine-tuning. LLM performance on structured prediction tasks has been shown to vary drastically based on slight changes to prompt structure (Lu et al., 2022), and other work suggests that LLMs lack meta-linguistic reasoning capability (Bonn et al., 2024b); we examine the ability of LLMs to identify covert aspectual information from a sentence, as well as produce a baseline against which to compare other neural approaches in the future. Although finding the optimal prompt for this task is intractable, we first ran a preliminary search across different prompt strategies on a validation set to determine if any of them boosts aspect prediction performance significantly.[7] We found minimal differences between prompt styles, and proceeded to the test phase. Our tests compare `Llama-3.1-8b-instruct` (Grattafiori et al., 2024) and `GPT-5mini`. We experiment with few-shot in-context learning using 3 examples per label (21 examples per prompt).

Feedforward Classifier. We investigate the ability of LLM encoder layers to capture representations that may be useful for aspect classification based on the hypothesis that contextual embeddings encode a broad range of linguistic phenomena (Arora et al., 2024). We do this by combining the token embeddings of the input sentence with a simple feedforward neural classifier to produce a label prediction from the text alone via supervised training.

[7] Details of the prompt-tuning experiments available in Appendix C.

To evaluate the usefulness of LLM embeddings out-of-the-box, we pass the natural language sentence through the encoder block of Llama-3.1 8B and average the resulting token embeddings to generate a sentence vector with standardized dimensions, then use a simple feedforward classifier head to produce a label prediction. We use the same averaged embedding as input for each event in the sentence. We train a fully-connected feedforward network to predict one of the seven aspect labels using that embedding as input. The results from this method serve as a useful benchmark for evaluating more complex strategies in future work.

6.3. Automatic Modeling Results

Table 4 displays the accuracies and F1-scores across the two LLMs and the feedforward neural classifier. We report weighted F1, average precision, and average recall; to address the imbalanced label distribution in the data, we also report macro F1. We evaluate all methods on the same stratified test set. The table also shows reported results for AutoAspect (Chen et al., 2021), a rule-based approach to UMR aspect classification. Note that the reported results use a different test set, so these serve as a general reference for rule-based approaches rather than a direct comparison. Finally, we show annotation agreement scores as an upper bound for the task.

LLM Prompting. The dataset's significant imbalance, with *State* and *NONE* labels being dominant and many classes being rare, heavily influences the results. Weighted F1 scores are skewed by the majority class, while lower macro F1 scores accurately reflect poor performance across most categories. GPT-5mini outperforms Llama across all metrics, which we attribute to architectural updates, including advanced knowledge distillation. GPT-5mini performance is relatively indifferent to in-context examples, while Llama's performance actually decreases with the addition of in-context learning, suggesting that more comprehensive training is needed for aspect prediction.

Type	Model	Acc.	Macro F1	Wtd. F1	Precision	Recall
Upper Bound	Single Human Annotator	0.84	0.76	0.84	0.77	0.82
LLM	LLaMA-3.1-8B-Instruct (zero-shot)	0.31	0.19	0.27	0.29	0.24
	LLaMA-3.1-8B-Instruct (3-shot)	0.25	0.16	0.22	0.32	0.21
	GPT-5mini (zero-shot)	**0.56**	**0.49**	0.56	**0.69**	**0.49**
	GPT-5mini (3-shot)	**0.56**	0.46	**0.60**	0.49	0.46
Neural	Feedforward MLP	0.45	0.27	0.44	0.29	0.32
Symbolic	AutoAspect[†]	0.39	0.23	0.40	—	—

Table 4: Baseline results on the test split (254 events, 72 sentences). Human performance reflects first-pass annotator accuracy against adjudicated gold labels as an upper bound on performance, against which to measure automated method results. Precision and Recall are macro averages across classes.

Feedforward Classifier. Neural classification using Llama embeddings results in middling performance, coming short in all three evaluation metrics compared to LLM prompting methods. Although sentence embeddings have been seen to capture semantic information in other tasks, these results demonstrate that embeddings alone are insufficient for capturing aspectual information.

Human Baseline. The human baseline scores show agreement between one annotator's first-pass labels for each of the events in the test set, compared against the final adjudicated labels. The success of the human baseline over the automated methods supports two conclusions: (i), the complexity of the aspect annotation task, and (ii), the need for automated methods which better utilize the sentential context and/or the inherent graphical nature of event-argument structures. We are currently developing aspectual classification models that learn from the graph structure as well as the surface form of the utterance.

7. Conclusion and Future Work

In this work, we introduce a new, carefully-annotated dataset of 1473 English sentences annotated with aspect labels within the UMR framework, achieving good agreement between annotators. We describe (and release) the annotation scheme and guidelines and detail our multi-stage annotation and adjudication process. Analysis of annotator disagreements follow expected patterns with respect to the confusability of same label pairs, motivating us to allow (and preserve) multiple labels per instance.

On the modeling side, we establish straightforward baselines for automated aspect classification using rule-based methods, embedding-based classifiers, and large language model prompting approaches. These results provide initial benchmarks for automatic UMR aspect classification, and we expect to see significant increase in model performance when we turn to more sophisticated architectures. The guidelines, dataset, stratified data splits, and initial benchmarks together lay a foundation for studying aspect in structured semantic representations and will support future work on automated UMR parsing and cross-linguistic semantic annotation.

Ethical Considerations

This work builds on existing publicly available corpora that were previously released for research purposes. Our dataset adds aspectual annotations to sentences drawn from these sources in accordance with their respective licenses. Annotation was conducted by trained researchers who are authors of this paper.

Because the dataset focuses on English sentences, it representatively only reflects information about English-language corpora and does not directly capture aspectual distinctions present in other languages. Future work will expand this annotation framework to additional languages in order to support broader cross-linguistic semantic analysis.

We do not anticipate significant risks of misuse for this dataset. However, some of the examples in our corpus were pulled from online message fora without censoring, and may contain offensive, explicit, or harmful language. The resource is intended to support research in semantic representation and natural language processing.

Limitations

We attempted to reimplement the AutoAspect rules-based classifier (Chen et al., 2021) on our novel set of annotated UMR graphs in order to compare its performance against the neural approaches as a benchmark. AutoAspect focuses on a structured set of rules which closely followed the UMR annotation guidelines and decision lattice to predict labels in a wholly deterministic method, without machine learning. However, due to dependency

issues with the semantic parser in the original AutoAspect codebase, we are unable to report this benchmark on our dataset, and instead provide the AutoAspect classifier's performance on the dataset with which it was published, as a reference for rule-based approaches in general.

Acknowledgments

This work is supported by a grant from the CNS Division of National Science Foundation (Award Number: NSF_2213805) entitled "Building a Broad Infrastructure for Uniform Meaning Representations." We extend our sincere gratitude to Julia Bonn for her invaluable insights and suggestions on the adjudication of aspectual decisions, as well as her learning materials on UMR. Thanks also to Bill Croft for his insights into edge cases on the UMR aspect lattice and other helpful advice given. Lastly, we express our appreciation to the reviewers for their helpful comments and feedback.

8. Bibliographical References

Aryaman Arora, Dan Jurafsky, and Christopher Potts. 2024. CausalGym: Benchmarking causal interpretability methods on linguistic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14638–14663, Bangkok, Thailand. Association for Computational Linguistics.

Emmon Bach. 1986. The algebra of events. *Linguistics and philosophy*, pages 5–16.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley FrameNet Project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Julia Bonn, Matthew J Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajic, Kenneth Lai, James H Martin, et al. 2024a. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547.

Julia Bonn, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. 2023. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C. Association for Computational Linguistics.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Julia Bonn, Harish Tayyar Madabushi, Jena D. Hwang, and Claire Bonial. 2024b. Adjudicating LLMs as PropBank adjudicators. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 112–123, Torino, Italia. ELRA and ICCL.

Nancy Chang, Daniel Gildea, and Srini Narayanan. 2022. A dynamic model of aspectual composition. In *Proceedings of the twentieth annual conference of the cognitive science society*, pages 226–231. Routledge.

Daniel Chen, Martha Palmer, and Meagan Vigus. 2021. AutoAspect: Automatic Annotation of Tense and Aspect for Uniform Meaning Representations. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 36–45, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jayeol Chun and Nianwen Xue. 2024. Uniform meaning representation parsing as a pipelined approach. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 40–52, Bangkok, Thailand. Association for Computational Linguistics.

Bernard Comrie. 1976. *Aspect: An introduction to the study of verbal aspect and related problems*, volume 2. Cambridge University Press.

William Croft. 2012. *Verbs: Aspect and causal structure*. OUP Oxford.

William Croft. 2022. Constructions of the World's Languages. In *Morphosyntax*.

Leon RA Derczynski. 2017. *Automatically ordering events and times in text*. Springer.

Lucia Donatelli, Michael Regan, William Croft, and Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 96–108.

David R Dowty. 1986. The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics? *Linguistics and philosophy*, pages 37–61.

Annemarie Friedrich, Alexis Palmer, and Manfred Pinkal. 2016. Situation entity types: automatic classification of clause-level aspect. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1757–1768.

Annemarie Friedrich, Nianwen Xue, and Alexis Palmer. 2023. A kind introduction to lexical and grammatical aspect, with a survey of computational approaches. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 599–622, Dubrovnik, Croatia. Association for Computational Linguistics.

William Gantt, Lelia Glass, and Aaron Steven White. 2022. Decomposing and recomposing event structure. *Transactions of the Association for Computational Linguistics*, 10:17–34.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo

Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models.

Erhard Hinrichs. 1986. Temporal anaphora in discourses of English. *Linguistics and philosophy*, pages 63–82.

Paul Kingsbury and Martha Palmer. 2003. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer.

Wolfgang Klein. 2013. *Time in language*. Routledge.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Tim O'Gorman, Martha Palmer, Nathan Schneider, and Madalina Bardocz. 2020. Abstract meaning representation (AMR) annotation release 3.0.

Thomas Kober, Malihe Alikhani, Matthew Stone, and Mark Steedman. 2020. Aspectuality across genre: A distributional semantics approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4546–4562, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Klaus Krippendorff. 2004. Measuring the Reliability of Qualitative Text Analysis Data. *Quality and Quantity*, 38:787–800.

Ronald W Langacker. 2011. Remarks on English aspect. In *Tense-aspect: Between semantics & pragmatics*, pages 265–304. John Benjamins Publishing Company.

Egoitz Laparra, Dongfang Xu, and Steven Bethard. 2018. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations. *Transactions of the Association for Computational Linguistics*, 6:343–356.

Charles N Li and Sandra A Thompson. 1989. *Mandarin Chinese: A functional reference grammar*. Univ of California Press.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Betsy Hicks McDonald. 1982. *Aspects of the American Sign Language predicate system*. State University of New York at Buffalo.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational linguistics*, 14(2):15–28.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd workshop on computing news storylines (CNS 2016)*, pages 47–56.

Barbara H Partee. 2011. Nominal and Temporal Semantic Structure. In *Prague Linguistic Circle Papers: Travaux du cercle linguistique de Prague nouvelle série. Volume 3*, pages 91–108. John Benjamins Publishing Company.

James Pustejovsky, Harry Bunt, and Annie Zaenen. 2017. Designing annotation schemes: From theory to model. *Handbook of Linguistic Annotation*, pages 21–72.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Hans Reichenbach. 1947. *Elements of Symbolic Logic*. Macmillan, New York.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Haibo Sun, Nianwen Xue, Jin Zhao, Liulu Yue, Yao Sun, Keer Xu, and Jiawei Wu. 2024. Chinese UMR annotation: Can LLMs help? In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 131–139, Torino, Italia. ELRA and ICCL.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Herman Van Olphen. 1975. Aspect, tense, and mood in the Hindi verb. *Indo-Iranian Journal*, 16(4):284–301.

Z Vendler. 1967. Linguistics in Philosophy Ithaca, NY: Cornell Univ.

Zeno Vendler. 1957. Verbs and times. *The philosophical review*, 66(2):143–160.

Shira Wein and Julia Bonn. 2023. Comparing UMR and cross-lingual adaptations of AMR. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 23–33, Nancy, France. Association for Computational Linguistics.

9. Language Resource References

Julia Bonn, Claire Bonial, Matt Buchholz, Hsiao-Jung Cheng, Alvin Chen, Ching-wen Chen, Andrew Cowell, William Croft, Lukas Denk, Ahmed Elsayed, Eva Fučíková, Federica Gamba, Carlos Gomez, Jan Hajič, Eva Hajičová, Jiří Havelka, Loden Havenmeier, Ath Kilgore, Veronika Kolářová, Lucie Kučová, Kenneth Lai, Bin Li, Jingyi Li, Markéta Lopatková, Marie MacGregor, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Skatje Myers, Michal Novák, Tim O'Gorman, Petr Pajas, Alexis Palmer, Martha Palmer, Jarmila Panevová, Claire Benét Post, James Pustejovsky, Petr Sgall, Jialin Song, Li Song, Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Haibo Sun, Yao Sun, Rosa VallejosYopán, Jens VanGysel, Meagan Vigus, Kristin Wright-Bettner, Jiawei Wu, Nianwen Xue, Dan Xing, Keer Xu, Zhixing Xu, Liulu Yue, Daniel Zeman, Jin Zhao, Šárka Zikánová, and Zdeněk Žabokrtský. 2025. Uniform meaning representation 2.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Julia Bonn, Chen Ching-wen, James Andrew Cowell, William Croft, Lukas Denk, Jan Hajič, Kenneth Lai, Martha Palmer, Alexis Palmer, James Pustejovsky, Haibo Sun, Rosa Vallejos Yopán, Jens Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2023. Uniform meaning representation. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

10. Appendix A - Data Statistics and Dataset Splits

For more information on aspect data statistics, Table 5 shows general information on aspect data that was labeled in the UMR data prior to this annotation project. Next, Table 6 shows the distributions and counts at the end of Phase 1 annotation and before Phase 2 adjudication during our project. Finally, Table 7 provides statistics for our dataset splits. Precise Sentence IDs for the splits are available on Github.[8]

11. Appendix B - Annotation

Training materials. These were developed mostly from existing UMR tutorial materials and supplemented with custom task-specific resources, including an explanatory slide deck[9] which summarizes the UMR guidelines[10] with added clarifications and examples.

Practice annotation. Table 8 shows the results from the Pear Story practice annotation task. Due to the different number of annotations each person performed, we report Gwet's AC1 as a measure for inter-annotator agreement (IAA) since this metric can be calculated for different numbers of labels. We report Fleiss' Kappa only for predicates that were labeled by all annotators. We find moderate-to-good IAA for the practice round, motivating the need for additional training that was conducted.

Annotation process diagram. Figure 3 illustrates the flow of data through the two phases of corpus building, including multiple rounds of tie-breaking. The 143 events listed as single-annotated in the first-pass were part of a teaching demonstration, but they did ultimately receive second annotations and were reviewed for consistency during adjudication; this detail was omitted from the diagram for visual simplicity.

12. Appendix C - Modeling

In this Appendix we provide additional information on the automatic aspect modeling design and results.

LLM Prompt Tuning. We try three strategies to gauge the impact of prompt structure on LLM performance. Initially, we manually draft a list of short definitions for each aspect class based on the experience gained from our annotator training sessions. In a second prompt attempt, we provide the initial prompt and instruct the model to generate a better prompt for our task, to investigate if the LLM's pretraining contains aspectual knowledge beyond our basic definitions, which resulted in a streamlined version with more general task instruction. Finally, to take advantage of extensive LLM

[8] https://github.com/clairepost/UMR_Aspect_Data.git

[9] Please see our GitHub for more information.

[10] https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md

Aspect	Little Prince	Minecraft	BOLT DF	WB	Pear Story	Lorelei	UMR 1.0	Total
State	63	20	119	45	121	0	62	430
Habitual	1	0	17	4	28	0	2	52
Process	2	0	5	5	44	1	1	58
Activity	9	0	31	14	57	0	21	132
Performance	35	2	43	18	159	3	57	317
Endeavor	0	0	0	0	2	0	14	16
Total	**110**	**22**	**215**	**86**	**411**	**4**	**157**	**1005**

Table 5: Aspect label distribution from different existing UMR datasets before any additional annotation was done.

Aspect	**Little Prince**	**Minecraft**	**BOLT DF**	**WB**	**Existing Labels**	**Total**
State	172	14	101	14	430	731
Habitual	41	0	4	0	52	97
Process	31	0	38	8	58	135
Activity	15	2	10	3	132	162
Performance	163	43	69	32	317	624
Endeavor	15	0	4	0	16	35
None	158	49	121	77	-	405
Total	**595**	**108**	**347**	**134**	**1,005**	**2,189**
Fleiss' Kappa	0.78	0.82	0.45	0.40	-	-

Table 6: Label distribution by corpus and annotated aspect. We report Fleiss' Kappa between the two initial annotators and do not include disagreements in the reported total.

Split	**Sentences**	**Events**
Train	333	999
Dev	71	220
Test	72	254
Total	**476**	**1,473**

Table 7: Stratified 70/15/15 train/dev/test split, divided at the sentence level using dominant aspect label for stratification.

context windows, we try providing the UMR guidelines for aspect[11] in their entirety and instructing the model to predict a label. We find marginally higher validation accuracy with the second strategy, and employ it in testing. We provide the full prompt we used in testing in Table 9, as well as in our GitHub repository.

Category	Metric	Value
Accuracy	State	0.82
	Habitual	0.63
	Activity	0.52
	Performance	0.80
	Endeavor	0.11
	Overall Accuracy	0.74
	Perfect Accuracy	0.35
F1	Macro F1	0.49
	Weighted Macro F1	0.76
IAA	Fleiss' Kappa	0.55
	Gwet's AC1	0.66

Table 8: Practice Annotation Results: *Overall Accuracy* is the ratio of the total number of correct annotations over the total number of predicates annotated. *Perfect Accuracy* is the ratio of predicates that were correctly annotated by all annotators. No occurrences of *Process* aspects were present in the practice set.

[11] github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md#part-3-3-1-Aspect

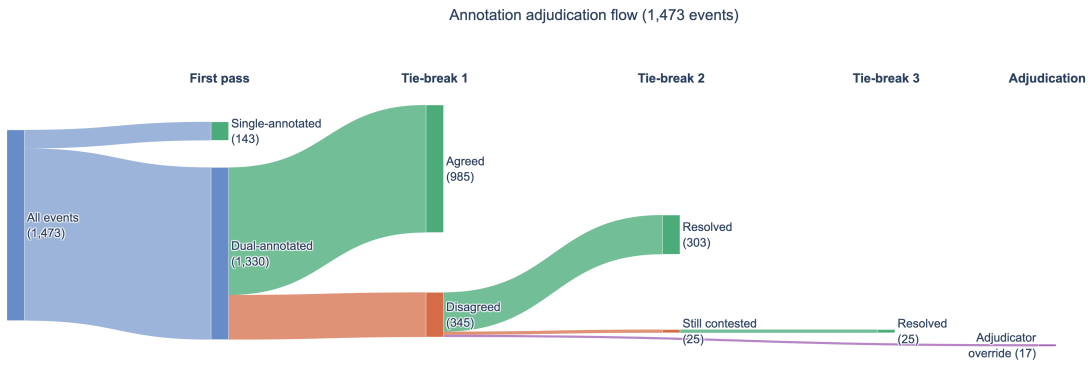


Figure 3: Flow diagram illustrating the annotation and adjudication phases, and how many labels were completed at each stage of the process.

Setting	Prompt
Experiment: Few-shot with definitions (3 examples per class, total 21 examples) **Model:** `gpt-5-mini` **Seed:** 42	**[SYSTEM TURN]** You are a UMR expert, required to predict the aspectual value of a predicate in a given sentence. The goal is to annotate the aspect of each predicate, which can be one of six distinct values: State, Habitual, Process, Activity, Endeavor, Performance, or a seventh 'None' option if you think the given predicate is not an event. Respond with only the label name and nothing else. Do not restate the sentence, predicate, or provide any reasoning. **[USER TURN]** **Definitions:** • state: stable condition or property (e.g. "knows", "believes") • habitual: recurring or generic action (e.g. "walks to school every day") • process: ongoing activity without clear endpoint (e.g. "is running") • activity: dynamic event (e.g. "built a house") • performance: bounded event with natural endpoint (e.g. "arrived") • endeavor: attempted action that may not fully complete (e.g. "tried to open") • none: no clear aspectual reading **Examples:** Sentence: "The new immigration law also permits the immigration service to provide " limited " biometric information on New Zealand citizens to neighboring countries ." | Predicate: "immigrate-01" -> none Sentence: "And as I had with me neither a mechanic nor any passengers , I set myself to attempt the difficult repairs all alone ." | Predicate: "attempt-01" -> endeavor Sentence: "long hours and lots of long nights." | Predicate: "long-03" -> none Sentence: "I am about to receive a visit from an admirer ! " he exclaimed from afar , when he first saw the little prince coming ." | Predicate: "come-01" -> activity Sentence: "However, I just have to say that I think it is ludicrous for John to accept any other explanation for his encounter except for what it is." | Predicate: "contrast-91" -> none Sentence: "" Yes ? " said the little prince , who did not understand what the conceited man was talking about ." | Predicate: "talk-01" -> activity Sentence: "At a glance I can distinguish China from Arizona ." | Predicate: "glance-01" -> endeavor Sentence: "" Yes ? " said the little prince , who did not understand what the conceited man was talking about ." | Predicate: "understand-01" -> state Sentence: "" What ! "" | Predicate: "say-91" -> performance Sentence: "" This man , " the little prince said to himself , " reasons a little like my poor tippler ... "" | Predicate: "reason-01" -> habitual Sentence: "Freedom of speech has never been understood to restrict the ability of people to sue other people for things like libel and slander." | Predicate: "restrict-01" -> state Sentence: "In the course of this life I have had a great many encounters with a great many people who have been concerned with matters of consequence ." | Predicate: "encounter-01" -> habitual Sentence: "And as I had with me neither a mechanic nor any passengers , I set myself to attempt the difficult repairs all alone ." | Predicate: "repair-01" -> endeavor Sentence: "Absurd as it might seem to me , a thousand miles from any human habitation and in danger of death , I took out of my pocket a sheet of paper and my fountain - pen ." | Predicate: "die-01" -> process Sentence: "" I admire you , " said the little prince , shrugging his shoulders slightly , " but what is there in that to interest you so much ? "" | Predicate: "shrug-01" -> activity Sentence: "" It is to raise in salute when people acclaim me ." | Predicate: "raise-01" -> habitual Sentence: "If I owned a flower , I could pluck that flower and take it away with me ." | Predicate: "possible-01" -> state Sentence: "This higher health spending is a function of different prices and different usage of medical care." | Predicate: "use-01" -> process Sentence: "[Builder puts down a green block at X:1 Y:2 Z:0]" | Predicate: "put-down-17" -> performance Sentence: "So I lived my life alone , without anyone that I could really talk to , until I had an accident with my plane in the Desert of Sahara , six years ago ." | Predicate: "talk-01" -> process Sentence: "Here you may see the best portrait that , later , I was able to make of him ." | Predicate: "make-01" -> performance **Classify:** Sentence: "My mom just retired." Predicate: "retire-01" Label: **[ASSISTANT TURN]**

Table 9: Few-shot prompt used for LLM aspect classification.

Modelling Idiomatic Expressions in Abstract Meaning Representation

Venera Gareeva[1], Johannes Heinecke[2]
[1]Université de Strasbourg, 67000 Strasbourg, France
[2]Orange Research, 22300 Lannion, France
venera.gareeva@etu.unistra.fr, johannes.heinecke@orange.com

Abstract

Idiomatic expressions, a subclass of multiword expressions (MWE), pose persistent challenges for semantic parsing, as their meanings often diverge from the compositional semantics of their constituent words and depend strongly on contextual cues. While Abstract Meaning Representation (AMR) parsers aim to capture sentence-level semantics in a structured graph form, existing datasets provide limited coverage of idiomatic language, constraining their ability to model such expressions accurately. To address this gap, we extended a subset of the MAGPIE dataset by constructing a corpus of potentially idiomatic expressions (PIE) annotated with their corresponding AMR graphs. The dataset includes both naturally occurring and synthetically generated sentences, covering idioms in literal and idiomatic contexts. We fine-tune a state-of-the-art AMR parser on this dataset and evaluate its capacity to generate context-sensitive graphs that correctly reflect idiomatic versus literal interpretations. Our results show that standard parsers often capture only literal meanings of such expressions, while fine-tuning on our dataset improves alignment with the intended interpretations.

Keywords: idiomatic expressions, Abstract Meaning Representation, idiom corpus, parsing

1. Introduction

Several formalisms have been proposed to capture semantic structures. Among them, Abstract Meaning Representation (AMR) is a formalism whose aim is to provide an abstract, standardized representation that is independent of syntax and structured as directed graphs (Banarescu et al., 2013). While AMR is designed to represent the meaning of sentences, a challenge arises in how to capture idiomatic expressions, whose interpretations cannot be derived literally, for example, *kick the bucket* ("to die") or *pull someone's leg* ("deceive someone jokingly").

Within the AMR framework, different types of MWEs (combinations of words whose meaning cannot always be directly inferred from the meanings of their individual components (Ramisch, 2023)) are handled through the semantic role structure provided by PropBank (Kingsbury and Palmer, 2002), which supplies verb-specific frames that guide annotation. As a result, constructions such as verb–particle combinations and light verb constructions are generally well represented, since their meanings align with established predicate–argument structures.In contrast, the representation of idiomatic expressions in PropBank, and consequently in AMR, remains limited.

This limitation is particularly critical because understanding idiomatic expressions in context is essential for capturing the intended meaning of a text. Unlike other types of MWEs (like collocations or light verb constructions), idioms cannot be resolved solely through lexical or syntactic cues; their interpretation depends on recognizing pragmatic and contextual information (Beck and Weber, 2020). Human processing of idioms relies heavily on interpreting the surrounding context, which determines whether an figurative or literal reading is activated. For example, *"in hot water"* can denote a physical situation (e.g., cooking vegetables) or a figurative state of trouble, depending on context.

In this paper, we investigate how idiomatic expressions are represented in AMR corpora and whether text-to-AMR parsers distinguish figurative from literal readings. We focus on verbal idioms and introduce a dedicated English corpus to fine-tune state-of-the-art AMR parsers for evaluating context-sensitive interpretations.While adaptations of PropBank exist for other languages, such as Universal Propositions (Akbik et al., 2015)[1], their limited coverage (e.g., 533 verbs for French), together with AMR's reliance on PropBank framesets, motivates English as a natural starting point before multilingual extension.

1.1. Abstract Meaning Representation

In the AMR formalism, sentences are represented as graphs whose nodes correspond to concepts and edges to semantic relations. AMR graphs can be encoded in several formats, including PENMAN notation (Kasper, 1989), a textual representation

[1]https://github.com/UniversalPropositions/UP-1.0

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 37–43
11 May 2026.

that uses nested parentheses to hierarchically organize concepts and their semantic roles, making graph structure explicit.

AMR is based largely on semantic frames derived from the PropBank project (Kingsbury and Palmer, 2002; Palmer et al., 2005), that provides a detailed inventory of verb-specific roles and their typical arguments. PropBank thus allows for a clear identification of "who does what to whom" in a sentence.

To train models that predict AMR graphs from text, several datasets are available, the largest being AMR 3.0 (Knight et al., 2020), distributed by the Linguistic Data Consortium (LDC2020T02). The corpus contains nearly 60,000 sentences, divided into 55,635 training, 1,722 validation, and 1,898 test instances.

AMR graphs are commonly evaluated using the Smatch metric (Cai and Knight, 2013), which measures overlap between predicted and gold graphs by comparing sets of graph triples. Because node variables are arbitrary, Smatch computes an optimal alignment that maximizes F1, based on precision and recall over matched triples. A more recent evaluation library is Smatch++ which provides a more accurate mapping of graphs (Opitz, 2023) or takes into account semantic similarity (Opitz et al., 2020).

2. Related Work

AMR is a powerful formalism for capturing propositional content, but its handling of figurative and idiomatic language is limited. Mansouri (2025) claims that AMR struggles with idioms and metaphors, often representing them literally or abstracting them away without capturing their figurative meaning. Thus, an idiomatic expression such as *"kick the bucket"* may be represented as an action involving a bucket rather than the intended metaphorical sense of death.

That said, it would be overly reductive to claim that AMR is completely incapable of handling idioms. The AMR 3.0 corpus (cf. section 1.1 above) includes some annotated idiomatic expressions. While coverage is far from exhaustive, these examples show that the framework can, at least in part, encode certain figurative meanings rather than collapsing them into literal interpretations. While there has been substantial work on multiword expressions (Baldwin and Kim, 2010; Zeng and Bhat, 2021; Ramisch, 2023), relatively little research has explored the use of semantic formalisms for idiomatic expressions. A closely related line of work by Evang et al. (2025) in Discourse Representation Structure (DRS) parsing shows that, with suitable training data, semantic parsers can learn to distinguish between literal and idiomatic readings, although predicting idiomatic meanings in context remains challenging. Otherwise, several non-semantic approaches have been proposed to identify idiomatic expressions (Zeng and Bhat, 2021; He et al., 2024). In chapter 3 we will notably use the work of Haagsma et al. (2020).

3. Multiword expressions

Multiword expressions (MWE) represent a complex linguistic phenomenon and pose challenges for natural language processing. Following the definition of Baldwin and Kim (2010), MWEs are defined as lexical items composed of multiple lexemes that exhibit lexical, syntactic, semantic, pragmatic, and/or statistical idiomaticity. Unlike named entities, which can be identified through formal indicators such as capitalization or trigger words, MWEs lack systematic markers, making their detection difficult (Savary et al., 2019). Their variability further complicates processing: verbal MWE, for instance, may appear in diverse syntactic forms through passivization, modification, or nominalization. However, in the context of AMR parsing, syntactic variability is less central, as AMR abstracts away from surface syntax and word order.

Beyond formal variability, the main difficulties posed by MWEs are semantic ambiguity and non-compositionality: they may be interpreted literally or figuratively depending on context, and their meanings are often not derivable from their parts.

Idiomatic expressions form a subclass of MWEs, sharing these properties. Their interpretation, however, relies heavily on shared cultural knowledge (Chung, 2024), making them opaque to non-native speakers and difficult to translate across languages. In line with (Nunberg et al., 1994), a distinction can be drawn between idiomatically combining expressions, whose meanings are partly distributed across their components, and idiomatic phrases, whose meanings are fixed and not compositionally derived from their parts. In our work, the set of verbal expressions under study includes examples of both types.

The MAGPIE corpus (Haagsma et al., 2020) is the largest annotated resource for potentially idiomatic expressions (PIEs) in English, containing over 56,000 instances across nearly 1,800 expression types. PIEs are defined as fixed or semi-fixed expressions that may be interpreted literally or figuratively depending on context.

For the purposes of this study, a subset of the MAGPIE corpus was manually selected, retaining only frequent idioms with both literal and idiomatic attestations. This resulted in a working corpus of 3,582 sentences covering 41 mostly verbal idioms. This focus reflects their frequency and suitability for studying contextual meaning variation,

and serves as a starting point for future extensions beyond verbal idioms. The selected subset was annotated according to the process described in Section 4.

4. Annotation

As a preliminary step, the AMR 3.0 corpus (Knight et al., 2020) was examined via the metAMoRphosED tool (Heinecke, 2023) to identify idiomatic expressions and assess how they were represented by the expert annotators. This exploration revealed that idioms were relatively rare and inconsistently treated: some were covered by PropBank frames (Palmer et al., 2005) (e.g., *steer clear* encoded as `steer-clear-02`), others were paraphrased into equivalent concepts (e.g., *goes without saying* encoded as `obvious-01`), while many were either absent or interpreted literally. Based on these observations, a consistent annotation strategy was established for our corpus. Each idiomatic expression identified in a sentence was handled according to the following principles:

1. If a corresponding semantic frame was already defined in PropBank, this frame was used directly to encode the idiomatic expression. This ensured compatibility with AMR standards. For instance, *follow suit* can be represented by `follow-suit-06`, which directly captures its figurative meaning without additional reformulation. A selection of such idiomatic frames for verbal expressions is presented in Table 1.

2. When no such frame is available, the idiomatic expression was paraphrased (when used in its figurative meaning) to capture its implicit meaning, and the AMR annotation was based on this paraphrase (e.g., *tie the knot* as `marry-01` or *bear fruit* as `produce-01`).

Subgraphs corresponding to idiomatic expressions in their figurative sense were systematically replaced using a rule-based script. Each idiom was associated with transformation rules that specify the patterns to delete and the semantic structures to insert (e.g., *tie the knot* → `marry-01`).

To illustrate, Fig. 1 and Fig. 2 show the expression *tie the knot* in its two different readings. Thus, in the figurative example, the subgraph corresponding to the idiom is replaced by the single semantic unit *marry-01*, whereas in the literal usage, the compositional structure is preserved, with `tie-01` taking *knot* as its (`:ARG1`) argument.

From the MAGPIE corpus, we initially gathered 3,352 sentences for the 41 selected expressions. During data preparation, literal usages of PIEs were underrepresented in the original MAGPIE corpus, so the dataset was augmented with 250 GPT-4.1-nano–generated sentences emphasizing literal meanings using a one-shot prompting strategy. The final dataset contains 3,582 sentences (2,886 train, 360 dev, 336 test). We do not report inter-annotation agreement as the corpus was annotated by a single annotator.

PropBank	Arguments
follow-suit-06	ARG0: imitator
	ARG1: thing imitated
	ARG2: action of following suit
change-hands-06	ARG1: thing changing hands
	ARG2: giver
	ARG3: getter
steer-clear-02	ARG0: avoider
	ARG1: avoided

Table 1: Semantic frames for selected idiomatic expressions in PropBank

```
(t / together
   :domain (a / and
      :op1 (p / person
         :name (n / name
            :op1 "Fiona"))
      :op2 (p2 / person
         :name (n2 / name
            :op1 "Paul")))
   :duration (t2 / temporal-quantity
      :quant 6
      :unit (y / year))
   :time (b / before
      :op1 (d / decide-01
         :ARG0 a
         :ARG1 (m / marry-01
            :ARG0 a))))
```

Figure 1: AMR graph for the sentence: *Fiona & Paul had been together for six years before deciding to tie the knot.* (the expression *tie the knot* is used in its figurative meaning)

```
(s / show-01
   :ARG0 (p / person
      :ARG0-of (s2 / sail-01))
   :ARG1 (t / thing
      :manner-of (t2 / tie-01
         :ARG0 p
         :ARG1 (k / knot)
         :ARG1-of (c / correct-02)))
   :ARG2 (w / we))
```

Figure 2: AMR graph for the sentence *The sailor showed us how to tie the knot correctly* (the expression *tie the knot* is used in its literal meaning)

5. Experiments

To evaluate whether text-to-AMR parsers are capable of distinguishing between the literal and figurative interpretations of expressions, pretrained *seq2seq* models were fine-tuned following the annotation process. By exposing the model to idiom-annotated data, we aim to assess its capacity to become idiom-aware, that is, to capture both figurative and literal meanings of PIE expressions.

However, *seq2seq* models cannot directly handle graph structures, which makes data preprocessing essential. Thus, data preprocessing was necessary to adapt AMR graphs to seq2seq models, which require linear sequences as input and output.To this end, AMR graphs originally represented in PENMAN were serialized into a simpler format while preserving semantic structure.

For the purpose of fine-tuning, we used the Flan-T5 model (Chung et al., 2022), a variant of T5 (Text-to-Text Transfer Transformer) pretrained under the instruction tuning paradigm, which has demonstrated strong performance across a wide range of NLP tasks and shown to achieve SoTA results in AMR parsing (Lee et al., 2023). We chose the Flan-T5 model since it proved to be the most reliable model of the T5 family (T5, Flan-T5 and the multilingual MT5). All these models come in different sizes (small, base, large, xl, xxl). Even though Flan-T5 xl and xxl gave slightly better results, we kept Flan-T5 base (250M parameters and a vector length of 768) since it needed less computational power (in terms of memory of the GPU device and compute time).

We also tried to use more recent LLMs like Qwen 2.5 (0.5B, 1.5B, 3B and 7B parameters) or Gemma 2 (1B and 2B), but again these models trained on the AMR 3.0 dataset performed slightly less well than flan-T5 base on AMR 3.0.

The first experimental step was to train the Flan-T5 model on AMR 3.0, producing a baseline system capable of generating AMR structures from natural language. This baseline was then further fine-tuned on our idiom-specific corpus. The fine-tuning was implemented using the Huggingface's Seq2Seq Trainer framework. Two experimental setups were then conducted. In the first, the FlanT5 model was fine-tuned sequentially—first on the AMR 3.0 corpus and then on the idiom-specific AMR-annotated corpus. In the second, both corpora were concatenated into a single dataset to enable joint fine-tuning. Overall, no major differences in performance were observed between the two approaches.

5.1. Results

To assess the performance of our fine-tuned models, we first evaluate the models trained on the AMR 3.0 training set and tested on the AMR 3.0 test set. Table 2 summarizes the results of AMR parsing experiments conducted with different learning rates and beam search configurations across various sizes of the Flan-T5 model (base and large). We choose the base model for future experiments since it performs nearly as well as the large model while remaining more computationally efficient and cost-effective.

model	init. LR	epochs	beam search	Smatch score
base	0.00005	15	No	75.12
base	0.00005	15	Yes	75.15
base	0.0001	10	No	81.72
base	**0.0001**	**10**	**Yes**	**82.12**
large	0.0001	5	No	82.29

Table 2: AMR parsing experiments with Flan-T5 models using different learning rates and beam search configurations

The next step was to fine-tune the models on our idiom-annotated dataset. Test showed a Smatch score of 84.15%.

Subsequently, we compared Smatch scores for each expression against its distribution of literal and figurative occurrences, revealing a clear trend: expressions with balanced examples in both senses generally achieve higher scores, indicating that the model benefits from diverse, evenly distributed training data.

To illustrate this observation, Table 3 provides an overview of selected expressions, showing both the distribution of examples by sense (literal/figurative) and the resulting Smatch scores.

idiom. expr.	total	fig. usage	lit. usage	Smatch fig.	Smatch lit.
go without saying	99	95	4	85	60
bear fruit	103	99	4	90	50
behind bars	98	27	71	82	91
tie the knot	42	18	24	91	93
spill the beans	31	30	1	87	0
ring a bell	122	16	106	100	90
cold feet	14	6	8	40	75
hald water	50	19	31	69	87

Table 3: Individual Smatch Scores by literal and figurative sense for a selection of idiomatic expressions

The corpus was then adjusted, as described in the annotation section (mostly to address cases where it lacked examples of expressions in their literal usage). A new fine-tuning was subsequently carried out on this rebalanced corpus. The performances were re-evaluated with Smatch score of

85.73%, which is an increase of nearly 1.6 points from 84.15% before the adjustments.

Comparing these individual scores to those obtained before the dataset adjustment (cf. Table 3) reveals a clear pattern: the parsing of literal usages improved substantially in most cases, particularly for expressions that initially had very few literal examples (e.g., the expressions *go without saying* and *bear fruit*). For instance, the Smatch score for literal usages of *go without saying* increased from 60% to 98%, while the score for the figurative sense remained nearly stable. Thus, bold figures in the table highlight significant improvements in literal usage parsing following dataset rebalancing, showing that the model learned to better capture literal meanings without significantly affecting performance on figurative usages. Therefore, the injection of idiomatic data into the training pipeline already yielded some improvements across all cases, and even where gains are more modest, a positive tendency is observed, suggesting that enriching the parser with dedicated idiomatic expressions is a promising direction for future development.

idiom. expr.	**total**	**fig. usage**	**lit. usage**	**Smatch fig.**	**Smatch lit.**
go without saying	159	95	64	82	**98**
bear fruit	153	99	54	92	**91**
tie the knot	65	18	47	92	93
spill the beans	52	30	22	67	**100**
ring a bell	122	16	106	100	90
miss the boat	36	14	22	91	89
off the hook	42	31	11	86	80
green light	109	63	46	84	85
cold feet	38	15	23	100	**70**

Table 4: Individual Smatch score for some idiomatic expressions after adjusting the distribution in the dataset. Bold figures indicate improved parsing of literal usages, while the performance on figurative usages remains nearly unaffected.

6. Conclusion

In this paper, we analyzed how idiomatic expressions are represented in AMR and evaluated whether text-to-AMR parsers can distinguish literal from figurative readings. An analysis of the AMR 3.0 corpus showed that idioms are underrepresented and inconsistently annotated.

To address this gap, we constructed a dedicated corpus of potentially idiomatic expressions from MAGPIE, covering over 3,500 instances of 41 frequent idioms with both literal and figurative uses. Each expression was semantically normalized using a rule-based procedure, either by mapping it to an existing PropBank frame or by replacing it with a figurative paraphrase. This corpus was then used to fine-tune Flan-T5 models for idiom-aware AMR parsing.

Experimental results show that balanced distributions of literal and figurative examples improve model performance, suggesting that generalization across idiomatic senses depends on exposure to both readings.

6.1. Limitations

However, several limitations remain. First, only a restricted set of idiomatic expressions was covered, which limits the generalization of the trained models. A larger and more diverse corpus would likely improve the robustness of idiom-aware parsing. Second, the annotation process lacked multiple independent annotators, so inter-annotator reliability was not assessed. Third, some idioms that are difficult to paraphrase still require the definition of additional PropBank frames to produce accurate AMR graphs (e.g., *break the ice*). Finally, the corpus is limited to English, restricting linguistic scope. Future work is needed to expand idiom coverage and other languages. Nonetheless, simply translating the existing corpus is unlikely to preserve idiomatic equivalence, since each language possesses its own distinct system of idioms.

7. Bibliographical References

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating high quality proposition Banks for multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407, Beijing, China. Association for Computational Linguistics.

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In *Handbook of Natural Language Processing*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Sara Beck and Andrea Weber. 2020. Context and literality in idiom processing: Evidence from self-paced reading. *Journal of Psycholinguistic Research*, 49.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Dang Chung. 2024. Challenges of translating idiomatic expressions: A cross-linguistic analysis at a university in hanoi, vietnam. *International Journal of Social Science and Human Research*, 07.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Kilian Evang, Rafael Ehren, and Laura Kallmeyer. 2025. The proper treatment of verbal idioms in German discourse representation structure parsing. In *Proceedings of the 16th International Conference on Computational Semantics*, pages 156–165, Düsseldorf, Germany. Association for Computational Linguistics.

Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. MAGPIE: A large corpus of potentially idiomatic expressions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 279–287, Marseille, France. European Language Resources Association.

Wei He, Marco Idiart, Carolina Scarton, and Aline Villavicencio. 2024. Enhancing idiomatic representation in multiple languages via an adaptive contrastive triplet loss. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12473–12485, Bangkok, Thailand. Association for Computational Linguistics.

Johannes Heinecke. 2023. metAMoRphosED, a graphical editor for Abstract Meaning Representation. In *Proceedings of the 19th Joint ACL-ISO Workshop on Interoperable Semantics (ISA-19)*, pages 27–32, Nancy, France. Association for Computational Linguistics.

Robert T. Kasper. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 1989–1993, Las Palmas, Canary Islands, Spain. European Language Resources Association.

Young-Suk Lee, Ramón Fernandez Astudillo, Radu Florian, Tahira Naseem, and Salim Roukos. 2023. Amr parsing with instruction fine-tuned pre-trained language models.

Behrooz Mansouri. 2025. Survey of abstract meaning representation: Then, now, future.

Geoffrey Nunberg, Ivan A. Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.

Juri Opitz. 2023. SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1595–1607, Dubrovnik, Croatia. Association for Computational Linguistics.

Juri Opitz, Anette Frank, and Letitia Parcalabescu. 2020. Amr similarity metrics from principles. *Transactions of the Association for Computational Linguistics*, 8(0):522–538.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Carlos Ramisch. 2023. *Multiword expressions in computational linguistics*. Habilitation à diriger des recherches, Aix Marseille Université (AMU).

Agata Savary, Silvio Cordeiro, and Carlos Ramisch. 2019. Without lexicons, multiword expression identification will never fly: A position statement. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 79–91, Florence, Italy. Association for Computational Linguistics.

Ziheng Zeng and Suma Bhat. 2021. Idiomatic expression identification using semantic compatibility. *Transactions of the Association for Computational Linguistics*, 9:1546–1562.

8. Language Resource References

Kevin Knight and Bianca Badarau and Laura Baranescu and Claire Bonial and Madalina Bardocz and Kira Griffitt and Ulf Hermjakob and Daniel Marcu and Martha Palmer and Tim O'Gorman and Nathan Schneider. 2020. *Abstract Meaning Representation (AMR) Annotation Release 3.0*. Linguistic Data Consortium. distributed via LDC: LDC2020T02, 3.0, ISLRN 676-697-177-821-8.

TrAinMR: an Annotator Training Website for Abstract Meaning Representation

Mina Yang, Shira Wein
Amherst College
Amherst, MA, United States
{miyang27, swein}@amherst.edu

Abstract

Abstract Meaning Representation (AMR) is a graph-based semantic representation which captures the core elements of meaning of a text. AMR has been incorporated into a variety of downstream tasks, which rely heavily on the availability of gold-annotated AMR corpora. While the annotation process is fairly lightweight, annotator training is still required even for linguists due to the extensive nature of the annotation guidelines and comprehensive set of roles. Therefore, all corpus development projects for AMR (and extensions of AMR) require the dataset curators to first train annotators. In this paper, we develop an online AMR annotation training system called TrAinMR in order to ease this training process and thus motivate the development of additional AMR corpora. The two main components of TrAinMR are (1) a written tutorial covering the basics of AMR annotation, and (2) an interactive practice module with corrective feedback. To measure the effectiveness of this tool, we conduct two pilot studies with five human annotators each. We find that the majority of annotators state their understanding of AMR improved as a result of TrAinMR, and some annotators show a positive trend in SMATCH scores after completing the practice module.

Keywords: semantics, annotation, abstract meaning representation

1. Introduction

The graph-based semantic representation Abstract Meaning Representation (AMR; Banarescu et al., 2013) captures the relationship between concepts in a sentence of the form "who did what to whom" (see Figure 1). AMR has been incorporated into a variety of downstream applications in areas such as machine translation (Wein and Schneider, 2024; Song et al., 2019) and summarization (Liao et al., 2018), leveraging prior efforts towards gold annotation of AMR.

While datasets such as AMR3.0 (Knight et al., 2020) are widely used, creating AMR corpora requires the work of human annotators, and for datasets containing over thousands of AMRs, the annotation process becomes costly in time and work. This is in large part due to the extensive nature of the AMR guidelines. While AMR annotation is fairly lightweight, in order to produce accurate annotations, AMR annotators must be trained to identify relationships in the text and produce the correct AMR relations. Furthermore, prior work investigating the automation of AMR production finds that relying on GPT models to automate the curation of AMR datasets is not feasible due to the frequency of mistakes in AMR annotation (Ettinger et al., 2023). Therefore, although AMR annotation by hand can be laborious, our goal is to make the training process faster by providing a ready-to-use training schema for any AMR project developers. TrAinMR is designed to lessen the barrier of entry to AMR annotation, enabling novice non-linguists to begin to contribute to annotation projects at scale. Our tool may also be helpful as a first step for crowdworkers, if the project developer selects to crowdsource AMRs and incorporate expert intervention for more complicated instances (Martin et al., 2020).[1]

Sentence: The cat has orange stripes.

AMR Graph:

```
(h / have-03
    :ARG0 (c / cat)
    :ARG1 (s / stripe)
        :mod (o / orange))
```

Figure 1: The AMR graph for the simple sentence "The cat has orange stripes." For the root `(h / have-03)`, `have` is the concept, `-03` is the sense number as identified via PropBank, and `h` is the variable name which enables coreferential relations (i.e., for concepts which are referred back to later in a graph).

To assist with the challenging process of training new annotators, we present TrAinMR (Training in Abstract Meaning Representation), a web-based tool that consists of two main components: a Tutorial module which introduces the fundamental guidelines of AMR annotation, and an interactive Annotation Practice module. The Annotation Practice module asks users to annotate 25 sentences, and provides immediate feedback by comparing the user's annotations to a gold reference annota-

[1]The AMR training website is available at https://acnlplab.github.io/amr-training-site/

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 44–53
11 May 2026.

tion.

In order to analyze the tool's effectiveness, we conduct two pilot studies, in which we measure TrAinMR's impact on performance. The study involves 10 novice human annotators who first read the Tutorial, then annotate 25 sentences. We analyze their performance using SMATCH scores (Cai and Knight, 2013), qualitative analysis on individual annotations, and collect post-study feedback from annotators on their experience.

2. Related Work

AMR is a graph-structured semantic representation designed to generalize away from surface-level grammatical details in order to capture the core meaning of a text (Banarescu et al., 2013). AMR concepts are reflected as nodes in the directed, rooted graph, and edges represent the relationship between those concepts; edges are marked with relations including numbered arguments from PropBank (e.g., `:ARGn`) or named relations (e.g., `:location`). Numerous extensions of the AMR annotation schema have been introduced, such as the Uniform Meaning Representation (Van Gysel et al., 2021), BabelNet Meaning Representation (Martínez Lorenzo et al., 2022), Spatial AMR (Bonn et al., 2020), and Gesture-AMR (Brutti et al., 2022)– all of which also require training in AMR before being able to produce the specialized annotations.

Key AMR materials include the AMR guidelines (Banarescu et al., 2019) and an AMR Tutorial presentation from NAACL 2015 that discusses foundational AMR concepts through practice examples (Schneider et al., 2015). Existing annotation interfaces include the (now defunct) AMR Editor (Hermjakob, 2013), CAMRA (Cai et al., 2023), LiDARR (Cai et al., 2025), Anafora (Chen and Styler, 2013), and X-AMR (Ahmed et al., 2024). These tools aid AMR annotation on the document- and corpus-level in order to improve the quality of the AMR, but are all designed to serve as annotation interfaces, not to train annotators. To the best of our knowledge, TrAinMR is the first tool designed for annotation training, serving as a centralized on-boarding for new annotators. By standardizing the initial training process, TrAinMR reduces the burden on new annotation efforts by allowing annotators to complete the TrAinMR system first.

In order to evaluate annotator performance, we use SMATCH, which is a standard approach for comparing two AMR graphs for semantic similarity (Cai and Knight, 2013). SMATCH calculates semantic similarity by aligning the concepts of the two graphs and counting their matching triples. Then, it calculates the precision, recall, and F1-score. The final score is a single value ranging from 0 (no match) to 1 (perfect match). Thus, we are able to compare the user's submission against a gold reference, expert-annotated AMR graph. SMATCH score can also be used to measure inter-annotator agreement in order to validate annotator training and consistency of a dataset. SMATCH scores for inter-annotator agreement of released AMR datasets range from 0.71 to 0.89 SMATCH (Wein, 2025).

3. System Overview

TrAinMR consists of two main components: a Tutorial module and an Annotation Practice module.

The Tutorial page contains written guidance on how to annotate an AMR graph, including instructions on PENMAN notation (Matthiessen and Bateman, 1991), PropBank framesets (Palmer et al., 2005), and common roles and relations.

After reading the Tutorial, users are directed to then practice writing their annotations in the Annotation Practice page. On the Annotation Practice page, we instruct the users to submit their annotations for 25 curated sentences, and detail that the user can view the gold AMR and explanations for the AMR after submitting an annotation. External resources, such as the PropBank repository (Palmer et al., 2005) and the AMR guidelines (Banarescu et al., 2019), are referenced on the Annotation Practice and Tutorial pages, and users are encouraged during both stages to supplement their understanding. By practicing after learning about the AMR guidelines, users are able to apply and reinforce their learning through active recall. This also enables users to quickly identify and remedy misunderstandings from the Tutorial as they receive corrective feedback on their AMR graph submissions.

3.1. Tutorial

The content of the Tutorial page largely follows the list of topics in the AMR paper Banarescu et al. (2013) and AMR Tutorial Slides (Schneider et al., 2015). These topics are: AMR, PENMAN notation, PropBank frames, nominalization, AMR roles and relations (including frame arguments and common relations), reification (converting a relation into a concept), and annotation of negation, modals, and questions. Each topic in the Tutorial includes a brief explanation, as well as example sentences and their corresponding AMR graphs which demonstrate those topics (see Figure 2 for an example). The example sentences in the beginning of the Tutorial are short sentences designed to exemplify a specific topic, such as the application of `:mod`. There are supplemental examples at the bottom of the Tutorial that feature longer AMRs with a combination of previous topics, including inverse relations, modification, and quantification.

Figure 2: An example of the "Listing Entities" section from the Tutorial page is shown, including the AMR for the sentence "Funk and soul."

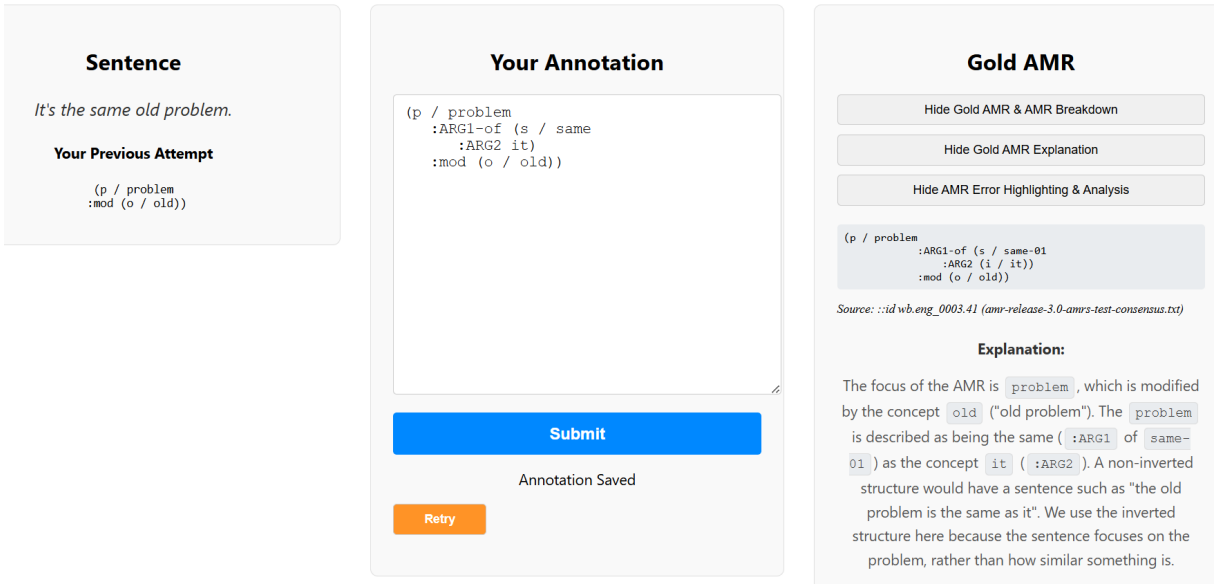


Figure 3: The Annotation Practice module displays feedback when prompted by the user. After selecting to show the Gold AMR, the AMR is shown as well as an explanation of the contents of the graph.

3.2. Annotation Practice

After having engaged with the Tutorial, the user is then able to try AMR annotation on the Annotation Practice page. We store the user's submissions in a Google Sheet and display the highlighted differences between the submission and the gold AMR. We also provide an explanation of the various components of the gold AMRs, including the root concept (the focus of the sentence), other concepts, and the roles and relations that connect them.

The gold AMR is hidden from view until an input is submitted. Once users submit an annotation, their attempt is shown below the sentence and they are able to resubmit if desired. To help users visually track their progress, their most recent submission for each sentence is saved and displayed.

Additionally, after submitting, users have the option to view the gold AMR and its line-by-line breakdown, the gold AMR explanation, and the AMR analysis separately.

3.2.1. User Interface

The Annotation Practice page consists of 25 practice annotations for sentences taken from the AMR3.0 dataset (Knight et al., 2020). We choose sentences manually to reflect a range of complexity and the topics mentioned on the Tutorial page. We define complexity using sentence length and the presence of combinations of topics, such as modals and inverse roles. For example, we consider a short sentence containing a single subject and verb to be simpler than a long sentence with multiple adjectives, subjects, verbs, and objects, which we consider to be more complicated to annotate.

The user can choose to show the sentences either in order of increasing complexity or in randomized order, which is the default.

Our Annotation Practice page consists of primarily three panels: the leftmost containing the sentence and an option for the user to view any previous submissions for that sentence, the middle

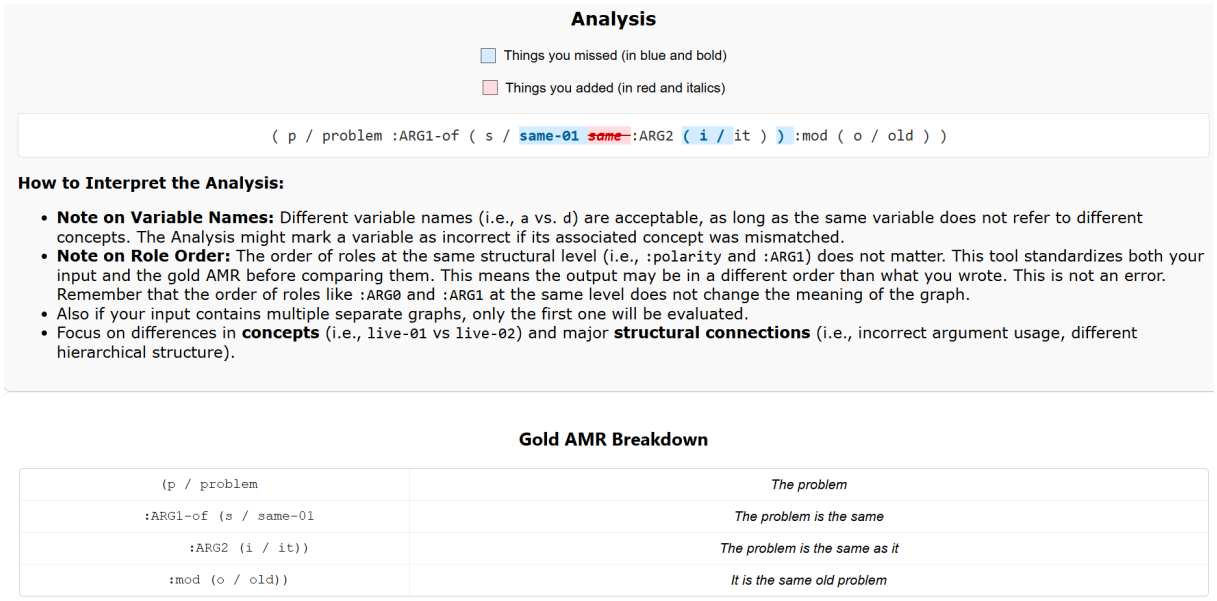


Figure 4: After selecting to show the Gold AMR in the Annotation Practice module, highlighted differences between the user-submitted and gold AMRs are shown as well as a natural-language interpretation of the gold AMR.

panel containing the textbox to submit the AMR annotation for the sentences, and the rightmost panel with options to view the gold AMR and the explanation of the contents of the gold AMR (see Figure 3 for an example). The explanation conceptually walks through the gold AMR graph's structure, highlighting the main concepts and roles in the context of the original sentence. We generate these explanations by hand for each of the 25 gold AMR graphs.

Finally, users can view a direct comparison between their submission and the gold AMR, which we show in a pane called Analysis at the bottom of the Annotation Practice page. Additionally, in the same pane, to aid users who are new to PENMAN notation, a Gold AMR Breakdown provides a natural-language explanation for each line of the gold AMR (Figure 4). This Gold AMR Breakdown is also generated by hand.

When the user input contains significant syntax errors, such as mismatched parentheses that make it difficult to parse the AMR, the submission is unable to be compared to the gold AMR via SMATCH. Thus, we show an error message to guide the user towards correcting the formatting of their input. Reasons for the AMR being invalid include the presence of mismatched parentheses, omitted or misspelled roles (such as forgetting `:` in front of `:ARGn`), and errors regarding concept variables (the letters that label concepts, such as `s` in `s / soul`). The most common variable errors are using duplicate variables for different concepts or omitting them entirely.

3.2.2. Backend

We make calls to the Google Sheets API in order to analyze user submissions.

The backend of the Analysis feature, which compares differences between the gold AMR and user submission, first checks for proper syntax by transforming the user input into PENMAN notation (Matthiessen and Bateman, 1991). Then, in order to verify that the order of roles is not mistakenly marked incorrect, we alphabetically sort all sibling role blocks (roles of the same structural level) in order, to easily find role matches between the submission and gold AMR. Then, we map the concept variable markers between the two graphs to ensure differences in variable naming are not counted as errors, since variable names are arbitrarily decided.[2] This mapping is executed sequentially without structural context, which is a naive approach that works efficiently for the purposes of providing feedback to users, but could be optimized to be more robust as a variant of SMATCH.

After normalization, we compare the tokenized strings of the two AMRs using the `jsdiff` library.[3] The resulting differences are then displayed in the following format: parts of the gold AMR that the user missed are shown in blue and bold, while the extra parts from the user's submission are showed in red and crossed-out italics. We note that using jsdiff is efficient for the sentence-level AMRs con-

[2]For example, `(f / funk)` and `(x / funk)` are equivalent AMRs though different variable markers are used for the `funk` concept.

[3]https://github.com/kpdecker/jsdiff

sidered here, but more scalable graph matching algorithms may be needed to handle much larger document-level graphs.

4. Pilot Studies

We conduct two pilot studies to evaluate the performance and effectiveness of TrAinMR. In particular, we set out to examine whether using TrAinMR leads to an improvement in a user's annotation quality, as measured by (1) SMATCH score against the gold reference, and (2) the user's perceived knowledge of AMR.

To answer these questions, each pilot study asks five annotators to complete the Tutorial and Annotation Practice modules of TrAinMR. We analyze their SMATCH scores against gold annotations and collect feedback on the user experience through post-study surveys.

4.1. Set-up

Each pilot study has five human annotators who are college students and are fluent in English, with little to no prior exposure to AMR, and zero annotation experience with AMR. The study instructions are as follows, provided orally:

- Read the Tutorial page
- Read the instructions at the top of the Annotation Practice page, then write AMR annotations for all 25 sentences
- Only one submission to each sentence is required, but we encourage additional attempts
- Rely on the Tutorial page and other resources linked from the page

After completing the task, annotators are also encouraged to fill out a feedback survey which includes a reflection on what aspects of TrAinMR they find most helpful during the annotation process, a quantitative rating of the utility of TrAinMR, and comments on what parts of the AMR annotation process they find particularly difficult.

In the first pilot study, we present the sentences in the Annotation Practice module in order of increasing complexity, as determined by the length of the sentence and the density of topics such as inverse relations and modals. Annotators also automatically see the gold AMR, explanation, breakdown, and highlighted differences after submitting their annotation. In this first pilot study, we do not track from which annotator each annotation is produced.

In the second pilot study, the study instructions remain the same as in the first pilot study. The study design is also kept the same, except for four logistical changes. First, the order of sentences is randomized per user, in order to assess whether annotator's abilities improve during the training process. The sentences are presented in random order generated using the Fisher-Yates shuffle algorithm (Durstenfeld, 1964). Second, the gold AMR and explanation are hidden by default and only appear after being selected by the user. Third, users are tracked using local storage. This means we can follow the trajectory of an individual user and analyze their performance. Finally, based on feedback from the first pilot study, an additional example is added to the inverse roles section in the Instructions, and reminders regarding parentheses-related syntax are added to the Tutorial and Annotation Practice pages.

4.2. SMATCH Score Results

In the first pilot study, the average SMATCH score (Cai and Knight, 2013) of all 210 annotations against gold standards is 0.669, with a standard deviation of 0.239. It is important to note that these scores exclude annotations that are unable to be evaluated by SMATCH. There are a total of 49 out of 210 annotations that produce this error. An extreme example is the sentence "He can't seem to help himself from apologizing for anything and everything", which only has two valid annotations that are used to compute the SMATCH score. The most common causes of parsing errors are mismatched parentheses and the usage of duplicate variable names for different concepts. For the first pilot study, we find that 16 out of the 25 sentences have an average SMATCH score between 0.600 and 0.800, and there is a small, negative correlation between sentence complexity and average SMATCH scores per sentence (R^2 = 0.198), suggesting that more complicated sentences are indeed harder to annotate.

Annotator ID	Average SMATCH	Variance
Annotator 1	0.434	0.012
Annotator 2	0.643	0.059
Annotator 3	0.497	0.036
Annotator 4	0.723	0.050
Annotator 5	0.645	0.033

Table 1: Average SMATCH and Variance per annotator in the second pilot study.

In the second pilot study, the average SMATCH scores (Cai and Knight, 2013) of all annotations against gold standards is 0.629, with a standard deviation of 0.229. The individual annotator scores and standard variance can be seen in Table 1.[4]

[4]Note that these scores exclude annotations that are unable to be parsed by SMATCH. There are a total of 80 out of 215 annotations that produce this error. The most common reason for this error is the presence of duplicate variable nodes.

Each annotator is assigned a unique, anonymous identifier (Annotator 1, Annotator 2, etc.) that is used consistently throughout the analysis. We see that Annotator 1 has the most consistent performance, as they have the lowest variance, while Annotator 2 exhibits the most varied performance. Individuals learn at different speeds and gain varying proficiency levels. Performing individual analyses reflects this reality and provides a guide for improving the training schema. For example, this data could advocate for an adaptive learning system that provides supplementary practice to annotators with lower scores, while providing more challenging examples to those who appear to learn more quickly.

Given that the second pilot study shows the sentences to annotators in a randomized order, we assess whether the average SMATCH score against the gold reference increases as the annotators practice more. We measure learning (as seen in Figure 5) by viewing annotator performance in the order of sentences that they annotated. In this second pilot study, Annotators 1-4 show a positive correlation between SMATCH score and annotations made over time. Annotator 1 and Annotator 4 show the highest correlation, with an R^2 value of 0.276 and 0.315 respectively, suggesting improvement in AMR annotation via TrAinMR. In contrast to the other annotators, Annotator 5 is able to achieve high SMATCH scores since the beginning of their annotation process, suggesting their learning from the Tutorial page. As a result, Annotator 5's performance shows a weak negative correlation (R^2 value is less than 0.1). We note that these R^2 values are from a simple linear fit and serve as a descriptive trend. Since we do not explicitly model variations in sentence difficulty nor assume a linear learning pattern, we place more emphasis on interpreting these results qualitatively.

The average SMATCH scores across all annotations between the first and second pilot study are relatively similar (0.669 and 0.629 respectively). These SMATCH scores indicate that annotators without previous exposure to AMR can produce annotations that show moderate overlap with gold-standard graphs after use of TrAinMR. While these performance scores are lower than the inter-annotator agreement score ranging from 0.71 to 0.89 SMATCH (Wein, 2025), they suggest that TrAinMR has the potential to help provide foundational knowledge for complex annotation tasks. However, this interpretation is limited given the small pilot size and number of invalid annotations.

4.3. Feedback Survey Results and Error Analysis

In the feedback survey with nine responses (one annotator in the first study did not respond to the form, hence nine total responses instead of ten), annotators report their prior knowledge of AMR annotation on a scale of 1 (None) to 5 (Expert). The responses show that most annotators are novices, with a mode of 1 and a mean score of 1.550. Eight out of nine participants rate their prior exposure as a 1 or a 2, and one rates their proficiency as a 3. When asked if TrAinMR helped them learn AMR annotation (1=Not at all, 5=Very much), the feedback is positive. The mode is 4 and the mean score is 3.330. This indicates that TrAinMR can help expose users to AMR in an interactive way, compiling the most necessary information to get one started in AMR annotation.

Annotators also note that time spent reading the Tutorial lasted approximately 20-30 minutes for each user. The fastest completion time for annotating the 25 sentences is approximately 1 hour and 13 minutes and the longest completion time is approximately 2 hours and 12 minutes. One participant in particular notes that the annotations took less time as they completed more practice.

Annotators state that they have the most trouble with finding the root of the sentence, distinguishing between roles and relations with similar uses, understanding inverse roles, and finding the correct PropBank frames. For example, for the sentence "In recently years, Finland has been keeping an obvious trade surplus in this region," we see that the annotators have difficulty identifying the correct concepts, including distinguishing when to use a PropBank frame or a word (`obvious` v.s. `obvious-01`).[5]

We identify several other common errors in the annotations. The sentence with the lowest average SMATCH score in the first pilot study is "The Japanese delegation will fly to Beijing on the 2nd" with SMATCH of 0.415. Although there are five annotators in this study, they are encouraged to resubmit their work, resulting in a total of eight submissions for this particular sentence. Of these eight submissions, four produce parsing errors and are thus excluded from the average score calculation. Out of the valid submissions, annotators appear to have trouble with the exact syntax of the `:name` role and wikification. Annotators also show difficulty in identifying the correct role for the locations in the sentence, mixing between `:destination`, `:mod`, and `:ARG1-of`.

The sentence with the lowest average SMATCH

[5]Note that the the original sentence contains "recently" though "recent" would be grammatical; we present the sentence as written to the annotator.

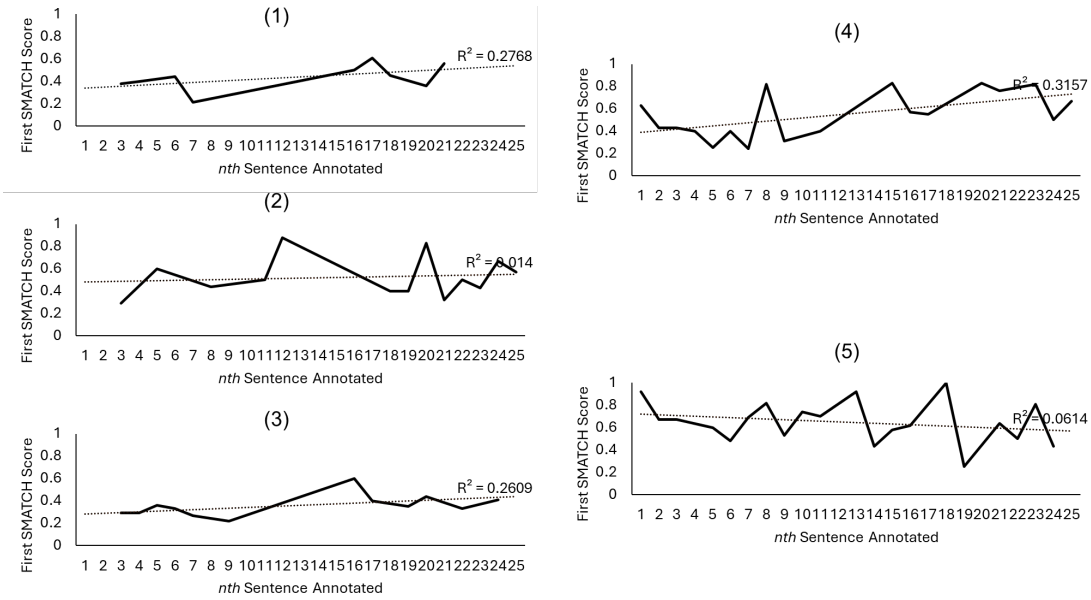


Figure 5: The performance of each annotator from the second pilot study is shown over time. The x-axis represents the nth sentence annotated, and the y-axis represents the SMATCH score of the first annotation submission per sentence. The first submission for each sentence per annotator is being compared. Missing values are included. Annotators 1-5 are represented by the labeled graphs.

score in the second pilot study is "a group of people of nine professions" with a SMATCH score of 0.373. Two out of five annotations produce parsing errors for this sentence, and thus are excluded from the score. The annotators seem to struggle with using reifications, where none of the annotators use `have-org-role-91` in their submission for this sentence.

Our pilot studies also reveal annotators' trouble in identifying the correct root concept. In regards to the first study, in the sentence "The policy is a matter of national sovereignty and security," six out of eight submissions exclude the relation `:domain`. Half of the submissions correctly identify `matter` as the root concept, of which two of them are incorrectly `matter-01` and `matter-02`. The other submissions have `policy` and `be-03` as the root concept. There are similar patterns of annotators confusing PropBank frames with regular words, such as in "During a time of prosperity and happiness, such a big earthquake suddenly struck," where the concept `happiness` in the gold AMR is written as `happy-01` and `happy-02` in two of the submissions. We also note that annotators seem to forget to add demonstrative pronouns sometimes such as "this," as seen in submissions for the sentence "He felt that, there were more new competitors from our country participating in this competition." In the second study, another example of annotators having a difficult time identifying the root concept is for the sentence "The acquisition is expected to be completed before April," where the root concept is `expect-01`. Eight out of ten submissions correctly identify this, however one annotator's initial submission uses `complete-01` as the root instead.

Another key finding from our analysis is related to concept selection. For instance, for one sentence, six out of nine submissions in the second study use the adjective `happy` while the gold AMR uses the noun `happiness`. While the annotators successfully identify the main semantic concept in this case, they struggle with nominalization. This suggests that our training is effective at conveying the main principles of AMR, but specific linguistic details like nominalization is a challenge for novices. This insight informs how to improve future versions of the training tutorial.

Perhaps unsurprisingly, we find annotators are unable to identify specific roles or relations that are not included in the Tutorial. For example, in the sentence "Why is it so hard to understand?", the gold AMR uses `:degree`, a relation that is not included in the Tutorial, for the concept `so`. Two out of the ten annotations (five being resubmissions) in the second study include the concept `so`. Annotator 1 connects the concept `so` to the relation `:manner` as their first submission. Annotator 5 connects the concept `so` to the relation `:quant` in their resubmission, whereas they omit `so` in their initial submission. The other annotators omit the concept `so` and `:quant` in their submissions.

Finally, when asked which resources users find most helpful, annotators could select multiple options. The responses indicate that a combination of resources is valuable for the learning process. Seven out of nine total responses across both stud-

ies indicate that the Tutorial page is most helpful, and six out of nine responses state that the explanation of the gold AMR is most helpful. External resources such as referencing PropBank frames and AMR guidelines, and the gold AMR itself, are also indicated as most helpful by five respondents. This indicates that users find TrAinMR particularly effective when supplemented with examples and further documentation.

5. Conclusion

In this paper, we present TrAinMR, an open-source, web-based tool designed to address the difficulty in creating large-scale AMR corpora due to the time and expertise needed to train new annotators. TrAinMR provides a foundational tutorial and an interactive practice environment that can be reused as a training schema. Its immediate feedback loop allows novices to compare their attempts to a gold standard AMR, helping them to learn the accurate application of AMR's guidelines across annotations. By having a self-contained learning module with original examples and explanations, TrAinMR makes the complex task of AMR annotation training more accessible to a broader audience.

We also review the results to two pilot studies evaluating how effective TrAinMR is. We find that many annotators make two common mistakes, which are missing or mismatching parentheses, and using the same variable label for different concepts. Other common mistakes include using incorrect PropBank frames and different roles and relations.

The average SMATCH score across all annotations in pilot study one and two are 0.669 and 0.629 respectively, which provides an initial indication of the potential utility of our training tool. We observe a trend in four annotators in the second pilot study where their performance is positively correlated with the number of annotations completed. Additionally, another annotator shows an increase in performance on complex sentences, which may suggest they become more proficient at writing annotations during their continued use of TrAinMR.

Common error patterns in the two pilot studies also highlight the importance of mitigating the effects of annotator subjectivity. Annotators may disagree with certain parts of an AMR graph depending on their own interpretation of words, such as finding a different verb in a sentence to be the root of an AMR, which can also be due to language ambiguity (Wein, 2025). We see this in the user annotations in our study, where places like the root concept and certain roles are different from the gold AMR, but the overall meaning of the AMRs are not dissimilar.

Future work may include enhancing TrAinMR with an LLM-based feedback feature to provide more interactive, natural language explanations of differences between a user's submission and the gold AMR for a large set of sentences. Our error analysis also informs the development of other training curriculum. If additional documentation on approaches to AMR training emerges, TrAinMR could be compared with other training approaches in future work. Finally, expansions to TrAinMR could support AMR extensions and other languages in order to make semantic annotation beyond AMR more accessible.

Limitations

Our study is limited to AMR annotation of texts in the English language, which may not generalize to other languages or annotation frameworks. Additionally, our sample size is reduced due to some submitted annotations being invalid AMR graphs, meaning we are unable to calculate SMATCH scores with them. We use pilot studies to iteratively improve and validate the utility of our tool; future pilot studies may examine how annotators trained using TrAinMR may perform over a baseline, self-training for the same amount of time, or assess how whether TrAinMR enables improved quality control of annotations.

Acknowledgments

We thank anonymous reviewers and members of the Amherst College NLP lab for their feedback. This work is supported by the Amherst College HPC, which is funded by NSF Award 2117377.

6. Bibliographical References

Shafiuddin Rehan Ahmed, Jon Cai, Martha Palmer, and James H. Martin. 2024. X-AMR annotation tool. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 177–186, St. Julians, Malta. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and

Nathan Schneider. 2019. Abstract meaning representation (amr) 1.2.6 specification.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract Meaning Representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583, Marseille, France. European Language Resources Association.

Jon Cai, Shafiuddin Rehan Ahmed, Julia Bonn, Kristin Wright-Bettner, Martha Palmer, and James H. Martin. 2023. CAMRA: Copilot for AMR annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 381–388, Singapore. Association for Computational Linguistics.

Jon Cai, Kristin Wright-Bettner, Zekun Zhao, Shafiuddin Rehan Ahmed, Abijith Trichur Ramachandran, Jeffrey Flanigan, Martha Palmer, and James Martin. 2025. LiDARR: Linking document AMRs with referents resolvers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 426–435, Vienna, Austria. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Wei-Te Chen and Will Styler. 2013. Anafora: A web-based general purpose annotation tool. In *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 14–19, Atlanta, Georgia. Association for Computational Linguistics.

Richard Durstenfeld. 1964. Algorithm 235: random permutation. *Communications of the ACM*, 7(7):420.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator": Limits of LLMs as analyzers of Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8250–8263, Singapore. Association for Computational Linguistics.

Ulf Hermjakob. 2013. Amr editor: A tool to build abstract meaning representations. *Marina del Rey, CA. USC Information Sciences Institute*.

Kevin Knight, Bianca Badarau, Laura Banarescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, et al. 2020. Abstract Meaning Representation (AMR) Annotation Release 3.0. Technical Report LDC2020T02, Linguistic Data Consortium, Philadelphia, PA.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Mary Martin, Cecilia Mauceri, Martha Palmer, and Christoffer Heckman. 2020. Leveraging non-specialists for accurate and time efficient amr annotation. In *Proceedings of the LREC 2020 Workshop on" Citizen Linguistics in Language Resource Development"*, pages 35–39.

Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.

Christian Matthiessen and John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter Publishers.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Nathan Schneider, Tim O'Gorman, and Jeffrey Flanigan. 2015. The logic of amr practical, unified, graph-based sentence semantics for nlp.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim

O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Shira Wein. 2025. Ambiguity and disagreement in Abstract Meaning Representation. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 145–154, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Shira Wein and Nathan Schneider. 2024. Assessing the cross-linguistic utility of abstract meaning representation. *Computational Linguistics*, 50(2):419–473.

Named Entity Recognition for Persian Literary Text: A Case Study on The Little Prince

Minoo Nassajian, Joakim Nivre, Daniel Zeman
Charles University, Faculty of Mathematics and Physics
Prague, Czech Republic
{nassajian, zeman}@ufal.mff.cuni.cz

Uppsala University, Department of Linguistics and Philology
Uppsala, Sweden
joakim.nivre@lingfil.uu.se

Abstract

Existing Persian Named Entity Recognition (NER) research has focused predominantly on news and social media domains, leaving literary texts—with their distinct linguistic characteristics—virtually unexplored. This paper addresses this gap by developing a new literary NER corpus using the Persian translation of *The Little Prince* story and evaluating existing state-of-the-art Persian NER tools on this corpus, trained exclusively on news and social media corpora. Our analysis reveals significant performance degradation on literary text, identifying systematic errors related to narrative-specific entities, metaphorical language, and discourse structures that challenge conventional NER approaches.

Keywords: Persian NER, Persian NLP, Persian computational literary

1. Introduction

Named entity recognition (NER), as a fundamental sub-task within the field of natural language processing (NLP), is concerned with the automatic identification and classification of entities in unstructured texts into predefined semantic categories including but are not limited to, the names of persons, organizations, and geographical locations (Grishman and Sundheim, 1996; Nadeau and Sekine, 2007).

As an important component in the NLP pipeline, the utility of NER extends far beyond mere identification. It serves as a foundational pipeline for a wide array of downstream applications and advanced research domains. Its contributions are integral to systems for information retrieval (Mandl and Womser-Hacker, 2005; Petkova and Croft, 2007; Guo et al., 2009), question answering (Mollá et al., 2006, 2007; Khalid et al., 2008), and machine translation (Babych and Hartley, 2003; Vu et al., 2020; Xie et al., 2022), where disambiguating entity references is essential for accuracy. Furthermore, NER provides a crucial input for more complex linguistic tasks such as relation extraction (Feldman and Rosenfeld, 2006; Gundluru et al., 2022), co-reference resolution (Zhao, 2009; Clark and Manning, 2016), and automatic text summarization (Khademi and Fakhredanesh, 2020; Berezin and Batura, 2022; Khan et al., 2024).

The majority of NER research has been conducted on a narrow range of domains—primarily news (Shabat and Omar, 2015; Al-Ash and Wibowo, 2018; Ruokolainen et al., 2020; Chavan and Patil, 2024), encyclopedic text (Balasuriya et al., 2009; Nothman et al., 2013; Li et al., 2019), and, more recently, social media (Aguilar et al., 2017; Nie et al., 2020; Yu et al., 2023) and clinical texts (Chen et al., 2015; Kundeti et al., 2016; Bose et al., 2021; Goyal and Singh, 2025)—where entity mentions are often explicit, capitalized, and anchored to real-world referents.

Unlike the clear, explicitly named entities common in journalistic or formal texts, literary entities often possess an ontological ambiguity—they may be fictional constructs rather than real-world referents (Bamman et al., 2019). Furthermore, their introduction and reference within the narrative are frequently implicit, relying on descriptive epithets, figurative language, or discourse-driven uniqueness, often without capitalization or explicit proper names (Silva and Moro, 2024). As a result, NER systems trained on surface-level regularities often fail to capture entities that are central to narrative meaning.

This paper investigates Persian NER in the literary domain through a detailed case study of *The Little Prince* story. We construct a Persian NER corpus, do pre-processing steps using a standard NLP pipeline, and annotate it according to a linguistically grounded framework that distinguishes strong named entities (e.g., proper names and rigid designators) from weak named entities (e.g., definite descriptions and discourse-unique referents) proposed by Borrega et al. (2007). We then evaluate three state-of-the-art Persian NER systems and compare their outputs to annotations produced via controlled in-context prompting of ChatGPT, which was instructed to follow the same theoretical guidelines. This work is part of a broader effort aimed at

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 54–64
11 May 2026.

developing a Persian Uniform Meaning Representation (UMR) corpus based on the existing Persian AMR resource. Reliable identification of named entities is a necessary prerequisite for UMR annotation, as entity types contribute to argument interpretation, animacy distinctions, and discourse-level coreference. The literary NER corpus introduced in this paper therefore serves as an enabling resource supporting ongoing Persian UMR development.

The remainder of this paper is organized as follows: Section 2 reviews foundational and contemporary literature. Section 3 and 4 introduce the annotated corpus and the domain-specific annotation framework developed for this work. Section 5 explains the NER tagset used in this research. Section 6 outlines our experimental methodology. Finally, section 7 presents the empirical results and a systematic error analysis. Section 8 concludes with a summary of contributions and future research avenues.

2. Related Works

NER emerged as a core task in information extraction with early benchmarks such as MUC, ACE, and CoNLL, which were largely grounded in newswire and broadcast data. These datasets shaped both annotation guidelines and modeling assumptions, privileging capitalized proper names, geopolitical entities, and organizations, and favoring relatively flat syntactic structures. Consequently, most high-performing NER systems were trained on news-domain corpora and implicitly encode assumptions about journalistic style, entity distributions, and referential clarity.

Subsequent work has demonstrated that NER models trained on news data degrade substantially when applied to other domains, including social media, historical documents, and literary texts. Augenstein et al. (2017) showed that domain shift leads to sharp drops in precision and recall, especially for entities that are frequent but stylistically atypical. Literary texts pose additional challenges, including metaphor, personification, and character roles that function as names within the story world. In response to this gap, a few recent studies have been undertaken to expand the methodological toolkit of computational literary analysis.

van Dalen-Oskam et al. (2014) introduced the Namescape project as one of the earliest systematic efforts to apply NER to large-scale literary corpora. Motivated by the needs of literary scholarship rather than traditional information extraction, their work emphasized the importance of identifying both real-world and fictional entities in literary texts, highlighting challenges such as referential ambiguity, name variation, and the prevalence of non-canonical entities that are typically absent from newswire corpora. They described the construction of Dutch literary corpora derived from digitized novels and historical texts, alongside initial annotation guidelines tailored to literary discourse, but did not primarily focus on model evaluation or benchmark performance. Building on this foundation, de Does et al. (2017) extended the Namescape initiative by producing fully annotated gold-standard literary corpora and conducting systematic experiments with supervised NER models trained on these texts. Using Conditional Random Fields (CRF) (Wallach, 2004) and Support Vector Machine (SVM) (Hearst et al., 1998) models, they demonstrated that models trained on in-domain literary data substantially outperform systems trained on news-domain corpora, reporting large improvements in F_1 score and confirming the inadequacy of news-based NER assumptions for literary language.

LitBank corpus is another computational literary effort on NER, introduced by Bamman et al. (2019). This is an annotated corpus of 210,532 tokens drawn from 100 English literary works. The annotation follows the ACE 2005 guidelines (Consortium et al., 2005), but crucially extends them to literary-specific phenomena by annotating both named and common noun phrases, allowing nested entities, and incorporating imagined or fictional locations and characters.

Complementary to corpus creation, Dekker et al. (2019) provide an extensive evaluation of off-the-shelf NER systems on literary novels, focusing on the task of social network extraction from fiction. Their study evaluates four widely used NER systems on 40 English novels (20 classic, 20 modern), manually annotating approximately one chapter per novel for PERSON entities.

In a recent study, Silva and Moro (2024) introduce a manually annotated corpus designed specifically for NER in Portuguese literary texts. The corpus comprises 25 public-domain literary works from Brazilian and European Portuguese literature, each contributing approximately 5,000 tokens.

Prior research on Persian NER has predominantly focused on developing datasets and systems for non-literary domains such as news (Poostchi et al., 2016; Shahshahani et al., 2018), social media, and Wikipedia (Asgari-Bidhendi et al., 2021; Aghajani et al., 2021a). Consequently, the performance of existing tools—trained on these corpora—on the distinct linguistic and stylistic features of full-length literary fiction remains largely unexamined. Furthermore, the field lacks systematic annotation guidelines grounded in linguistic theory specifically tailored for the Persian literary domain. This constitutes a significant gap, as a systematic evaluation of Persian NER on narrative texts is essential for advancing computational literary studies and digital humanities in Persian.

The central contribution of the present work is the creation of the first manually annotated Persian literary NER corpus, which is released as a publicly available resource to support future research. The corpus is based on the Persian translation of *The Little Prince* story and it complements ongoing work on Persian AMR-to-UMR conversion by providing consistent entity annotations required for semantic role interpretation and discourse tracking. Using this gold-standard dataset, we further evaluate 3 existing state-of-the-art Persian NER systems (ParsNER (Team, 2021), Shekar (Amirivojdan, 2025), and ParsTwiNER (Aghajani et al., 2021b)) and analyze their performance in the literary domain. In the next sections, we will explain about the statistical information of the corpus and NER tagsets used in this research.

3. Corpus

Our research employs the Persian Abstract Meaning Representation (PAMR) corpus, developed by Takhshid et al. (2024). This corpus is based on the Persian translation of *The Little Prince* and it contains no prior named entity annotations. The corpus consists of 1,562 sentences (14,427 tokens) with a vocabulary of 3,520 unique word forms; sentence lengths range from 1 to 65 tokens. We preprocessed the text by normalizing and tokenizing it using the Stanza toolkit (Qi et al., 2020). The corpus was then formatted according to the BIO tagging scheme to facilitate subsequent manual named entity annotation and sequence-labeling experiments.

4. Annotation Guideline

To establish annotation guidelines suited to literary texts, we ground our approach in the theoretical framework of Borrega et al. (2007), which argues that named entities should not be defined solely by formal properties (e.g., capitalization[1]), but rather by referential behavior in discourse. Central to this framework is the distinction between Strong Named Entities (SNEs) and Weak Named Entities (WNEs), and the notion of Trigger Words (TWs), which play a decisive role in identifying entities that lack canonical proper names. In the following sections, we will discuss how we annotate *The Little Prince* story using this distinction.

4.1. Strong Named Entities

SNEs are expressions whose referent can be identified independently of discourse context, typically through a conventional name. In *The Little Prince* story, there are some words that are considered as SNEs. For example, the word "Earth" does not refer to an abstract concept or a generic surface in this text; instead, it denotes the planet Earth as a unique astronomical entity. Similarly, "Africa" refers to a specific continent with a fixed real-world referent.

4.2. Weak Named Entities

WNEs are nominal expressions that function as unique referents within a specific discourse due to contextual anchoring, despite lacking external uniqueness. Their detection is operationalized via Trigger Words (TWs). TWs are common nouns that, as semantic heads of noun phrases, signal potential entities. TWs require discourse-specific modifiers—such as definiteness, relational adjectives, or narrative uniqueness—to trigger entity interpretation. A noun phrase is annotated as a WNE if it meets these criteria:

- Head: Contains a Trigger Word
- Uniqueness: Referent is unique in the discourse
- Trackability: Referent is maintained across mentions
- Role: Referent holds a distinct narrative or semantic role

A clear example of a WEN is found in example (1). The common noun ("flower") functions as the Trigger Word. Through the discourse, this nominal expression becomes a unique and trackable entity. This occurs as the definite reference ("the flower") first establishes specificity, after which subsequent mentions ("that flower") maintain a coherent coreferential chain. Ultimately, by functioning as a central character within the narrative, it gains significant semantic weight. Thus, despite its common noun head, the phrase satisfies all criteria for annotation as a WEN.

(1) "I am not a weed," the flower replied.[2]

This annotation principle applies consistently to other WNEs in *The Little Prince*, including "the king", "the fox", and "the geographer". Each of these nominal expressions, headed by a common noun Trigger Word, similarly acquires unique referential status through definiteness, narrative anchoring, and sustained discourse presence.

[1] Unlike Latin-script languages, Persian orthography does not employ capitalization to distinguish proper nouns.

[2] In this work, WNEs are limited to noun phrases. Pronouns were not annotated as WNEs, even when they refer to uniquely identifiable entities, as they function primarily as coreferential expressions rather than entity-denoting mentions.

5. NER tagset

The named entity inventory employed in this study encompasses a deliberately broad range of semantic categories, some of which are quite specific: **Person, Animal, Plant, Organization, Location, Product, Publication, Nationality, Time, and Planet.**[3] Following the standard BIO tagging convention, tokens that do not belong to any named entity span were assigned the label **O**, which represents non-named-entity tokens. Table 1 shows the entity distribution in the Persian corpus of *The Little Prince*.

Entity Type	Entities
Person-W	261
Animal-W	43
Plant-W	45
Planet	33
Location	24
Time	17
Product	8
Nationality	3
Publication	2
Organization	1
Total	**437**

Table 1: Entity Distribution in persian corpus of *The Little Prince*

5.1. Inter-Annotator Agreement

To further assess the reliability of the gold annotations, a second linguist independently annotated a subset of 100 sentences selected to include diverse named entity types. Inter-annotator agreement was measured at the token level using Cohen's κ coefficient. The resulting agreement score was $\kappa = 0.99$, indicating almost perfect agreement. Disagreements were resolved through adjudication with a third annotator, and the finalized labels were incorporated into the gold dataset.

6. Experimental Setup

In this work, we evaluate the robustness of existing Persian NER tools on the preprocessed Persian corpus of *The Little Prince*. In the next parts, we will explain the pre-processing steps and Persian NER tools.

[3]Within the annotation schema, the SNE/WNE distinction is implemented via a dedicated suffix. The tag -W is appended to standard BIO tags to flag a Weak Named Entity (e.g., B-person vs. B-person-W)

6.1. Preprocessing and Input Standardization

The Persian Little Prince corpus was preprocessed using the Persian pipeline in Stanza, including text normalization and tokenization. The processed corpus was then converted into BIO sequence-labeling format for named entity annotation and evaluation. Since our goal is to assess the behavior of existing Persian NER taggers on literary text, we treat the entire corpus as test-only data and report model performance over all sentences.

6.2. Automatic Annotation Systems

To assess how existing Persian NER systems perform on literary narrative text, we evaluate three state-of-the-art transformer-based Persian NER tools—ParsNER (Asgari-Bidhendi et al., 2021), Shekar (Amirivojdan, 2025), and ParsTwiNER (Aghajani et al., 2021b)—all of which were originally developed and trained on non-literary corpora such as news, Wikipedia, or social media. These systems differ substantially in model architecture, training data, and tagset granularity, reflecting their original design goals and target domains.

In addition, we evaluate a guideline-driven LLM-based annotator (ChatGPT-5.2, OpenAI), prompted to follow a linguistically motivated NER framework distinguishing SNEs and WNEs, along with a fixed set of 20 annotated examples drawn from different chapters of the text. These examples were selected to cover diverse entity types and narrative phenomena typical of literary discourse. Moreover, These sentences were excluded from the evaluation set to prevent prompt–test overlap and ensure unbiased performance measurement. Remaining unseen portion of the corpus were automatically annotated and subsequently reviewed by authors, who manually corrected incorrect labels and added missing entity annotations to address both precision and recall errors. This process ensured consistency with the annotation guidelines and resolved systematic errors, particularly in the identification of weak named entities and discourse-dependent references. The resulting annotations constitute a manually validated gold dataset.

6.3. Label Mapping for Cross-System Comparability

Because the compared systems differ substantially in tagset and granularity, we conduct evaluation using two complementary protocols. In the primary evaluation ("shared tagset"), all outputs were mapped to a common label inventory consisting of

PER, LOC,[4] ORG, DAT, and O. Dataset-specific labels not belonging to this inventory (e.g., MON, PCT, PRO; POG; or fine-grained narrative categories) were mapped to O.

In addition to the shared-tagset comparison, we perform a second evaluation protocol focusing on ChatGPT's guideline-driven annotation scheme. Unlike the automatic taggers, ChatGPT was instructed to annotate literary entities using a fine-grained label set (introduced in NER tagset section) that distinguishes both entity category and referential status (Strong vs. Weak NEs).

6.4. Metrics

We report entity-level Precision (P), Recall (R), and F_1 using BIO span evaluation as implemented in the seqeval library. Micro-averaged F_1 over entity types is used as the main metric due to label imbalance and the predominance of O tokens, while macro-averaged scores and per-class F_1 are also reported for interpretability. Confusion matrices (row-normalized) are provided to characterize systematic confusions as well.

7. Results and Error Analysis

Table 2 presents entity-level Precision, Recall, and F_1-score for the three Persian NER taggers (ParsNER, ParsTwiNER, and Shekar) and the LLM-based guideline-driven annotation system (ChatGPT) on the Persian Little Prince corpus. All outputs were normalized into a unified tagset (PER/LOC/ORG/DAT), and evaluation was performed using BIO entity spans.

Across the automatic systems, ParsNER achieves the strongest performance among the supervised taggers (micro-$F_1 = 0.52$), while ParsTwiNER and Shekar degrade sharply (micro-$F_1 = 0.21$ and 0.09), indicating severe domain shift from their original training corpora (news/Wikipedia/Twitter) to literary narrative. In contrast, ChatGPT substantially outperforms all supervised taggers, reaching micro-$F_1 = 0.90$, driven primarily by very high recall (0.92). This confirms that a guideline-constrained LLM annotation protocol is particularly suitable in literary settings, and NER models trained on news, Wikipedia, or Twitter do not transfer reliably to Persian literary narrative, where entity mentions are often discourse-dependent, stylistically variable, and frequently realized as non-canonical noun phrases.

The per-class analyses, presented in Tables 3, 4, 5, and 6, illustrate the nature of domain mismatch. ParsNER achieves high precision for LOC (0.93) but only moderate recall (0.39), indicating that it detects a subset of easily recognizable locations but fails on narrative or ambiguous location mentions. ParsTwiNER, optimized for social media, demonstrates catastrophic performance on literary character recognition (PER recall $= 0.08$, $F_1 = 0.13$) and completely fails to identify temporal expressions (DAT recall $= 0.00$). While it achieves reasonable precision for locations (0.85), its recall remains low (0.34), suggesting it recognizes only the most conventional location mentions. Shekar, trained on encyclopedic Wikipedia text, shows a different pathology: it achieves moderate recall for dates (0.41) but with near-zero precision (0.05), indicating severe over-prediction of temporal expressions. Most critically, it fails entirely to recognize character mentions (PER recall $= 0.00$, $F_1 = 0.01$), rendering it fundamentally unsuitable for narrative analysis.

System	P	R	Micro-F1	Macro-F1
ParsNER	0.64	0.43	0.52	0.30
ParsTwiNER	0.48	0.13	0.21	0.32
Shekar	0.12	0.07	0.09	0.22
ChatGPT	**0.88**	**0.92**	**0.90**	**0.55**

Table 2: Entity-level performances of Persian NER systems on the *Little Prince* corpus

Class	P	R	F1	Support
DAT	0.20	0.06	0.09	15
LOC	0.93	0.39	0.55	55
ORG	0.20	1.00	0.33	1
PER	0.63	0.46	0.53	255

Table 3: Class-wise performance metrics for the ParsNER system on the *Little Prince* corpus.

In contrast, Table 6 shows that ChatGPT exhibits particularly high effectiveness on the dominant entity categories in the narrative, reaching F_1 scores of 0.97 for PER and 0.82 for LOC, indicating that it successfully detects nearly all character and location mentions. This supports the hypothesis that large language models can generalize entity recognition beyond the news domain and remain highly sensitive to narrative discourse structure. In contrast, DAT is substantially harder ($F_1 = 0.40$), reflecting the ambiguity of temporal expressions in Persian literary text, where time is frequently encoded through discourse-relative adverbs (e.g., tonight, tomorrow, that night) rather than explicit calendar dates, and the low macro-F_1 (0.55) is largely driven by these low-frequency and ambiguous classes. These findings suggest that guideline-driven ChatGPT annotation is particularly suitable for corpus bootstrapping in literary settings, where maximizing entity coverage (recall) is critical for constructing reliable gold datasets.

Based on Figures 1 and 2, we can see a clear performance gap between the two systems, especially

[4] References to celestial bodies (planets) and nationalities within *The Little Prince* were aligned with the LOC label.

Class	P	R	F1	Support
DAT	0.00	0.00	0.00	15
LOC	0.85	0.34	0.49	55
ORG	0.50	1.00	0.67	1
PER	0.33	0.08	0.13	255

Table 4: Class-wise performance metrics for the ParsTwiNER system on the *Little Prince* corpus.

Class	P	R	F1	Support
DAT	0.05	0.41	0.09	15
LOC	0.35	0.24	0.28	55
ORG	0.33	1.00	0.50	1
PER	0.12	0.00	0.01	255

Table 5: Class-wise performance metrics for the Shekar system on the *Little Prince* corpus.

for PER and DAT entities. ChatGPT demonstrates near-ceiling recognition of PER mentions (98.0%), while ParsNER correctly predicts only 57.2% of PER tokens and incorrectly maps a substantial portion to the non-entity class (O; 42.5%). This indicates that ParsNER systematically fails to recover character mentions in narrative discourse, due to genre mismatch and its reliance on news-style naming conventions. For LOC, both models show confusion with O, but ChatGPT achieves substantially higher accuracy (81.5%) compared to ParsNER (42.0%), suggesting stronger contextual inference for spatial references in descriptive passages. Finally, both models show difficulty with temporal expressions (DAT), though ChatGPT retains moderate accuracy (62.9%) whereas ParsNER almost entirely collapses DAT into O (94.3%). Overall, the confusion matrix comparison confirms that ChatGPT is substantially more robust to literary-domain entity realizations, while ParsNER exhibits heavy domain-shift degradation, especially for discourse-driven person references and non-canonical time mentions.

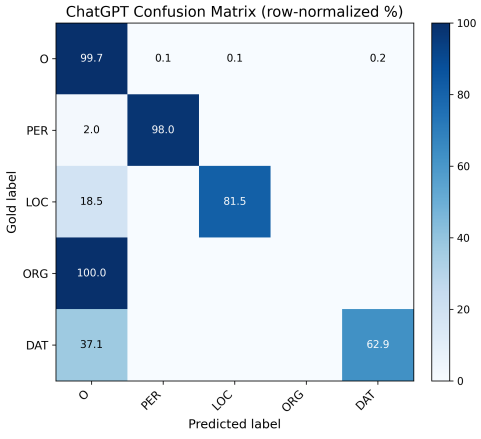


Figure 1: ChatGPT Confusion Matrix (Mapped Tagset)

Class	P	R	F1	Support
DAT	0.30	0.59	0.40	15
LOC	0.82	0.82	0.82	55
ORG	0.00	0.00	0.00	1
PER	0.97	0.97	0.97	255

Table 6: Class-wise performance metrics for the ChatGPT system on the *Little Prince* corpus.

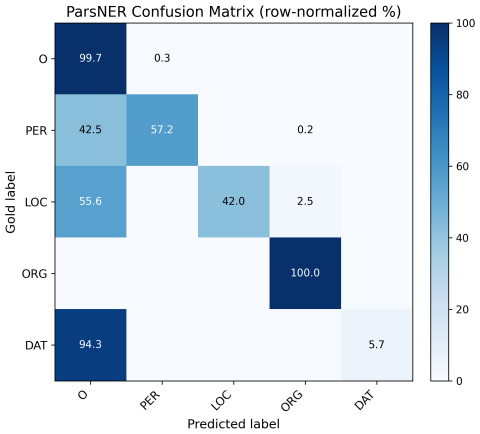


Figure 2: ParsNER Confusion Matrix (Mapped Tagset)

Under Protocol 2, which evaluates performance on the full fine-grained tagset without label mapping, ChatGPT demonstrates exceptionally strong performance on the Persian Little Prince corpus, achieving a micro-averaged F_1-score of 0.87. Table 7 reveals the model's particular proficiency with narrative-specific, discourse-grounded entities. Most notably, it achieves near-perfect recognition of weak person references (person-W) with an F_1-score of 0.97 on substantial support ($n = 255$). This indicates a robust capacity to track characters through implicit, descriptive, and role-based mentions—a core challenge in literary analysis. The model also excels at identifying other distinctive narrative entities, such as speaking animals (animal-W $F_1 = 0.92$) and symbolic flora (plant-W $F_1 = 0.79$), alongside strong performance for conventional categories like location ($F_1 = 0.82$). However, performance remains uneven across the tagset.

Temporal expressions (time $F_1 = 0.40$) and several rare categories with minimal support—such as organization, product, and nationality—yield low or unstable scores. The pronounced discrepancy between the macro-averaged (0.53) and micro-averaged (0.87) F_1-scores underscores this class-wise variance, reflecting the challenge of generalizing across infrequent entity types. Moreover, plant has support $n = 0$, meaning that this label does not occur in the gold annotations; consequently, it cannot receive a meaningful preci-

sion/recall score. These results demonstrate that a guideline-driven LLM is a highly capable annotator for literary NER. Its strength lies in accurately recognizing the discourse-anchored entities—such as characters referred to by description rather than name—that are essential for understanding narrative but are poorly captured by conventional models.

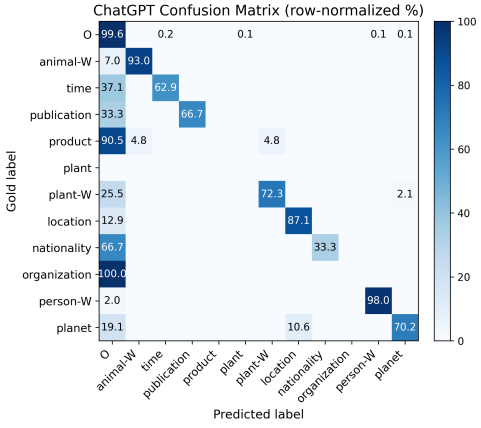


Figure 3: ChatGPT Confusion Matrix (Full Tagset)

Figure 3 shows the row-normalized confusion matrix for ChatGPT under Protocol 2 (full tagset). The model exhibits its strongest separability on discourse-central weak entities, especially person-W (98% correctly predicted) and animal-W (93%), confirming reliable tracking of recurring narrative referents. Performance is also strong for location (87%), while planet is only moderately distinguished (70%), with frequent confusion into O (e.g. the sun[5]) and occasionally location (e.g. the Earth). The main weaknesses occur in categories that are either rare in the corpus or linguistically ambiguous, such as time (substantial confusion into O due to relative temporal expressions), and sparsely represented classes like organization, nationality, and product, which are often collapsed into O[6] Overall, the matrix indicates that ChatGPT is particularly effective for highly salient discourse entities (weak NEs) but less stable on low-frequency or boundary-vague categories.

[5]The *planet* category is used as a general label for celestial bodies mentioned in the narrative, including planets and stars such as Earth and the Sun. This operational definition prioritizes annotation consistency and referential function over strict astronomical classification.

[6]Some entity types appear very rarely or not at all in the gold annotations (for example, the *plant* category has a support of 0), which makes their performance less reliable and explains the unstable patterns observed in the confusion matrix (see Table 7 for support values).

8. Conclusion and Future Work

This paper presented a corpus-based study of Named Entity Recognition in Persian literary text and introduces the first linguistically validated annotated corpus derived from *The Little Prince*, which is made freely available for research.[7] High inter-annotator agreement demonstrates the reliability of the proposed annotation scheme and establishes a solid foundation for future extensions of literary NER resources in Persian.

We also developed an annotation framework grounded in the theoretical distinction between strong and weak named entities, operationalized through a discourse-sensitive tagset designed to capture referential patterns characteristic of narrative texts. Using the gold-standard corpus, we conducted an empirical evaluation comparing three state-of-the-art Persian NER systems with a guideline-driven annotation approach implemented using ChatGPT.

The experimental results reveal a pronounced domain shift challenge: existing pretrained taggers exhibited significant performance degradation when applied to literary text, attributable to mismatches in referential structure, entity realization, and Persian-specific linguistic ambiguity. In contrast, the LLM-based approach achieved substantially higher overall F_1-scores. This suggests that large language models, when guided by formal annotation criteria and supported by human validation, can serve as an effective tool for both corpus construction and the study of discourse-mediated namedness.

Future work may enrich the corpus with complementary linguistic layers, such as coreference chains, nested entity structures, and quotation attribution. These additions, which interact fundamentally with narrative entities, would enable more comprehensive modeling of Persian literary discourse.

9. Ethics Statement

We are not aware of any ethical concerns related to this work. The corpus was manually annotated as part of academic research. In addition to the primary annotator, independent linguistically trained annotators contributed to annotation validation and agreement assessment. All annotation work was conducted voluntarily for research purposes, and no sensitive or personal data were involved, as the corpus is based on a publicly available literary text.

10. Limitations

Several limitations should be considered when interpreting the findings. First, the corpus is based on

[7]http://hdl.handle.net/11234/1-6136

Entity Type	Precision	Recall	F1	Support
animal-W	0.91	0.93	0.92	40
location	0.75	0.91	0.82	21
nationality	1.00	0.33	0.50	3
organization	0.00	0.00	0.00	1
person-W	0.97	0.97	0.97	255
planet	0.76	0.71	0.73	31
plant	0.00	0.00	0.00	0
plant-W	0.89	0.71	0.79	43
product	0.00	0.00	0.00	6
publication	1.00	0.50	0.67	2
time	0.30	0.59	0.40	15
Micro Avg	**0.86**	**0.87**	**0.87**	**417**
Macro Avg	**0.60**	**0.51**	**0.53**	**417**
Weighted Avg	**0.88**	**0.87**	**0.87**	**417**

Table 7: ChatGPT performance on the Persian *Little Prince* corpus under Protocol 2 (full tagset evaluation without label mapping). "-W" denotes weak named entities (WNEs). Note that the *plant* category has a support of 0 in the gold annotations.

a single literary work, which limits stylistic diversity and may not fully represent entity behavior across Persian literature. While *The Little Prince* contains a rich set of narrative referents, it is structurally simpler than many Persian novels and may underrepresent complex constructions such as heavy embedding, poetic metaphor, and culturally grounded naming patterns. Future expansions to additional works are therefore necessary for broader generalization.

Second, entity label distribution is imbalanced. PER and LOC dominate the annotated mentions, whereas ORG occurs rarely. As a result, performance metrics for low-frequency labels are unstable, and confusion matrix interpretation is more reliable for frequent categories. This imbalance also impacts macro-averaged scores, which penalize models heavily for rare labels even when those labels contribute little to overall entity mass.

Finally, evaluation requires label mapping across heterogeneous tagsets. Shared-tag evaluation provides a fair comparison but necessarily collapses distinctions in the original tools (e.g., ParsNER's extended categories such as money/percent/product). Conversely, ChatGPT's richer tagset (including animal, plant, planet) cannot be directly compared under standard NER metrics without multi-label mapping. Therefore, our results should be interpreted as two complementary analyses: (i) standardized shared-tag performance evaluation and (ii) descriptive coverage statistics for the full tagsets.

Acknowledgments

The work described herein was supported by the Charles University, project GAUK No. 394625. The second author was also funded by *LINDAT/CLARIAH-CZ* (Project No. LM2023062) of the Ministry of Education, Youth, and Sports of the Czech Republic. This research was also partially supported by SVV project number 260 821.

11. Bibliographical References

MohammadMahdi Aghajani, AliAkbar Badri, and Hamid Beigy. 2021a. Parstwiner: A corpus for named entity recognition at informal persian. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 131–136.

MohammadMahdi Aghajani, AliAkbar Badri, and Hamid Beigy. 2021b. ParsTwiNER: A corpus for named entity recognition at informal Persian. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 131–136, Online. Association for Computational Linguistics.

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 148–153.

Herley Shaori Al-Ash and Wahyu Catur Wibowo. 2018. Fake news identification characteristics using named entity recognition and phrase detection. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 12–17. IEEE.

Ahmad Amirivojdan. 2025. Shekar: A Python Toolkit for Persian Natural Language Processing. *Journal of Open Source Software*, 10(114):9128.

Majid Asgari-Bidhendi, Behrooz Janfada, OR Roshani Talab, and Behrouz Minaei-

Bidgoli. 2021. Parsner-social: A corpus for named entity recognition in persian social media texts. *Journal of AI and Data Mining*, 9(2):181–192.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83.

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 2009 workshop on the people's web meets NLP: Collaboratively constructed semantic resources (People's Web)*, pages 10–18.

David Bamman, Sejal Popat, and Sheng Shen. 2019. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2138–2144.

Sergey Berezin and Tatiana Batura. 2022. Named entity inclusion in abstractive text summarization. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 158–162.

Oriol Borrega, Mariona Taulé, and M Antònia Martí. 2007. What do we mean when we speak about named entities. In *Proceedings of Corpus Linguistics*, pages 1–27. Citeseer.

Priyankar Bose, Sriram Srinivasan, William C Sleeman IV, Jatinder Palta, Rishabh Kapoor, and Preetam Ghosh. 2021. A survey on recent named entity recognition and relationship extraction techniques on clinical texts. *Applied Sciences*, 11(18):8319.

Tejal Chavan and Seema Patil. 2024. Named entity recognition (ner) for news articles. *Dev.(IJAIRD)*, 2(1):103–112.

Yukun Chen, Thomas A Lasko, Qiaozhu Mei, Joshua C Denny, and Hua Xu. 2015. A study of active learning methods for named entity recognition in clinical text. *Journal of biomedical informatics*, 58:11–18.

Kevin Clark and Christopher D Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Linguistic Data Consortium et al. 2005. Ace (automatic content extraction) english annotation guidelines for entities. *Version*, 5(6):2005–08.

Jesse de Does, Katrien Depuydt, Karina Van Dalen-Oskam, Maarten Marx, et al. 2017. Namescape: named entity recognition from a literary perspective. *CLARIN in the Low Countries*, pages 361–370.

Niels Dekker, Tobias Kuhn, and Marieke van Erp. 2019. Evaluating named entity recognition tools for extracting social networks from novels. *PeerJ Computer Science*, 5:e189.

Ronen Feldman and Benjamin Rosenfeld. 2006. Boosting unsupervised relation extraction by using ner. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 473–481.

Nandita Goyal and Navdeep Singh. 2025. Named entity recognition and relationship extraction for biomedical text: A comprehensive survey, recent advancements, and future research directions. *Neurocomputing*, 618:129171.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Nagaraja Gundluru, Dharmendra Singh Rajput, Kuruva Lakshmanna, Rajesh Kaluri, Mohammad Shorfuzzaman, Mueen Uddin, and Mohammad Arifin Rahman Khan. 2022. Enhancement of detection of diabetic retinopathy using harris hawks optimization with deep learning model. *Computational Intelligence and Neuroscience*, 2022(1):8512469.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Mohammad Ebrahim Khademi and Mohammad Fakhredanesh. 2020. Persian automatic text summarization based on named entity recognition. *Iranian Journal of Science and Technology,*

Transactions of Electrical Engineering, pages 1–12.

Mahboob Alam Khalid, Valentin Jijkoun, and Maarten De Rijke. 2008. The impact of named entity normalization on information retrieval for question answering. In *European Conference on Information Retrieval*, pages 705–710. Springer.

Imaad Zaffar Khan, Amaan Aijaz Sheikh, and Utkarsh Sinha. 2024. Graph neural network and ner-based text summarization. *arXiv preprint arXiv:2402.05126*.

Srinivasa Rao Kundeti, J Vijayananda, Srikanth Mujjiga, and M Kalyan. 2016. Clinical named entity recognition: Challenges and opportunities. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1937–1945. IEEE.

Maolong Li, Qiang Yang, Fuzhen He, Zhixu Li, Pengpeng Zhao, Lei Zhao, and Zhigang Chen. 2019. An unsupervised learning approach for ner based on online encyclopedia. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 329–344. Springer.

Thomas Mandl and Christa Womser-Hacker. 2005. The effect of named entities on effectiveness in cross-language information retrieval evaluation. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1059–1064.

Diego Mollá, Menno Van Zaanen, and Steve Cassidy. 2007. Named entity recognition in question answering of speech data. In *Proceedings of the 2007 Australasian Language Technology Workshop*, pages 57–65. ALTA.

Diego Mollá, Menno Van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian language technology workshop 2006*, pages 51–58.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Yuyang Nie, Yuanhe Tian, Xiang Wan, Yan Song, and Bo Dai. 2020. Named entity recognition for social media texts with semantic augmentation. *arXiv preprint arXiv:2010.15458*.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

Desislava Petkova and W Bruce Croft. 2007. Proximity-based document representation for named entity retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 731–740.

Hanieh Poostchi, Ehsan Zare Borzeshi, Mohammad Abdous, and Massimo Piccardi. 2016. Personer: Persian named-entity recognition. In *COLING 2016-26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2020. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, 54(1):247–272.

Hafedh Ali Shabat and Nazlia Omar. 2015. Named entity recognition in crime news documents using classifiers combination. *Middle-East Journal of Scientific Research*, 23(6):1215–1221.

Mahsa Sadat Shahshahani, Mahdi Mohseni, Azadeh Shakery, and Heshaam Faili. 2018. Peyma: A tagged corpus for persian named entities. *arXiv preprint arXiv:1801.09936*.

Mariana O Silva and Mirella M Moro. 2024. Pportal_ner: An annotated corpus of portuguese literary entities. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12927–12937.

Reza Takhshid, Tara Azin, Razieh Shojaei, and Mohammad Bahrani. 2024. Persian Abstract Meaning Representation: Annotation guidelines and gold standard dataset. In *Proceedings of the 2024 UMR Parsing Workshop*, pages 8–15, Boulder, Colorado. Association for Computational Linguistics.

Hooshvare Team. 2021. Pre-trained ner models for persian. https://github.com/hooshvare/parsner.

Karina van Dalen-Oskam, Jesse de Does, Maarten Marx, Isaac Sijaranamual, Katrien Depuydt, Boukje Verheij, and Valentijn Geirnaert. 2014. Named entity recognition and resolution for literary studies. *Computational Linguistics in the Netherlands Journal*, 4:121–136.

Van-Hai Vu, Quang-Phuoc Nguyen, Kiem-Hieu Nguyen, Joon-Choul Shin, and Cheol-Young Ock.

2020. Korean-vietnamese neural machine translation with named entity recognition and part-of-speech tags. *IEICE TRANSACTIONS on Information and Systems*, 103(4):866–873.

Hanna M Wallach. 2004. Conditional random fields: An introduction.

Shufang Xie, Yingce Xia, Lijun Wu, Yiqing Huang, Yang Fan, and Tao Qin. 2022. End-to-end entity-aware neural machine translation. *Machine Learning*, 111(3):1181–1203.

Jianfei Yu, Ziyan Li, Jieming Wang, and Rui Xia. 2023. Grounded multimodal named entity recognition on social media. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9141–9154.

Jun Zhao. 2009. A survey on named entity recognition, disambiguation and cross-lingual coreference resolution. *Journal of Chinese Information Processing*, 23(2):3–17.

Towards Consistent UMR Annotation of Deverbal Nouns: Evidence from Czech and Latin

Hana Hledíková, Federica Gamba, Marketa Lopatkova, Jan Štěpánek
Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
Malostranské nám. 2/25, 118 00 Prague 1, Czechia
{hana.hledikova, gamba, lopatkova, stepanek}@ufal.mff.cuni.cz

Abstract

Deverbal nouns pose challenges for semantic annotation frameworks that aim to represent event structures consistently across lexical categories. This paper examines problematic phenomena in the annotation of deverbal nouns in Czech and Latin within the Universal Meaning Representation (UMR) framework, addressing both manual graph construction and rule-based automatic conversion from existing resources. Current UMR guidelines lack operational criteria for deciding when a noun should be treated as an eventive concept, particularly in the absence of a PropBank-like lexicon with sufficient nominal coverage. We therefore propose practical annotation principles: deverbal nouns denoting events (such as *učení* 'teaching'), results of events (*řešení* 'solution'), or event participants (*učitel* 'teacher') should be related to underlying event concepts (represented as verbs in their particular senses, i.e., *učit-001* 'to teach', *vyřešit-001* 'to solve', and *učit-001* 'to teach', respectively), while other deverbal nouns should remain unrelated to respective events (such as *učebna* 'teaching room'). To reduce inter-annotator variation, we further suggest systematic strategies for selecting verbal labels, including the use of light-verb constructions, synonymous verbs, and a preference for imperfective verbs in Czech aspectual pairs. For automatic conversion, we outline a rule-based approach that combines multiple lexical resources and frequency-based heuristics to identify corresponding verb senses. Our findings provide guidelines for more consistent UMR annotation across languages.

Keywords: UMR, event nouns, Czech, Latin, Prague Dependency Treebank, Latin Dependency Treebank, automatic conversion

1. Introduction

Uniform Meaning representation (UMR, see esp. van Gysel et al., 2021; Bonn et al., 2024; Bonn et al., 2026)[1] is a framework designed to capture the semantic content of a text in any language. UMR is based on Abstract Meaning Representation (AMR, Banarescu et al., 2013; Wein and Bonn, 2023), originally developed primarily for English with its rich linguistic resources, extending this approach to make it applicable to other languages, particularly those with rich morphology and limited linguistic resources.

In this vein, UMR has been employed to represent Czech and Latin, languages that have relied so far on a sophisticated dependency-based description of deep syntax (Sgall et al., 1986). In addition to preparing a sample of manually annotated data for both languages, two existing corpora—PDT-C (Hajič et al., 2020; Hajič et al., 2024) for Czech and LDT[2] for Latin (Bamman and Crane, 2006; Passarotti, 2014; Gonzalez Saavedra and Passarotti, 2014)—have been leveraged to create a large automatically converted UMR dataset for Czech and Latin (Štěpánek et al., 2025a; Štěpánek et al., 2025b).

During the work on the dataset, *deverbal nouns* and *deverbal adjectives* were identified as problematic phenomena for UMR for two main reasons: first, it is often unclear what the "correct" UMR annotation should be, as multiple analyses may be plausible (Lopatková et al., 2025b); second, even when a preferred annotation can be determined, producing it reliably through automatic conversion remains challenging.

1.1. Need of a PropBank-like lexicon for events

A fundamental requirement of UMR annotation is the distinction between entities and events, the latter of which can be further divided into states and processes. Events are prototypically expressed through predication, i.e., by verbs; however, they may also be realized by event-denoting nominals or adjectives.

In this respect, English UMRs are based on the PropBank lexicon (Palmer et al., 2005; Pradhan et al., 2022),[3] which provides 'frame files' with detailed information on event participants and argument structure; the frames are populated not only with verbs but also with participles, light-verb constructions, event-denoting nouns, and event-denoting adjectives; cf. the 'break.01' frame (∼

[1] https://umr4nlp.github.io/web/index.html

[2] https://itreebank.marginalia.it/

[3] https://github.com/propbank/

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 65–80
11 May 2026.

break, cause to not be whole), which lists break (v.), break (n.), breaking (n.), make_break (l.), broken (j.).

For Czech, two datasets can be used: (i) the PDT-Vallex lexicon (Hajič et al., 2003; Urešová et al., 2021), developed as a resource for valency annotation in PDT-C, and (ii) the Czech part of the SynSemClass ontology (Urešová et al., 2025a; Urešová et al., 2025b). While entries for verbal predicates were successfully (semi)automatically converted to PropBank-like frames (Hajič et al., 2024), the coverage of event-denoting nouns and adjectives is very limited in these resources; therefore, they have not been processed so far.

For Latin, two valency lexicons have been developed over the years: (i) Latin Vallex 1.0 (Passarotti et al., 2016), consisting of approximately 2,500 valency frames, grounded in the tectogrammatical layer of LDT and ITTB (Passarotti, 2019),[4] but not differentiating frames on semantic grounds; and (ii) Latin Vallex 2.0 (Mambrini et al., 2021), which expands coverage to over 45,000 valency frames and links entries to WordNet synsets via the LiLa Knowledge Base (Passarotti et al., 2020), but does not provide cross-references to LDT. Additionally, its usability is limited due to the absence of illustrative examples and the frequent inclusion of highly similar, or even identical, frames. Since these two resources are essentially independent of each other, with little to no overlap beyond some common lemmas, the two valency lexicons have been (partially) combined into Vallex4UMR,[5] suitable for UMR annotation but covering only a subset of LDT. These resources are not restricted to verbs, but also encompass nouns and adjectives.

In this paper, we focus on the challenges that deverbal nouns pose for UMR annotation in Czech and Latin, two languages with rich derivational morphology, but without any PropBank-like valency lexicon that would systematically include nouns along with verbs and could therefore be readily exploited for automatic conversion. Section 2 introduces the issues connected with determining the appropriate UMR representation of deverbal nouns in Czech and Latin, Section 3 focuses on the issues for automatic conversion of existing data formats into UMR and presents some preliminary findings on how additional data resources can be utilized to address them. Section 4 closes the paper with a summary of the findings and brief concluding remarks.

[4]Consequently, each predicate occurring in the tectogrammatical layer is associated with a corresponding valency frame recorded in Vallex 1.0.

[5]https://github.com/fjambe/Vallex4UMR

2. Deverbal nouns with unclear UMR representation

2.1. Events vs. non-events denoted by deverbal nouns

As we have described in Section 1.1, eventive concepts in UMR are not necessarily realized by verbs. Nouns that are derived from verbs and also denote events (i.e., event nouns) are annotated using an eventive concept labeled with the corresponding verb-sense in the reference valency lexicon (cf. the noun *učení* 'teaching' in example 1). Furthermore, derived nouns that denote a participant in an event can also be annotated using an eventive concept in combination with an inverse participant relation, because the eventive meaning of the verb is also clearly referred to by the deverbal noun; cf. agent nouns such as *učitel* 'teacher' in example 2.

Deverbal nouns can also denote other kinds of participants and circumstances; cf. the noun *učivo* 'teaching material; curriculum', which denotes the second argument of *učit-001* 'to teach', or the noun *učebna* 'teaching room', which denotes the location. In such examples, it is less clear whether the UMR representation should still use the eventive concept (although the noun clearly refers to it); rather, a simple entity concept labeled with the noun's lemma seems more appropriate.

In general, the guidelines[6] do not offer testable criteria to identify nouns that should refer to events. There are nouns that are not derived from verbs, but are prototypical participants of events; cf. the primary noun *žák* 'pupil', which is also conceptually associated with an event of teaching (cf. the potentially possible representation in 3), but we would use an entity concept to represent it (cf. 4). Even nouns derived from verbs can be prototypically associated with multiple different events; cf. the noun *jídlo* 'food', which is derived from *jíst* 'to eat', but there is no principled reason not to annotate it with reference to a different event in which it also participates, such as *vařit* 'to cook'. Although the fact that a noun is derived from a verb is a certain indication that may lead to preferring a certain eventive concept, morphology is language-specific and should not, in principle, be the criterion for the UMR representation. In the absence of a reference valency lexicon that would list the nouns that are linked to a particular verbal frame, some testable criteria need to be specified to choose the appropriate representation–either a simple entity concept or a corresponding eventive concept.

(1) *učení* 'teaching'

```
(s1u1 / učit-001 'to teach'
    ...)
```

[6]https://github.com/umr4nlp/umr-guidelines/blob/master/guidelines.md

(2) *učitel* ‘teacher’

```
(s1p1 / person
        :refer-number singular
        :ARG0-of (s1u1 / učit-001
                 'to teach'
                 ...))
```

(3) *žák* ‘pupil’

```
(s1p1 / person
        :refer-number singular
        :ARG1-of (s1u1 / učit-001
                 'to teach'
                 ...))
```

(4) *žák* ‘pupil’

```
(s1z1 / žák 'pupil'
        :refer-number singular)
```

Essentially, we can view deverbal nouns that denote events as differing from verbs simply in the “information packaging” (Croft, 2001, 2022): the same semantic type (a process) is presented using the referential function rather than the predication function, which is reflected in the use of a different part of speech—a noun instead of a verb. In Kuryłowicz (1936), this kind of noun formation is discussed under the term “syntactic derivation”, in contrast to “lexical derivation” which also involves a change in the kind of concept that is denoted by the derived word vs. the base word and is exemplified by the nouns *učitel* ‘teacher’, *učivo* ‘teaching material; curriculum’ or *učebna* ‘teaching room’. In nouns of this second type, we suggest formulating clear criteria for manual annotation to decide when to use an eventive concept, such as limiting the eventive label to cases where the noun denotes an argument of the verb, and not another kind of circumstance that would correspond to a non-argument semantic relation (such as `:place-of` in the case of *učebna* ‘teaching room’). The first type of nouns (i.e., those created by “syntactic derivation”) should clearly be represented as an event, but there is still the issue of identifying the appropriate verb-sense that should be used in the eventive concept label in the absence of a PropBank-like lexicon.

2.2. Choosing the appropriate verb entry

Even if choosing the verb does not seem problematic at first look, there are certain properties of derivation which make the decision not obvious in some nouns. Derivation is traditionally characterized as only partially predictable and partially productive (cf. e.g. Tuggy, 1985). In contrast to inflection, where non-existent forms for a given paradigm cell are rather unexpected, unavailable forms for a given derivational meaning are frequently found in derivation (Stump, 2019). While most nouns with an eventive meaning do have a corresponding verb (see, e.g., *boj* and *pugna* ‘fight’ in Table 1), it is also possible to find nouns for which a corresponding verb does not exist. The reason is often lexical borrowing (cf. *akvizice* ‘acquisition’ in Table 1, where only the noun was borrowed into Czech but the verb was not), but it also concerns native vocabulary (cf. *krok* ‘step’ and *iter* ‘journey’ in Table 1). This poses a problem for the UMR annotation: if the noun is not included in the reference valency lexicon, a corresponding verb-sense has to be identified, and it is not clear what to do for nouns without a corresponding verb available in the language.

Light-verb constructions. In such cases, a light-verb construction can be used instead of the missing corresponding verb, e.g., *provést akvizici* ‘to carry out an acquisition’, *udělat krok* ‘to make a step’. Because the Czech reference valency lexicon PDT-Vallex (Urešová et al., 2021) (cf. Section 1.1) also contains light-verb constructions, it is possible to refer to those entries in the annotation. For the noun *akvizice*, the concept `provést-akvizici-004` can be chosen, corresponding to the PDT-Vallex frame *provést* `v41hrmF` ‘to carry out an action’. Problems are posed by event nouns that do not have any light-verb construction available in PDT-Vallex (e.g., *publicita* ‘publicity’, where a light-verb construction such as *dělat publicitu* ‘to do publicity’ is not available in the lexicon) and, conversely, by event nouns that have multiple possible light-verb constructions available in PDT-Vallex (e.g., *učinit krok* ‘to carry out a step’ and *udělat krok* ‘to make a step’, which are both synonymous light-verb constructions for the noun *krok* ‘step’ and which are both included in PDT-Vallex). Both options are essentially correct and both can be used, the issue is only with introducing unnecessary inter-annotator disagreements into the data in case it is evaluated.

For Latin, none of the available valency lexicons (cf. Section 1.1) contains entries for light-verb constructions. This presents a challenge for event nouns that lack a corresponding single verb (e.g., *iter* ‘journey’, which is conventionally used in the light-verb construction *iter facere* ‘to make a journey’). Unlike in Czech, where light-verb constructions are included in PDT-Vallex and can be referenced in annotation, Latin event nouns of this type cannot be linked to an existing light-verb frame, which complicates the annotation process.

Aspectual pairs. Multiple synonymous light-verb constructions are not the only type of situation where there are several possibilities to choose from when annotating an event noun. A frequently occurring example in Czech is associated with the characteristics of its morphological system, namely the verbal category of grammatical aspect. Many Czech verbs form so-called aspectual pairs, i.e., pairs of corresponding verbs with the same root and

Czech	
action noun	**verb(s)**
boj 'fight'	*bojovat* 'to fight'
adopce 'adoption'	*adoptovat* 'to adopt'
akvizice 'acquisition'	—
krok 'step'	—
prodávání 'selling'	*prodávat* 'sell (imperf.)'
prodej 'sale'	*prodat* 'sell (perf.)', *prodávat* 'sell (imperf.)'
řešení 'solution'	*řešit* 'solve (imperf.)', *vyřešit* 'solve (perf.)'

Latin	
action noun	**verb(s)**
pugna 'fight'	*pugno* 'to fight'
acquisitio 'acquisition'	*acquiro* 'to acquire'
adventus 'arrival'	*advenio* 'to arrive'
iter 'journey'	—
usus 'use'	*utor* 'to use'
venditio 'selling/sale'	*vendo* 'to sell'

Table 1: Pairs of an event noun and its corresponding verb(s) where available.

basic lexical meaning, but differing in grammatical aspect—one is imperfective (imperf.) and one is perfective (perf.); cf. *prodat* 'sell (perf.)' – *prodávat* 'sell (imperf.)' and *řešit* 'solve (imperf.)' – *vyřešit* 'solve (perf.)' in Table 1. Each of the two verbs is treated as a separate lexeme in both traditional dictionaries and in PDT-Vallex.

Some deverbal nouns, specifically those formed by the suffix *-ní/tí*, can explicitly preserve information about the grammatical aspect of the base verb in their form, and it is therefore clear which verb to use for annotating the eventive concept (cf. *prodávání* 'selling' in Table 1). However, nouns formed via other processes, such as conversion (but also some *-ní/tí* nouns, cf. *řešení* 'solution' in Table 1), usually do not preserve aspectual information and both verbs from the aspectual pair could be chosen (cf. *prodej* 'sale' in Table 1). Sometimes the sentential context disambiguates the perfective vs. imperfective reading (cf. examples 5 and 6), but it is often the case that both verbs from the aspectual pair could be chosen in a particular occurrence—grammatical aspect is simply unspecified. In spite of this, only one of the verbs in the aspectual pair has to be chosen, because they are treated as separate lexemes in the reference valency lexicon.

(5) *Prohlásil, že vypracoval* ***řešení****.*
'He announced that he worked out a **solution**.'
```
(s1t1 / thing
    :refer-number singular
    :result-of (s1v1 / vyřešit-001
                'to solve (pf.)'
                ...))
```

(6) *při* ***řešení*** *matematické úlohy* (...)
'when **solving** a mathematical problem' (...)
```
(s1r1 / řešit-001
        'to solve (impf.)'
        ...)
```

Polysemy. The polysemy of derivational processes may also be problematic for manual annotation. Because a single derived noun may have different types of semantic relations to its base verb depending on the particular sense, this means that the annotator has to determine which sense was used in each particular context. This is difficult especially in nouns with abstract meanings; for instance, the aforementioned noun *řešení* 'solution' was shown to have both the eventive meaning (the action of solving as in 6) and the resultative meaning (the result of solving as in 5). However, in some contexts, these two meanings are difficult to tell apart, sometimes to such degree that the word may be considered ambiguous (cf., e.g., the sentence *Ale situace nedovolovala jiné řešení* 'The situation did not allow for a different solution'). A very similar issue arises in Latin with deverbal nouns such as *solutio* 'solution' or *venditio* 'selling/sale'. For instance, *solutio* can denote either the act of solving or the resulting solution. The noun *venditio* likewise exhibits this polysemy, displaying an eventive meaning, as in *Venditio est rei suae in alium translatio* 'A sale (act of selling) is the transfer of one's property to another', and a resultative meaning, as in *Antequam venditio transferatur* 'Before the sale (object sold) is transferred'.

It is especially the eventive and resultative meanings that are typically difficult to distinguish. We suggest annotating resultative nouns with an eventive concept in combination with the `:result-of` relation, as it is very close to the eventive meaning, but we expect that the choice between the eventive and resultative reading is a potential source of inter-annotator variation.

A decision tree that supports annotators' decisions is provided in Appendix A.

3. Deverbal nouns in the automatic conversion

When it comes to the automatic generation of UMR graphs from available corpora—PDT-C for Czech and LDT for Latin (see Section 3.1 for the basic characteristics of the corpora; the conversion procedure is discussed by Štěpánek et al., 2025a; Lopatková et al., 2025b)—the main issue is identifying whether a particular noun occurrence should be represented using an eventive concept or not.

The eventive representation is appropriate for those noun occurrences that have an eventive meaning, for those with a resultative meaning, and for those that denote an argument in an event. As already mentioned, derivational morphology can be a guide in this regard, but as we have seen, most derivational processes are polysemous.

Although neither PDT-C nor LDT provides explicit semantic annotation for nouns, both resources annotate certain nouns' syntactic dependents with participant roles (e.g., Actor, Patient). However, this is not done in a systematic way in PDT-C. In LDT, some nouns are also associated with a Vallex 1.0 valency frame. This can be interpreted as an indication that the noun has an eventive meaning in that particular occurrence.

When the eventive representation is chosen, in the absence of an entry for the noun-sense in the reference valency lexicon, a base verb for the given derived noun must be identified. It is not particularly difficult since derivational resources exist both for Czech and Latin (cf. Section 3.1), making it possible to automatically identify the verb from which a noun was derived in the majority cases. However, it is necessary to identify not only the verb, but also the particular sense of the verb from which the noun was derived (in case the verb is polysemous).

Another step that has to be carried out once the correct eventive concept (i.e., a particular sense of the verb) is identified is to map the original noun's dependents onto the argument roles of the eventive concept. When the noun's dependents are annotated using participant roles, the mapping can be carried out using the same procedure that is applied to verbs and their arguments (Hajič et al., 2024). Example 7 shows the changes that are necessary to convert the noun *debata* 'debate' modified by the possessive pronoun *jejich* 'their' and the prepositional phrase *o systémech* 'about systems' from the tectogrammatical layer in PDT-C to the UMR representation. Similarly, example 8 illustrates for Latin how the noun *studium* 'zeal', together with the possessive modifier *suum* 'his' and the prepositional complement *in rem publicam* 'toward the state', is transformed from its tectogrammatical representation in LDT into the corresponding UMR structure.

In cases where the dependents are not annotated using participant roles in the original data format, the task of identifying the participant roles is more difficult, but morphological features such as preposition, nominal case, or pronoun type can be used to estimate the participant role in certain cases. In the Czech example, the preposition *o* 'about' + the locative case in *o systémech* 'about systems' indicates that this is the second argument (`ARG1`) of the verb *debatovat* 'to debate'. However, many morphological forms are ambiguous and cannot be used to decide the type of argument with certainty.

(7) *jejich debata o systémech*
'their debate about systems'

a. PDT-C (tectogrammatical layer)

```
(debata 'debate'
    :ACT oni 'they'
    :PAT systém 'system'
    ... )
```

b. UMR

```
(s1d1 / debatovat-001
    'to debate'
    :ARG0 (s1p1 / person
        :refer-number plural
        :refer-person 3)
    :ARG1 (s1s1 / systém 'system'
        :refer-number plural)
    ... )
```

(8) *studium suom in rem publicam*
'his zeal for the state'

a. LDT (tectogrammatical layer)

```
(studium 'zeal'
    :ACT is 'he'
    :PAT (res 'thing'
        :RSTR publicus 'public')
    ... )
```

b. UMR

```
(s2s1 / studeo-001
        'to devote oneself to'
    :ARG0 (s2p1 / person
        :refer-number singular
        :refer-person 3)
    :ARG1 (s2r1 / res 'thing'
        :mod (s2p2 / publicus
                     'public')
        :refer-number singular)
    ... )
```

For nouns that denote an argument of the verb or have the resultative meaning, the graph structure must also be substantially changed. Compare, for instance, the annotation of the phrase in 9: while in PDT, the agent noun *učitel* 'teacher' is the parent node of two dependent nodes *můj* 'my' and *dějepis* 'history', the UMR structure is more complex: an abstract concept `person` serves as the parent of a new verbal concept `učit-001` 'to teach', being its `ARG0` (thus `ARG0-of` relation is used); the attributes *můj* 'my' and *dějepis* 'history' are arguments of the verbal concept.

(9) *můj učitel dějepisu*
'my history teacher'

a. PDT-C (tectogrammatical layer)

```
(učitel 'teacher'
    :RSTR můj 'my'
    :RSTR dějepis 'history'
    ... )
```

b. UMR

```
(s1p1 / person
  :refer-number singular
  :ARG0-of (s1u1 / učit-001
                    'to teach'
        :ARG1 (s1d1 / dějepis
                     'history'
            :refer-num. sing.)
        :ARG2 (s1p1 / person
            :refer-num. sing.
            :refer-person 1)
        ...))
```

3.1. Data resources for deverbal nouns in UMR

To address the issues presented in the previous section in the automatic conversion of the Czech and Latin data into the UMR format, several types of data resources can be used.

PDT-C and LDT corpora. The source data used for the conversion are represented by the Prague Dependency Treebank (PDT-C))[7] (Hajič et al., 2020; Hajič et al., 2024) for Czech and the Latin Dependency Treebank (LDT)[8] (Bamman and Crane, 2006; Passarotti, 2014; Gonzalez Saavedra and Passarotti, 2014) for Latin. Both treebanks share the same dependency-based annotation scenario, which is centered on the predicate-argument structure (valency) and other deep syntactic relations. It is further enriched with semantically relevant morphological features (e.g., number and gender for nouns; tense, aspect, and modality for verbs), topic–focus articulation, and coreference annotation. More detailed comparison of the two frameworks, including an overview of automatic conversion, is presented in (Lopatková et al., 2024; Štěpánek et al., 2025a; Lopatková et al., 2025a).

PDT-Vallex lexicon. To deal with derived nouns, it is necessary to combine the source data with additional data resources. The most straightforward information for Czech nouns is provided in PDT-Vallex: for a limited number of deverbal nouns with the suffix *-ní/tí*, explanatory notes identify base verbs (but not the particular sense of the verb). Some deverbal nouns created using other word-formation processes are also included, but not in a systematic way, and their entry does not contain any information about their base verbs.

DeriNet and WFL. Furthermore, both Czech and Latin have fairly large derivational networks available: DeriNet (Olbrich et al., 2025) contains over 1 million Czech lexemes connected via word-formation links, while Word Formation Latin (WFL) (Litta et al., 2018) is a lexicon for Classical and Late Latin covering 69,682 lemmas and modeling word formation through rules represented as directed one-to-many input–output relations between lemmas. However, using these resources directly is not possible, because they do not include any information about the derived words' semantic relations to their base words (except for the identification of diminutives, female counterparts, and iterativity in DeriNet) and no information about valency. Additional steps are therefore required to link a derived noun to a particular verb-sense in the reference valency lexicon.

NomVallex lexicon. For Czech, a resource that provides information on both semantics and valency of derived nouns is the NomVallex lexicon (Kolářová et al., 2024), which contains 730 deverbal or deadjectival nouns, and deverbal, denominal, deadjectival and primary adjectives. Each word's entry comprises one or more senses, and each sense is associated with a valency frame, as well as a semantic category (such as 'action', 'abstract result of an action', 'material', 'object') and a particular verb-sense in VALLEX (Lopatková et al., 2022) from which the particular sense of the noun is derived. For example, the noun *řešení* 'solution' has the following three senses in NomVallex:

1. ∼ *seeking a satisfactory solution; discussing*
   valency: ACT(2,7,poss), PAT(2,poss)
   derived from: *řešit*-001
   semantic category: action

2. ∼ *the finding of a satisfactory solution*
   valency: ACT(2,poss,od+2), PAT(2,poss,že)
   derived from: *řešit*-001
   semantic category: abstract result of an action

3. ∼ *technical or artistic arrangement*
   valency: ACT(2,7,poss,od+2), PAT(2,poss)
   derived from: *řešit*-001
   semantic category: action / abstract result of an action

Nominals in Latin Vallex. For Latin, both Vallex 1.0 and 2.0 (and consequently Vallex4UMR) cover not only verbal predicates but also nominal and adjectival entries. However, the general issues identified for these resources (see Section 1.1) also affect the nominal domain. Most notably, frames in Vallex 1.0 lack semantic grounding, whereas those in Vallex 2.0 are not linked to LDT, except for the subset manually merged in Vallex4UMR. Unlike NomVallex for Czech, nominal entries in Latin Vallexes do not reference the verb-sense from which their particular meaning is

[7] http://hdl.handle.net/11234/1-5813
[8] https://itreebank.marginalia.it/

derived; instead, they are treated as independent lexical entries.

Smaller datasets of particular types of derivatives were also compiled to serve as material for research on Czech word-formation and can be useful in the automatic UMR annotation, namely an agent nouns dataset and a conversion dataset.

Czech agent nouns dataset. A sample of 2,828 agent nouns derived from both verbs, nouns, and adjectives using a number of different suffixes (e.g., the suffix *-tel* as in *žadatel* 'applicant' < *žádat* 'to apply', or the suffix *-ák* as in *divák* 'spectator' < *dívat se* 'to watch') was collected using the DeriNet lexicon combined with additional manual annotation.

Czech conversion dataset. A sample of pairs of suffixless nouns and their verbal counterparts, i.e., pairs of verbs and nouns where either the noun is created from the verb or the verb is created from the noun via conversion (such as *koncert* 'concert' – *koncertovat* 'to give a concert', *poprava* 'execution' – *popravit* 'execute (perf.)' / *popravovat* 'execute (imperf.)') has been compiled for Czech. This dataset is highly diverse—it contains pairs of nouns and verbs without making any decisions about the direction of conversion (deverbal vs. denominal). 50 concordances were extracted for each pair from a corpus of contemporary Czech (Křen et al., 2015) and then the concordances were manually annotated for the semantic relation between the noun and the verb in each particular occurrence, using categories such as 'action', 'result', 'agent', etc. (Ševčíková et al., 2023a; Ševčíková et al., 2023b).

3.2. Using the data resources in the automatic conversion

PDT-Vallex lexicon. To involve the annotation of eventive nouns in automatic conversion, we started by focusing on the most straightforward and systematic category of deverbal nouns: those ending with *-ní/-tí*. In total, we identified 1,690 such nouns in the PDT-C data. Using DeriNet and MorfFlex (Hajič et al., 2020), we were able to process 1,675 of these nouns and identify their base verb lexemes. These verbs are described by 2,248 valency frames (i.e., senses) in the PDT-Vallex lexicon. Almost half of them (1,062 nominal valency frames, 47 %) can be unambiguously mapped onto verbal valency frames using only participant labels; another small part can be mapped based on the morphological form of their participants.

NomVallex lexicon. This lexicon can aid the automatic conversion in cases where the participant roles are not used with the nouns in PDT-C. The process is not straightforward, because although each noun-sense in NomVallex is provided with the base verb's sense and a semantic category, there is no way of directly connecting a particular occurrence of a noun in PDT-C to a particular sense in NomVallex, as the two resources are not interlinked. However, nouns that have only a single sense in NomVallex (and can therefore be considered monosemous) can be annotated fully automatically. Out of the 730 lexemes in NomVallex, there are 91 nouns with a single sense denoting an 'action' and 34 nouns with a single sense denoting an 'abstract result of action'. In the conversion procedure, these can be either assigned an eventive concept labeled with the base verb's sense (for the former case) or such an eventive concept in combination with the `:result-of` participant relation (for the latter case).

Czech agent nouns dataset. The dataset contains lexemes along with their base word, and can therefore be used to identify nouns that should be annotated using their base verb in combination with the `:ARG0-of` inverse relation. Nouns in the dataset that are derived from parts of speech other than verbs can be disregarded, leaving 1,178 lexemes in the dataset.

An issue that needs to be addressed is that due to affix polysemy, some agent nouns can denote both an agent and an instrument; for example, the noun *nosič* 'carrier' can refer both to a person that carries something (a porter) or an instrument used for carrying something. However, this issue is easily solved in Czech because the agent nouns typically have masculine grammatical gender, and masculine nouns express animacy. The category of animacy is part of the morphological annotation in PDT-C, and it is therefore possible to automatically tell apart the non-animate (and therefore instrumental) nouns from the animate (and therefore agentive) nouns.

Once the agent noun and its base verb are identified, the particular sense of the verb that corresponds to the noun has to be found in PDT-Vallex in the next step. Out of the 1,178 agent nouns, 401 have a base verb that only has a single sense in PDT-Vallex, and this sense can therefore be assigned fully automatically. For 235 out of the agent nouns that have a base verb with multiple senses, we were able to manually identify a verb-sense that will very likely correspond to the agent noun in all of its occurrences. For example, the base verb *bruslit* 'to skate' corresponding to the noun *bruslař* 'skater' has two senses in PDT-Vallex (1. to act skillfully in some activity, 2. to skate), but the agent noun is clearly only related to the second sense of the base verb. Taken together, this means that there is

a total of 636 agent nouns that can be automatically assigned a particular verb-sense in the conversion.

Czech conversion dataset. Using this dataset presents some specific challenges. Firstly, because the dataset was compiled without making any decisions about whether the noun was converted from the verb or vice versa, it contains some nouns that are clearly not deverbal, such as *šéf* 'chief, boss' (along with its corresponding denominal verb *šéfovat* 'to be the boss'). Therefore, we only focus on the nouns that are annotated as denoting an 'action' in all corpus concordances in the sample, because although the direction of conversion is still unclear in many cases (cf. e.g. Ševčíková, 2021, for a discussion on directionality in Czech conversion), they should uncontroversially be annotated using the eventive concept in UMR. There are 257 such nouns in the dataset.

In this case, the issue connected with the Czech aspectual system (cf. Section 2.2) is prominent: 161 of these nouns have both an imperfective and a perfective corresponding verb that differs in the thematic suffix (e.g., *vyrobit* 'to produce (perf.)' and *vyrábět* 'to produce (imperf.)' for the noun *výroba* 'production'), and because the thematic suffix is not part of the noun, it is not immediately clear from the noun's form which verb should be chosen in the annotation. As we have mentioned, sometimes this is ambiguous even in a particular sentential context. We were able to manually identify the verb that will likely correspond to the noun in all its occurrences for 96 out of the 161 nouns. Furthermore, there are 90 verbs with only an imperfective verb and 6 nouns with only a perfective verb available. Therefore, it is possible to identify a single verb for 192 nouns. Out of these nouns, we were able to manually identify a particular verb-sense in PDT-Vallex that is likely to correspond to the noun in all its occurrences for 137 nouns.

Further, for these 137 nouns, we also tried to look into a procedure that would automatically label the noun's dependents as arguments of the eventive concept in case the dependents are not annotated with participant labels in PDT-C. In this procedure, we can use information about the morphological realization of arguments that is available in PDT-Vallex for 81 of the verbs; cf., e.g., the mapping for the noun *debata* 'debate':

- genitive, possessive pronoun → `ARG0`
- *o* 'about' + locative, *nad* 'over' + instrumental, *zda*, *zdali*, *jestli* 'if' → `ARG1`
- *s* 'with' + instrumental → `ARG2`

Using this mapping, the example sentence containing *debata* 'debate' given in 7 would be correctly converted into the UMR format even if it was not annotated using participant roles in PDT-C, by applying information about the dependents' forms in the morphological layer. Morphological information is not always fully deterministic, as some forms can be ambiguous as to which argument they express (cf. e.g. Kolářová et al., 2019). This is typical for instance for the genitive case, which can often refer either to the first argument or the second argument; cf. the noun *bojkot* 'boycott' in example 10, where the genitive refers to `ARG0` (the countries are the ones doing the boycotting), vs. in example 11, where the genitive refers to `ARG1` (the goods are what is being boycotted).

(10) *bojkot západních zemí*.`ARG0`
'boycott by the western countries'

(11) *bojkot zboží*.`ARG1`
'boycott of the goods'

Vallex4UMR. For Latin, Vallex4UMR can partially support automatic conversion. A subset of nouns from Vallex 1.0, which also appear in LDT, has been manually mapped to the corresponding entries in Vallex 2.0 and is therefore included in Vallex4UMR. These entries can consequently be converted with relative ease, but they only amount to 111 eventive nouns in LDT. However, beyond this subset, there is no direct way to link a specific occurrence of a noun in LDT to a particular sense in Vallex 2.0, as the two resources are not interlinked. For the remaining entries, the conversion process is further complicated by the fact that noun senses are not associated with their corresponding base verbs and semantic categories, as is the case in NomVallex for Czech. Nevertheless, nouns that are monosemous in Vallex4UMR and denote an event[9] (i.e., 464 out of the 1,857 eventive nominal lexemes in Vallex4UMR) can be annotated fully automatically. These nouns can be assigned the corresponding sense during conversion; however, the distinction between eventive and resultative interpretations cannot be made, as such information is not represented in the resource.

In summary, we have proposed how to use the available data resources for the identification of the appropriate UMR representation of derived nouns based on eventive concepts for varying numbers of nouns (Table 2). The numbers indicate nouns that can be assigned a representation with high confidence. For other nouns, certain heuristics can be applied; for example, assigning the most frequent verb-sense to polysemous nouns derived from polysemous verbs, or using the morphological information of a noun's dependents to infer

[9] In Vallex 2.0, and thus in Vallex4UMR, non-eventive nouns are also defined with senses. We identify eventive nominals by their annotation of participant roles.

correct argument labels. However, it remains unclear whether such heuristics would improve the accuracy of the automatic conversion, and whether they would be useful in creating pre-annotated data for subsequent manual correction, as they might instead introduce additional challenges for annotators. This needs to be tested in the future.

Decision trees that summarize the proposed automatic conversion are in Appendices B and C.

CZECH	
data resource	**number of nouns**
PDT-Vallex	1,062
NomVallex	135
Czech agent nouns dataset	636
Czech conversion dataset	137

LATIN	
data resource	**number of nouns**
Monosemous nouns in Vallex4UMR	464
Manually linked nouns in LDT	111

Table 2: The number of nouns that can be automatically assigned a verb-sense in the automatic conversion, for each language and resource.

4. Conclusion

In this paper, we have identified phenomena that are problematic for the annotation of deverbal nouns in Czech and Latin, both in terms of determining the appropriate annotation that would best conform to the guidelines when creating the UMR graphs manually and in terms of creating the UMR graphs via a rule-based automatic conversion from other existing data formats.

In terms of deciding on the appropriate annotation, we have discussed the issues that have to be solved in the absence of a PropBank-like lexicon that would contain sufficient coverage of nouns. As the guidelines do not offer testable criteria for identifying nouns that should be represented using an eventive concept, we suggest that deverbal nouns that clearly refer to their base verb's meaning and denote an event or an argument in the event should be annotated with an eventive concept (in combination with the appropriate inverse argument relation in the latter case). Additionally, we suggest that deverbal nouns denoting the result of an action, which is semantically very close to the eventive meaning, should be annotated with the `:result-of` relation.

When a verb corresponding to the noun has to be chosen for labeling the eventive concept, such verb may not exist in the given language, or multiple possible verbs may be available even in a specific sentential context. In case of a non-existent verb, a light-verb construction can be searched in the reference valency lexicon. When several synonymous light-verb constructions are available, they are both in principle correct and we suggest that annotators should agree on a simple rule, such as using the one that is listed first, to avoid introducing unnecessary inter-annotator disagreements into the data. Where there is no light-verb construction available in the reference valency lexicon, we suggest that a synonymous verb should be found and used instead (cf. some similar examples where a synonymous word is used instead of a multi-word expression in Bonn et al., 2023). In case there are multiple verbs corresponding to a particular noun occurrence, a systematic decision on which of them to use should be made to avoid unnecessary inter-annotator disagreements again—for the Czech aspectual pairs, we suggest preferring the imperfective verb, as it is sometimes taken to be the unmarked member of the pair (cf. e.g. Komárek et al., 1985, p. 180). The approach fully complies with the UMR ontology—all the solutions that we have proposed use existing concepts, relations and attributes.

As for the automatic conversion, the principal issue is identifying the particular sense of the deverbal noun and the particular verb-sense that corresponds to it. We have shown that there are several resources that can be used in combination with the original PDT-C and LTD data to find the corresponding verb-sense for derived nouns in a rule-based way. This is possible in cases where the noun is identified as monosemous and the base verb is either monosemous or polysemous but the noun is supposed to only relate to one of the senses in all its occurrences. Additional heuristics could be applied to other cases, such as choosing the most frequent sense of the noun and the corresponding verb for the annotation, because some of the resources do also provide information about how comparatively frequent the individual senses are. In the next step, it is necessary to test the procedures suggested for the automatic conversion on actual data and formally evaluate the result.

The procedures we suggest to apply to the Czech and Latin data in the process of manual annotation can be in principle applied to any language, although different languages may present additional problematic phenomena that we have not focused on. Extending the automatic conversion to other languages is not straightforward, as the procedure is dependent on the specific data resources that are available for a given language.

5. Acknowledgements

The work described herein has been supported by the grants *LINDAT/CLARIAH-CZ* (Project No. LM2023062) of the Ministry of Education, Youth, and Sports of the Czech Republic, GAUK No. 104924 of the Charles University, and the Charles University Research Centre program No. 24/SSH/009.

The project has been using data and tools provided by the *LINDAT/CLARIAH-CZ Research Infrastructure* (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

6. Ethic statement

All data used in this study are derived from publicly available language resources; therefore, no ethical approval was required and no ethical guidelines were violated.

7. Bibliographical References

David Bamman and Gregory Crane. 2006. The design and use of a Latin dependency treebank. In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT2006)*, pages 67–78. Citeseer.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E. L. Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italia. ELRA and ICCL.

Julia Bonn, Andrew Cowell, Jan Hajič, Alexis Palmer, Martha Palmer, James Pustejovsky, Haibo Sun, Zdeňka Urešová, Shira Wein, Nianwen Xue, and Jin Zhao. 2023. UMR annotation of multiword expressions. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 99–109, Nancy, France. Association for Computational Linguistics.

William Croft. 2001. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press, Oxford.

William Croft. 2022. *Morphosyntax: Constructions of the World's Languages*. Cambridge University Press, Cambridge.

Berta Gonzalez Saavedra and Marco Carlo Passarotti. 2014. Challenges in enhancing the Index Thomisticus treebank with semantic and pragmatic annotation. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT-13)*, pages 265–270.

Jan Hajič, Eduard Bejček, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague Dependency Treebank - Consolidated 1.0. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, pages 5208–5218, Marseille, France. European Language Resources Association.

Jan Hajič, Eva Fučíková, Markéta Lopatková, and Zdeňka Urešová. 2024. Mapping Czech Verbal Valency to PropBank Argument Labels. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations (DMR 2024)*, pages 88–100, Torino, Italia. ELRA and ICCL.

Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová, and Petr Pajas. 2003. PDT-VALLEX: Creating a large-coverage valency lexicon for treebank annotation. In *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, volume 9 of *Mathematical Modeling in Physics, Engineering and Cognitive Sciences*, pages 57–68, Vaxjo, Sweden. Vaxjo University Press.

Veronika Kolářová, Anna Vernerová, and Jonathan Verner. 2019. Non-systemic valency behavior of Czech deverbal nouns based on the NomVallex lexicon. *Jazykovedný časopis / Journal of Linguistics*, 70(2):424–433.

Miroslav Komárek, Jan Kořenský, Jan Petr, and Jarmila Veselková. 1985. *Mluvnice češtiny 2. Tavrosloví*. Academia, Prague.

Jerzy Kuryłowicz. 1936. Derivation lexicale et derivation syntaxique. *Bulletin de la Société de linguistique*, 32:79–92.

Markéta Lopatková, Eva Fučíková, Federica Gamba, Jan Hajič, Hana Hledíková, Marie Mikulová, Michal Novák, Jan Štěpánek, Daniel Zeman, and Šárka Zikánová. 2025a. UMR 2.0 - Czech: Release Notes. Technical Report TR-2025-74, ÚFAL MFF UK, Prague, Czechia.

Markéta Lopatková, Eva Fučíková, Federica Gamba, Jan Štěpánek, Daniel Zeman, and Šárka Zikánová. 2024. Towards a conversion of the Prague Dependency Treebank data to the Uniform Meaning Representation. In *Proceedings of the 24th Conference Information Technologies – Applications and Theory (ITAT 2024)*, pages 62–76, Košice, Slovakia. Univerzita Pavla Jozefa Šafárika v Košiciach, Košice, Slovakia, CEUR-WS.org.

Markéta Lopatková, Hana Hledíková, Jan Štěpánek, and Daniel Zeman. 2025b. From the Prague Dependency Treebank to the Uniform Meaning Representation: Gold-standard Czech UMR data and partial automatic conversion. In *Proceedings of the 25th Conference Information Technologies – Applications and Theory (ITAT 2025)*, pages 179–190, Košice, Slovakia. CEUR-WS.org.

Francesco Mambrini, Marco Passarotti, Eleonora Litta, and Giovanni Moretti. 2021. Interlinking valency frames and wordnet synsets in the LiLa knowledge base of linguistic resources for Latin. In *Further with Knowledge Graphs*, pages 16–28. IOS Press.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Marco Passarotti. 2014. From Syntax to Semantics. First Steps Towards Tectogrammatical Annotation of Latin. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 100–109, Gothenburg, Sweden. Association for Computational Linguistics.

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. *Digital Classical Philology*, 10:299–320.

Marco Passarotti, Francesco Mambrini, Greta Franzini, Flavio Massimiliano Cecchini, Eleonora Litta, Giovanni Moretti, Paolo Ruffolo, and Rachele Sprugnoli. 2020. Interlinking through lemmas. The lexical collection of the LiLa knowledge base of linguistic resources for Latin. *Studi e Saggi Linguistici*, 58(1):177–212.

Marco Passarotti, Berta González Saavedra, and Christophe Onambele. 2016. Latin Vallex. A Treebank-based Semantic Valency Lexicon for Latin. In *Proceedings LREC 2016*, pages 2599–2606, Portorož, Slovenia. ELRA.

Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O'Gorman, James Gung, Kristin Wright-Bettner, and Martha Palmer. 2022. PropBank comes of age—larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. ACL.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Reidel, Dordrecht.

Jan Štěpánek, Daniel Zeman, Markéta Lopatková, Federica Gamba, and Hana Hledíková. 2025a. Comparing Manual and Automatic UMRs for Czech and Latin. In *Proceedings of the Sixth International Workshop on Designing Meaning Representations (DMR 2025)*, pages 1–12, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gergory Stump. 2019. Some sources of apparent gaps in derivational paradigms. *Morphology*, 29:271–292.

David H. Tuggy. 1985. The inflectional/derivational distinction. *Work Papers of the Summer Institute of Linguistics, University of North Dakota Session:*, 29:209–222.

Zdeňka Urešová, Eva Fučíková, Cristina Fernández-Alcaina, and Jan Hajič. 2025a. Linking an Event-type Ontology to Morphosyntax of the Predicate-Argument Structure. *Dictionaries: Journal of the Dictionary Society of North America*, 46(1):207–227.

Jens van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, James Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, and Rosa Vallejos. 2021. Designing a uniform meaning representation for natural language processing. *KI - Künstliche Intelligenz*, 35(2):343–360.

Shira Wein and Julia Bonn. 2023. Comparing UMR and cross-lingual adaptations of AMR. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations (DMR 2023)*, pages 23–33, Nancy, France. Association for Computational Linguistics.

Magda Ševčíková. 2021. Action nouns vs. nouns as bases for denominal verbs in Czech: A case study on directionality in derivation. *Word Structure*, 14(1):97–128.

Magda Ševčíková, Hana Hledíková, Lukáš Kyjánek, and Anna Staňková. 2023a. Semantics of noun/verb conversion in czech: lessons learned from corpus data annotation. *SKASE Journal of Theoretical Linguistics*, 20(4):74–92.

8. Language Resource References

Julia Bonn and Claire Bonial and Matt Buchholz and Hsiao-Jung Cheng and Alvin Chen and Ching-wen Chen and Andrew Cowell and William Croft and Lukas Denk and Ahmed Elsayed and Eva Fučíková and Federica Gamba and Carlos Gomez and Jan Hajič and Eva Hajičová and Jiří Havelka and Loden Havenmeier and Hana Hledíková and Ath Kilgore and Veronika Kolářová and Lucie Kučová and Kenneth Lai and Bin Li and Jingyi Li and Markéta Lopatková and Marie MacGregor and Marie Mikulová and Jiří Mírovský and Anna Nedoluzhko and Skatje Myers and Michal Novák and Tim O'Gorman and Petr Pajas and Alexis Palmer and Martha Palmer and Jarmila Panevová and Benét Post and James Pustejovsky and Petr Sgall and Jialin Song and Li Song and Magda Ševčíková and Jan Štěpánek and Zdeňka Urešová and Haibo Sun and Yao Sun and Rosa Vallejos Yopán and Jens Van Gysel and Meagan Vigus and Kristin Wright-Bettner and Jiawei Wu and Nianwen Xue and Dan Xing and Keer Xu and Zhixing Xu and Liulu Yue and Daniel Zeman and Jin Zhao and Šárka Zikánová and Zdeněk Žabokrtský. 2026. *Uniform Meaning Representation 2.2*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jan Hajič and Eduard Bejček and Alevtina Bémová and Eva Buráňová and Eva Fučíková and Eva Hajičová and Jiří Havelka and Jaroslava Hlaváčová and Petr Homola and Pavel Ircing and Jiří Kárník and Václava Kettnerová and Natalia Klyueva and Veronika Kolářová and Lucie Kučová and Markéta Lopatková and David Mareček and Marie Mikulová and Jiří Mírovský and Anna Nedoluzhko and Michal Novák and Petr Pajas and Jarmila Panevová and Nino Peterek and Lucie Poláková and Martin Popel and Jan Popelka and Jan Romportl and Magdaléna Rysová and Jiří Semecký and Petr Sgall and Johanka Spoustová and Milan Straka and Pavel Straňák and Pavlína Synková and Magda Ševčíková and Jana Šindlerová and Jan Štěpánek and Barbora Štěpánková and Josef Toman and Zdeňka Urešová and Barbora Vidová Hladká and Daniel Zeman and Šárka Zikánová and Zdeněk Žabokrtský. 2024. *Prague Dependency Treebank - Consolidated 2.0 (PDT-C 2.0)*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jan Hajič and Jaroslava Hlaváčová and Marie Mikulová and Milan Straka and Barbora Štěpánková. 2020. *MorfFlex CZ 2.0*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Veronika Kolářová and Václava Kettnerová and Jana Klímová and Jiří Mírovský and Anna Vernerová. 2024. *NomVallex 2.5*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Michal Křen and Václav Cvrček and Tomáš Čapka and Anna Čermáková and Milena Hnátková and Lucie Chlumská and Tomáš Jelínek and Dominika Kováříková and Vladimír Petkevič and Pavel Procházka and Hana Skoumalová and Michal Škrabal and Petr Truneček and Pavel Vondřička and Adrian Jan Zasina. 2015. *SYN2015: A Representative Corpus of Written Czech*. Prague, Institute of the Czech National Corpus, Faculty of Arts, Charles University; http://www.korpus.cz.

Litta, Eleonora and Passarotti, Marco and Culy, Chris. 2018. *Morphology Beyond Inflection. Building a Word Formation-Based Lexicon for Latin*. Cambridge Scholars Publishing, Newcastle upon Tyne.

Markéta Lopatková and Václava Kettnerová and Jiří Mírovský and Anna Vernerová and Eduard Bejček and Zdeněk Žabokrtský. 2022. *VALLEX 4.5*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Michal Olbrich and Viktória Brezinová and Šárka Dohnalová and Vojtěch John and Lukáš Kyjánek and Aleš Papáček and Emil Svoboda and Magda Ševčíková and Jonáš Vidra and Zdeněk Žabokrtský. 2025. *DeriNet 2.3*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Ševčíková, Magda and Kyjánek, Lukáš and Hledíková, Hana and Staňková, Anna. 2023b. *Semantic annotation of noun/verb conversion in Czech*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Jan Štěpánek and Markéta Lopatková and Daniel Zeman and Federica Gamba and Hana Hledíková and Eva Fučíková and Michal Novák and Šárka Zikánová and Eva Hajičová and Jiří Havelka and Veronika Kolářová and Lucie Kučová and Marie Mikulová and Jiří Mírovský and Anna Nedoluzhko and Petr Pajas and Jarmila Panevová and Petr Sgall and Magda Ševčíková and Zdeňka Urešová and Zdeněk Žabokrtský and Jan Hajič. 2025b. *Uniform Meaning Representation 2.1 (Czech and Latin)*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zdeňka Urešová and Eva Fučíková and Jan Hajič and Veronika Kolářová and Cristina Fernández Alcaina and Peter Bourgonje and Eva Hajičová and Georg Rehm and Kateřina Rysová and Karolina Zaczynska. 2025b. *SynSemClass 5.5*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zdeňka Urešová and Alevtina Bémová and Eva Fučíková and Jan Hajič and Veronika Kolářová and Marie Mikulová and Petr Pajas and Jarmila Panevová and Jan Štěpánek. 2021. *PDT-Vallex: Czech Valency lexicon linked to treebanks 4.0 (PDT-Vallex 4.0)*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Appendix A: Decision Tree for Manual Annotation (Czech and Latin)

For each noun decide whether it denotes / relates to an event!

Does the **noun itself denote** an event?

YES: Represent it as an eventive concept. (Add argument relations, see below!)
e.g., *běhání* 'running', *příchod* 'arrival', *akvizice* 'acquiring'

- Does **a single corresponding verb** exist in the valency lexicon?
  - YES: Use the verb's lemma + particular sense in the valency lexicon.
    e.g., *běhání* --> běhat-002
  - NO: Do **multiple corresponding verbs** exist in the valency lexicon?
    - YES: Is it an aspectual pair?
      - YES: Use the imperfective verb.
        e.g., *příchod* --> přicházet-008
      - NO: Use the one listed first in the valency lexicon.
    - NO: (= no corresponding verb)
      - Does **a single LVC exist** in the v. lexicon with the given noun?
        - YES: Use the light-verb's lemma + the noun's word-form.
          e.g., *akvizice* 'acquiring' --> provést-akvizici-004
        - NO: Do **multiple LVCs** exist?
          - YES: Use the one listed first
          - NO: (= no LVC exists)
            Use a synonymous verb.
            e.g., *publicita* 'publicity' --> propagovat-001

NO: (= the noun does not denote event)

- Does the noun denote **an argument** of an event?
  - YES: Represent it as an abstract concept
    with the event attached using the given :ARGx-of relation.
    e.g., *učitel* 'teacher' --> (p / person :ARG0-of (u / učit-001 'teach'))
    e.g., *nabídka* '(an) offer' --> (t / thing :ARG1-of (n / nabídnout-001 '(to) offer')
  - NO: Does the noun denote **an abstract result** of an event?
    - YES: Represent it as an abstract concept (typically thing)
      with the event attached using the :result-of relation.
      e.g., *informace* 'information' (t / thing :result-of (i / informovat-001 'inform'))
      e.g., *plán* '(a) plan' (t / thing :result-of (p / plánovat-001 '(to) plan'))
    - NO: Represent as entity concept (labeled with noun's lemma)

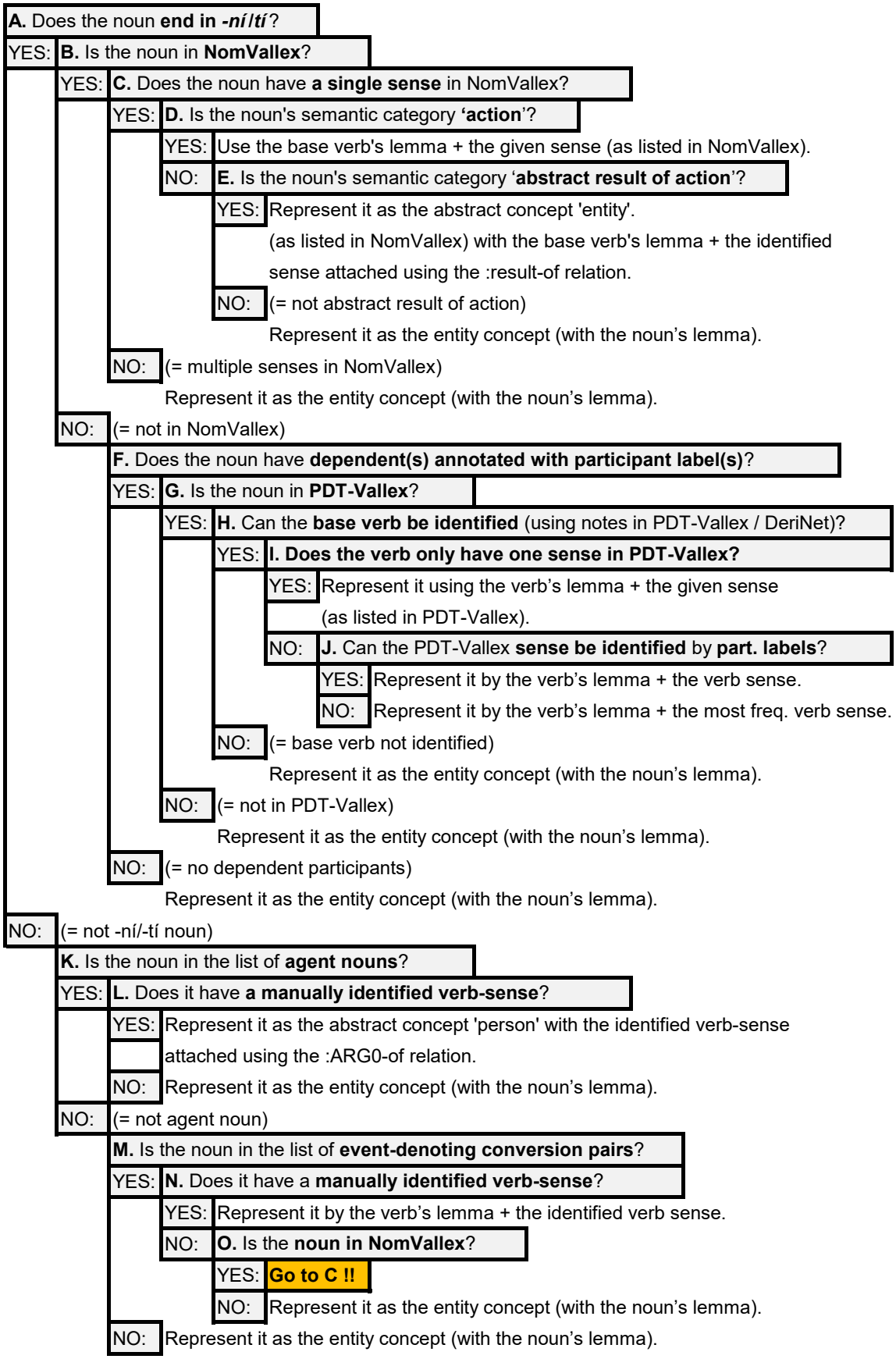
If the eventive concept is chosen: add argument relations!

- Does the dependent correspond to **an argument of the eventive** concept?
  - YES: Represent the dependent's concept as an ARG relation of the eventive concept.
  - NO: Represent the dependent's concept as a non-argument role relation
    of the eventive concept.

Add any other arguments listed in the valency lexicon as obligatory as (unspecified) ARG relations!

Appendix B: Decision Tree for Automatic Conversion (Czech)

For each noun decide how to represent it!



Appendix C: Decision Tree for Automatic Conversion (Latin)

For each noun decide how to represent it!

A. Is the noun mapped onto corresponding **entry in Vallex 2.0**?	
YES:	Represent it by the noun's lemma + the given noun sense.
NO:	**B.** Is it a **monosemous eventive noun** in **Vallex4UMR**?
	YES: Represent it by the noun's lemma + the given noun sense.
	NO: Represent it as the entity concept (with the noun's lemma).

If the eventive concept is chosen: add argument relations!

A. Does the noun has dependents annotated with participant labels?	
YES:	Use the same mapping algorithm that is used for verbs.
NO:	**B.** Is there an **unambiguous mapping** based on morphological form available?
	YES: Use morphological layer to map forms onto arguments.
	NO: Leave arguments unmapped.

SAVI: Web-based Multilayered Semantic Annotation Validation Interface

Sashank Tatavolu[*], Soma Paul[*], Pratibha Rani[*], Sukhada Sukhada[**]
[*]IIIT Hyderabad, [**]IIT (BHU), Varanasi
sashank.tatavolu@research.iiit.ac.in, soma@iiit.ac.in, ranipratibha@gmail.com,
sukhada.hss@iitbhu.ac.in

Abstract

This paper presents SAVI, a web-based interface for multilayer semantic annotation validation of Universal Semantic Representation (USR). USR encodes meaning across interdependent lexical, constructional, relational, discourse, and co-reference layers, making validation challenging using conventional annotation tools. SAVI addresses this limitation through structured tab-based layer separation, constraint-aware editing mechanisms, and role-based review workflows. The system integrates a multilingual concept dictionary to ensure sense-level consistency, along with a Hindi text-generation module and dependency-based visualization to support interpretation and correction. SAVI is implemented using a Flask backend, Flutter frontend, and PostgreSQL for structured data management. Evaluation results demonstrate effective governance of concept proposals and improved efficiency in multilayer USR correction, positioning SAVI as a structured validation framework for scalable semantic corpus development.

Keywords: NLP, Annotation Tool, Semantic Representation, Annotation, Validation, Universal Semantic Representation, Indian Grammatical Tradition

1. Introduction

Semantically rich representations are essential for advanced NLP applications such as inference-based question answering, summarization, and knowledge-driven systems. However, validating semantic annotations can be cognitively demanding when representations span multiple interdependent linguistic layers.

This paper presents **SAVI**, a web-based interface designed for validating **Universal Semantic Representation (USR)**. USR encodes meaning across five interconnected layers: lexico-conceptual, constructional, relational, discourse, and co-reference. Because these layers interact with each other, errors in one layer can propagate across others, making annotation validation structurally complex.

SAVI addresses this challenge through a structured validation framework that separates annotation layers into dedicated tabs while enforcing backend constraints to maintain cross-layer consistency. The system also integrates a multilingual concept dictionary for sense-level governance, along with visualization modules and Hindi text generation to support semantic interpretation during validation.

Unlike existing annotation tools that primarily focus on single-layer or span-based annotation schemes, SAVI is designed specifically for synchronized multilayer semantic validation with controlled concept management and role-based review workflows.

We evaluate SAVI through dictionary governance statistics, validation efficiency analysis, and usability studies conducted with annotators. The results demonstrate improved validation efficiency and consistent multilayer annotation control in practical annotation settings.

The remainder of the paper is organized as follows. Section 2 introduces USR. Section 3 reviews related annotation tools. Section 4 describes the architecture and features of SAVI. Section 5 presents evaluation results. Section 6 concludes, followed by limitations in Section 7.

2. Overview of USR

Universal Semantic Representation (**USR**) has been designed and developed in the last five years by a research team to apply the insights of Pāṇini and the Indian Grammatical Tradition (**IGT**) (Sukhada and Paul, 2023) to modern Indian language technology (Paul et al., 2025; Sukhada et al., 2023). USR is a multi-layered semantic representation system at the document level.

Unlike other semantic representations that completely abstract away from syntactic representations and preserves semantic relations alone (Copestake et al., 2005; Van Gysel et al., 2021; Abzianidze et al., 2017), USR uniquely captures the speaker's intent, how he/she wants to express a situation through what is called **syntactico-semantic** (kāraka in IGT (Sukhada and Paul, 2023; Garg et al., 2023)) relations. For example, the concept '*train_1*' in the sentence given in Figure 1 is in reality an argument of the event '*move_1*' and is so represented in all existing semantic representations. However, the sentence conveys that the actor (*Ram*) boarded a *train* that was already in motion at that particular time. Therefore, '*move_1*' is represented as a present-participial modifier of the concept '*train_1*' in the USR, as shown in Figure 1. The benefit of

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 81–90
11 May 2026.

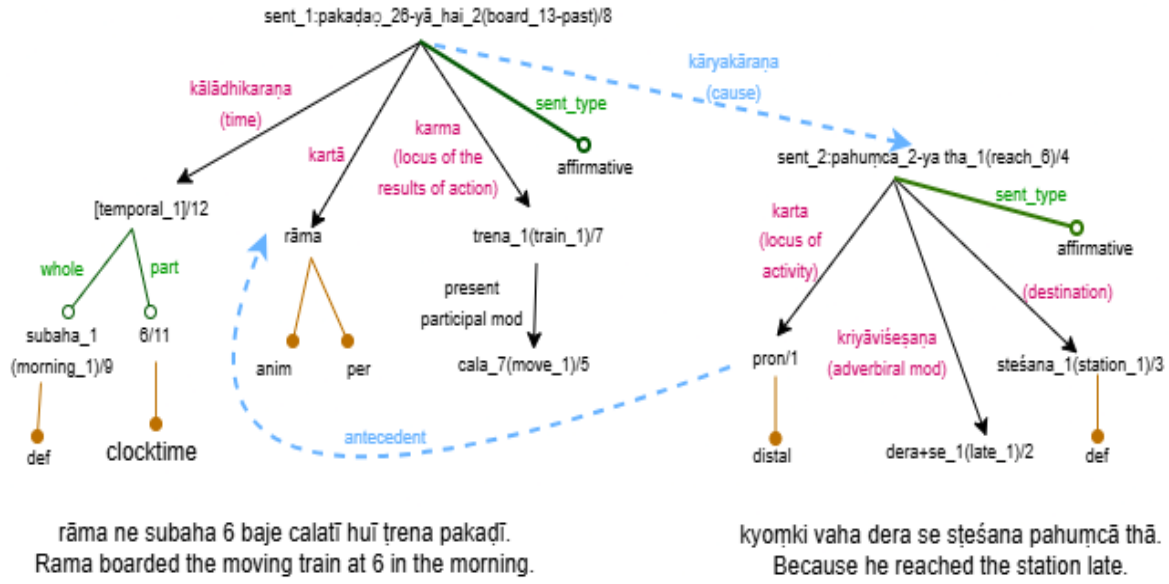


Figure 1: Universal Semantic Representation of a text snippet.

syntactico-semantic relations is that they capture how a speaker intends to delineate a situation. This representation is thus a quite suitable input to the Natural Language Generation system.

Figure 1 is an example USR for a short text snippet. The nodes in the graph represent unique concepts (*morning_1, move_1*) or Complex Concept (CC) labels (*[temporal_1]*). CCs specify complex expressions that have an internal structure. Each member of the CC is given a label, which is specified on the edge. In Figure 1, *[temporal]* has two components: *morning_1* and *6*, which are whole and part in a **part-whole** relationship. Different edge types signify different layers of information. For example, arrows (→) denote **dependency relations**, dotted arrows represent **discourse connective relations**, dotted lines specify **pronominal co-reference**, lines with square denote **CC components**, lines with circle denote **semantic categorical** information of a concept such as person, place, season, day-of-week, week-of-month, month-of-year, male/female and lines with diamond denote **pragmatic information** such as definiteness, respect, proximal/distal, emphasis, exclusiveness/inclusiveness, certainty - meaning that can be expressed by discourse particles.

3. Related Work

In this section, we review some of the annotation tools that have been developed to assist linguists and researchers in annotating text data. The latest web-based tool is EEVEE (Sorensen et al., 2024), which supports Named Entity Recognition (NER), POS tagging, and intent classification tasks.

UD Annotatrix (Tyers et al., 2018) is a language-independent browser-only client-side tool designed to edit dependency trees according to the guidelines of the Universal Dependencies (UD) project. UCCAApp (Abend et al., 2017) is a web-based tool for syntactic and semantic phrase-based annotation, particularly for the UCCA (Universal Conceptual Cognitive Annotation) framework.

BRAT Rapid Annotation Tool (BRAT) (Stenetorp et al., 2012) is a widely used web-based annotation tool that allows users to annotate textual data with entity relationships and dependencies. WebAnno (Yimam et al., 2013) provides annotation support for various linguistic layers, including syntax, co-reference, and discourse relations.

Prodigy (Montani and Honnibal, 2018) is an AI-assisted annotation tool that incorporates active learning to assist annotators.

UMR-Writer (Zhao et al., 2021) is a web-based application used to annotate Uniform Meaning Representation (UMR). UMR is a graph-based, cross-linguistically applicable semantic representation designed to support interpretable natural language applications that require deep semantic analysis.

TrEd (Mirovsky et al., 2010) is a customizable and programmable graphical editor and viewer of tree-like structures such as dependency trees. It is used as the primary annotation tool for syntactic and tectogrammatical annotations in the Prague Dependency Treebank (PDT).

While existing tools provide robust support for syntactic or frame-based annotation, they do not natively support multilayer semantic validation with enforced cross-layer constraints and sense-level dictionary governance. SAVI addresses this gap by integrating structured tab-based validation with

Feature	BRAT	INCEpTION	WebAnno	TrEd (PDT)	UMR-Writer	SAVI
Primary Focus	Entity & relation annotation	Multi-layer annotation + ML support	Multi-layer annotation	Syntax + semantic valency	Deep semantic representation	**Multilayer semantic validation (USR)**
Representation Type	Span + relation-based	Multi-layer structured	Multi-layer structured	Tree-based	Graph (sentence + document)	**Layered tab-based views**
Annotation Layers	Limited	Syntax, discourse, coref	Syntax, discourse, coref	Morphology, analytical, tectogrammatical	Concepts, relations, modality, co-ref	**Lexical, Constructional, Relational, Discourse, Co-ref**
Tool Interface	Web	Web	Web	Desktop	Web	**Web-based modular UI**
Multilingual Support	Yes	Yes	Yes	Czech-centric	Multilingual	**Multilingual (multi-script support)**
Co-reference Support	Limited	Yes	Yes	Yes	Yes	**Yes**
Language Generation Support	No	No	No	No	No	**Yes (Hindi generation)**
Constraint Enforcement	No	Partial	Partial	Limited	No	**Strong (tab-based constraints)**
Document-level Annotation	Limited	Yes	Yes	Limited	Yes	**Yes (speaker-level view)**
Unique Strength	Simplicity	ML-assisted annotation	Flexible annotation layers	Deep syntactic layering	Cross-lingual semantics	**Structured multi-layer validation + dictionary control**

Table 1: Comparison of **SAVI** with existing annotation tools and frameworks.

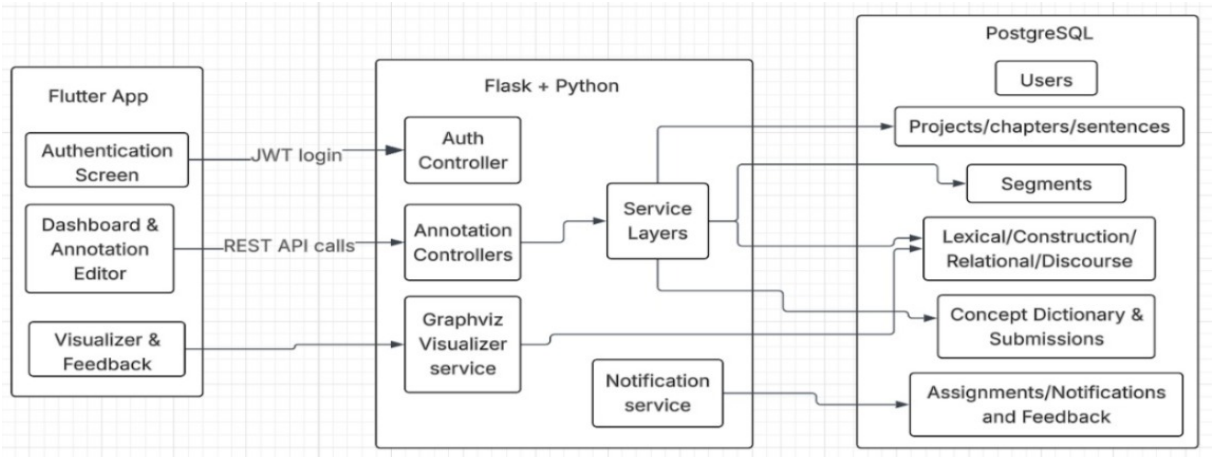


Figure 2: High-level system architecture of **SAVI** showing Backend (Flask), Database (PostgreSQL), and Frontend (Flutter) interaction.

role-based review workflows and concept-level consistency control.

USR is a multi-layer semantic representation that encodes lexical, relational, constructional, discourse, and co-reference information. Its interdependent structure makes annotation difficult using off-the-shelf tools, which primarily support single-layer or span-based schemes without cross-layer constraint enforcement or dictionary-governed validation. SAVI addresses this limitation by separating USR components into structured validation tabs, integrating sentence- and discourse-level visualization, and enforcing role-based consistency control.

Visualization tools such as Stanford NLP Dependency Visualizer tool (Manning et al., 2014), UDPipe (Straka et al., 2016), and spaCy (AI, 2020) present sentence-level graph structure. The SAVI USR-visualizer module provides graphical representations at both sentence and discourse levels using Graphviz (Ellson et al., 1991) and its component Digraph; thus making the tool a competent candidate for document-level semantic annotation. Table 1 presents the comparison of the features of **SAVI** with some of the existing annotation tools.

4. Our Proposed System Details

SAVI is designed for collaborative, distributed linguistic annotation and validation. This section presents the features of our tool, along with its architecture and implementation details.

4.1. Architecture and Implementation Details

SAVI follows a frontend–backend architecture (Figure 2) in which the frontend is implemented using Flutter (Google, 2017) and the backend is developed in Python (Python Software Foundation, 1991) using Flask (Ronacher and the Pallets team, 2010) framework, with modular blueprints for the projects, chapters, annotations, and dictionary management systems (explained in Section 4.3).

The PostgreSQL (PostgreSQL Global Development Group, 1996) database is used for persistent storage, accessed through the SQLAlchemy ORM (Bayer and contributors, 2006), ensuring ACID compliance and indexed lookups for annotation-heavy queries, and Gunicorn (Chesneau and Developers, 2025) for optimized performance. Graphviz is used to generate dependency trees, discourse and co-reference graphs, which are served to the frontend in SVG format.

SAVI provides a role-based user interface with dedicated dashboards for the **Admins**, **Annotators**, **Reviewers**, and **Dictionary Validators** roles.

4.2. User Roles

Our tool supports the following user roles with distinct responsibilities:

- **Admin**: This user role is responsible for uploading input data, assigning annotation tabs (explained in Section 4.3.2) to the validators, sending assignment notifications, and managing user roles.
- **Annotator**: This user role logs into the Flutter app, accesses the assigned segments, and annotates data tab by tab. Each tab enforces constraints (e.g., a head must be chosen in *Relational* **tab**, spans must be consistent in *Construction*). They can also submit new concept entries to the concept dictionary. Once satisfied, they finalize their work, which locks the tab until review.
- **Reviewer**: This user role receives annotated USR data for verification. They can either approve the finalized annotations or return them to the annotators with row-specific feedback. Notifications ensure that annotators are aware of required corrections. This iterative loop continues until the data gets final approval.
- **Dictionary Validator**: This user role reviews proposed new concepts, verifies linguistic accuracy, and approves/rejects them. Approved entries are added to the Concept Dictionary and linked back to the annotation layer.

4.3. Tool Features

This section presents the features available in our tool to help users have a smooth experience.

4.3.1. Chapter and Segment Tabs

The Chapter tab displays the running text for which semantic annotation is performed. The Segment tab displays segments and their segment IDs. Complex sentences are split into segments, with each segment having one finite verb. This tab allows the reviewer to edit segments, split a sentence into segments, if needed, and delete a wrong segment.

4.3.2. Semantic Annotation Tabs

The annotation editor is organized into five tab-based modules, each corresponding to a distinct USR layer:

- **Lexico-Conceptual Tab**(L): Displays concepts with unique ID, their semantic category, and also morpho-semantic and some pragmatic features such as definiteness, respect, honorificity, inclusiveness, exclusiveness, and emphasis associated with the concept under Speaker's View as shown in Figure 3.
- **Construction Tab**(C): Supports annotation of the components of Complex Concepts with enforced span consistency to maintain alignment with token boundaries as shown in Table 1 of Figure 4.
- **Relational Tab**(R): Enables dependency annotation between head and child concepts. Structural constraints enforce a single-head requirement, ensuring valid dependency structures. As shown in Table 2 of Figure 4, the dependency-annotation tab does not display the components of the Complex Concepts, thereby reducing annotators' cognitive load during validation. However, the information is still available in Table 1 of the same interface for annotators to consult if needed.
- **Discourse Connective Tab**(D): Captures discourse-level relations across segments (e.g., contrast, cause, elaboration) using predefined tagsets.
- **Co-reference Tab**(Coref): Supports annotation of pronoun and antecedent co-reference chains across segments via controlled selection mechanisms.

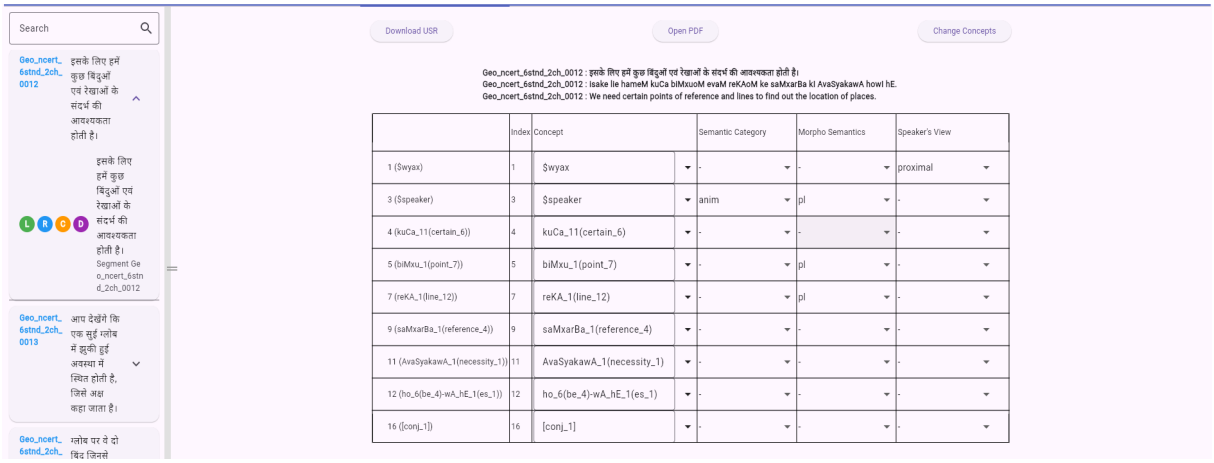


Figure 3: Lexico-conceptual tab of **SAVI**.

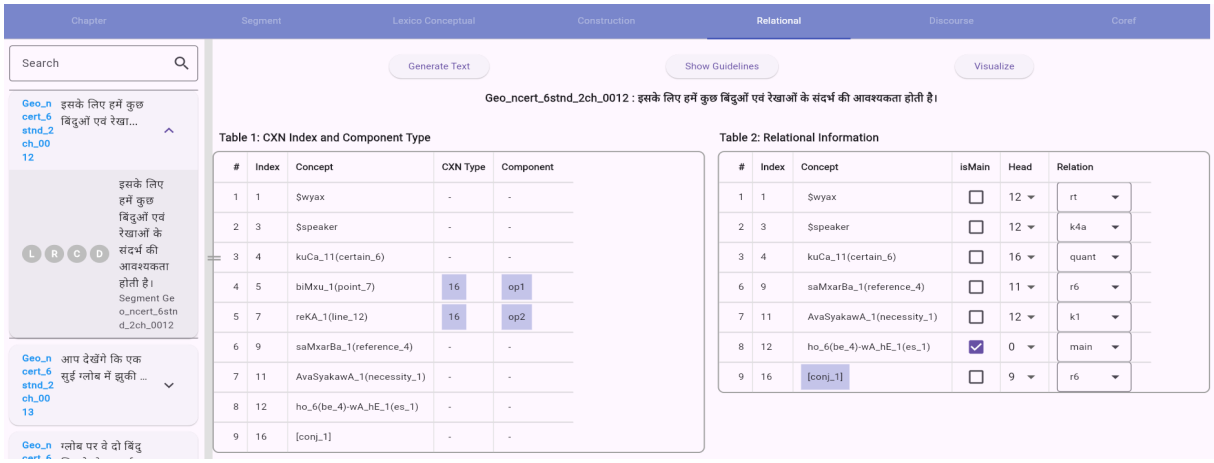


Figure 4: Relational tab of **SAVI**.

No tags are manually entered in these tabs. Instead, they are retrieved dynamically from the backend and displayed as drop-down menus, standardizing input across annotators and preventing invalid entries

4.3.3. Integrated Concept Dictionary

To ensure the postulation of unambiguous concepts and the cross-lingual alignment of concepts, SAVI integrates a **Concept Dictionary** in the Lexico-Conceptual Tab that stores sense representations of a concept across different languages. Each concept entry is assigned a unique `Concept_ID`, which is mapped to the sense representation across multiple languages. Currently, the dictionary includes entries for Hindi, English, Tamil, and Sanskrit. Operating at the sense level mitigates the ambiguity arising from homonymy and polysemy while enforcing stable cross-lingual equivalence.

Annotators select the correct Concept ID for a selected concept in the lexico-conceptual tab, and if no suitable ID exists, they can create a multilingual entry with an appropriate ID and submit it. After submission, the entry is stored with a *pending* flag.

Concept	Hindi	English	Tamil	Sanskrit
read_1	*paḍha_1*	*read_1*	*paṭi_1*	*paṭh_1*
study_1	*paḍha_2*	*study_1*	*paṭi_2*	*adhī_1*

Table 2: Illustrative sense-level multilingual mappings.

4.3.4. Review System

The Review System supports: (1) Concept review and (2) Validation of USR layers through various tabs.

For Concept review, the Dictionary reviewer can review the proposal with *pending* flag and *accept* or *reject* it. The reviewer can also edit the entry. This controlled mechanism prevents duplication and regulates the population of entries in the concept dictionary.

Notifications and email alerts are automatically

sent to annotators via SMTP (Klensin, 2008) when reviews are added. The system enforces a feedback loop that prevents annotators from finalizing unrelated segments until reviewer comments are resolved, ensuring data quality.

4.3.5. Visualization and Search Bar

SAVI provides a visualization module for USR data implemented using Graphviz. The visualizer helps annotators, reviewers, and validators interpret the hierarchical and relational structures encoded in annotations. It displays sentence-level dependency and construction structures as well as discourse-level links across segments. Examples of the visualization output are shown in Figure 5.

The interface also includes usability features to support efficient navigation during validation. A search bar allows users to locate annotation tags from predefined drop-down lists, and a quick-jump option enables direct navigation to a specific `segment_id`. In the segment sidebar, color coding indicates the completion status of annotation layers. Tabs corresponding to lexical (L), constructional (C), relational (R), and discourse (D) layers are highlighted with distinct colors when finalized, while unfinished layers appear in gray. This visual feedback enables annotators to quickly identify segments that require further validation.

4.4. System Workflows

The workflow shown in Figure 6 begins with the Admin uploading a chapter, its sentences, and the corresponding segments. Once uploaded, the system automatically indexes chapters, sentences, and segments, providing structured input for further processing.

Once uploaded, the Admin assigns segments to the validators for verification. Once verified, the segments are passed to the USR-Builder, which automatically creates multilayered USRs by applying a set of heuristics to the outputs of various NLP tools, including a Dependency Parser, a Morph Analyzer, a Named Entity Recognizer (NER), Discourse Connective markers, and Concept Identifiers.

This USR output is then parsed and stored in four database tables: lexico-conceptual (L), relational (R), construction (C), and Discourse (D). The information from these tables is displayed in separate tabs in the interface for manual validation and subsequent review.

Assignments for validation and review are role-specific. Annotators make corrections at L/R/C/D/Coref tabs (refer to Section 4.3.2) and finalize their work, and reviewers approve or return it with feedback. Role-based constraints ensure that annotators can only edit the segments and annotation types assigned to them.

Mode	Median (min)
SAVI-Assisted	2.84
Without Interface	3.73

Table 3: Median validation time per segment.

5. Evaluation, Results and Discussion

This section presents the statistics of the quantitative validation of the Concept Dictionary, the efficiency analysis, and the usability findings of SAVI in real annotation settings.

5.1. Validation Efficiency Analysis

To assess workflow efficiency, we compared median validation time per segment under two conditions: (i) SAVI-assisted validation using SAVI's structured multilayer interface and (ii) validation performed without structured interface support.

The experiment involved 10 annotators, each assigned 15 segments, resulting in 150 validated segments per condition. The same set of segments was used in both conditions to ensure comparability and to control for content complexity.

As shown in Table 3, SAVI-assisted validation reduces median correction time from 3.73 to 2.84 minutes per segment, corresponding to a **23.9% reduction in validation effort**. This improvement was observed consistently across annotators, suggesting that structured tab separation, constraint-aware editing, and controlled tag enforcement reduce annotator overhead related to formatting, structural alignment, and cross-layer consistency checking.

Even modest per-segment savings scale significantly for larger corpora, making the interface suitable for large-scale semantic corpus construction. Although the study is observational and limited in scale, the consistent improvement across annotators indicates that the efficiency gains are systematic and attributable to interface design.

5.2. Usability and Performance Evaluation

System usability was evaluated using the **System Usability Scale (SUS)** (Brooke, 1995, 2013), a widely used questionnaire consisting of ten statements rated on a five-point Likert scale ranging from "Strongly Disagree" to "Strongly Agree," producing an overall usability score between 0 and 100.

Five annotators participated in the evaluation. Three had less than three months of experience with the tool, while two had more than one year of experience. SAVI obtained an **average SUS score of 66.5**, indicating acceptable usability. The dis-

Current Segment: Geo_ncert_7stnd_3ch_0002b: *"Ora ise jala ke AXe Bare blkara meM raKa xIjie."* Connected to Geo_ncert_7stnd_3ch_0002a: *"eka raMgIna kAgaja kI CotI-sI goll lIjie."*

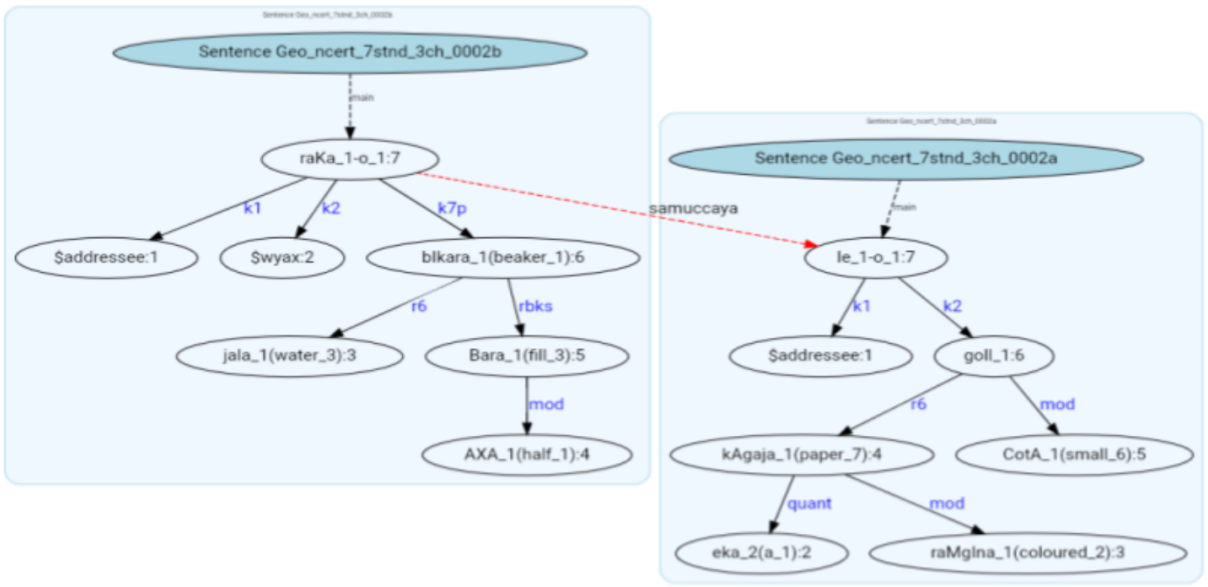


Figure 5: Sample of Visualization of Discourse relation provided by **SAVI**.

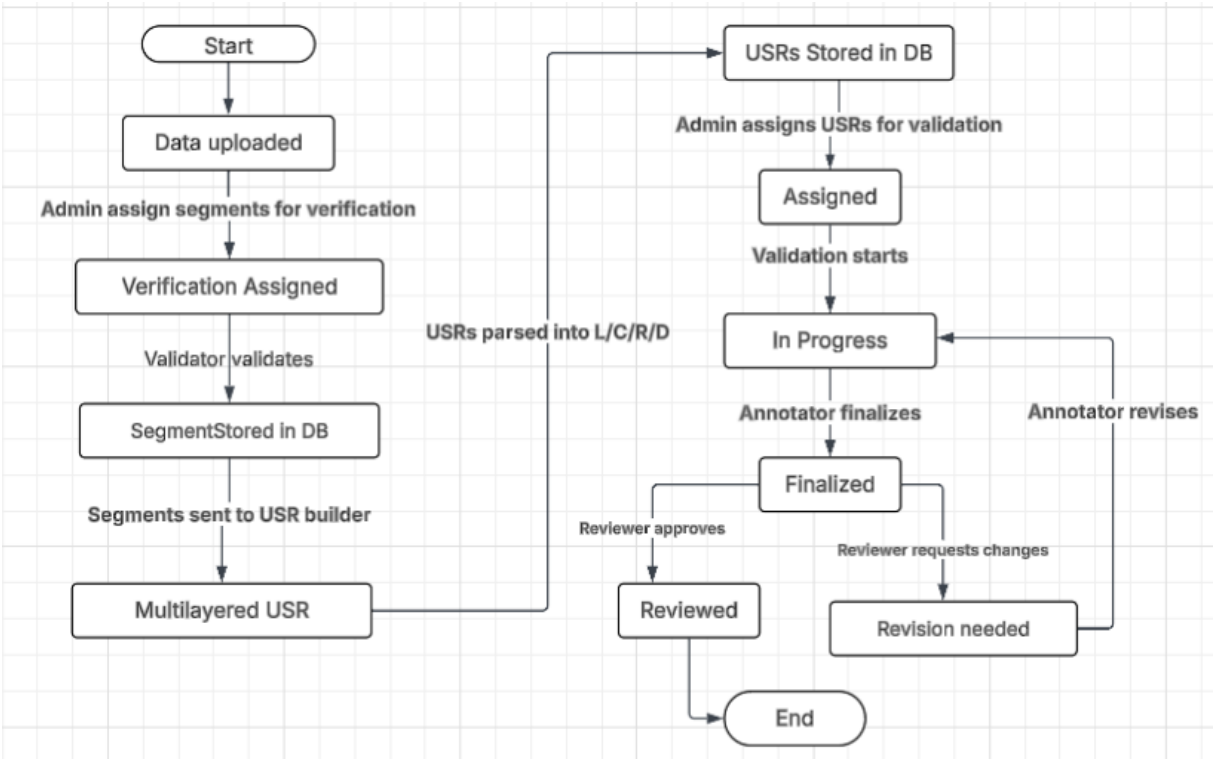


Figure 6: Overview of System Workflow of **SAVI**.

tribution of responses across SUS questionnaire items is shown in Figure 7. Most responses fall into the "Agree" or "Strongly Agree" categories, suggesting that users found the interface intuitive and well-integrated.

Additional usability feedback was collected through a short questionnaire assessing ease of use and interface features. Aggregated responses are summarized in Table 4. The majority of responses fall under the "Good" or "Excellent" categories, indicating that annotators found the interface easy to learn and effective for editing and vali-

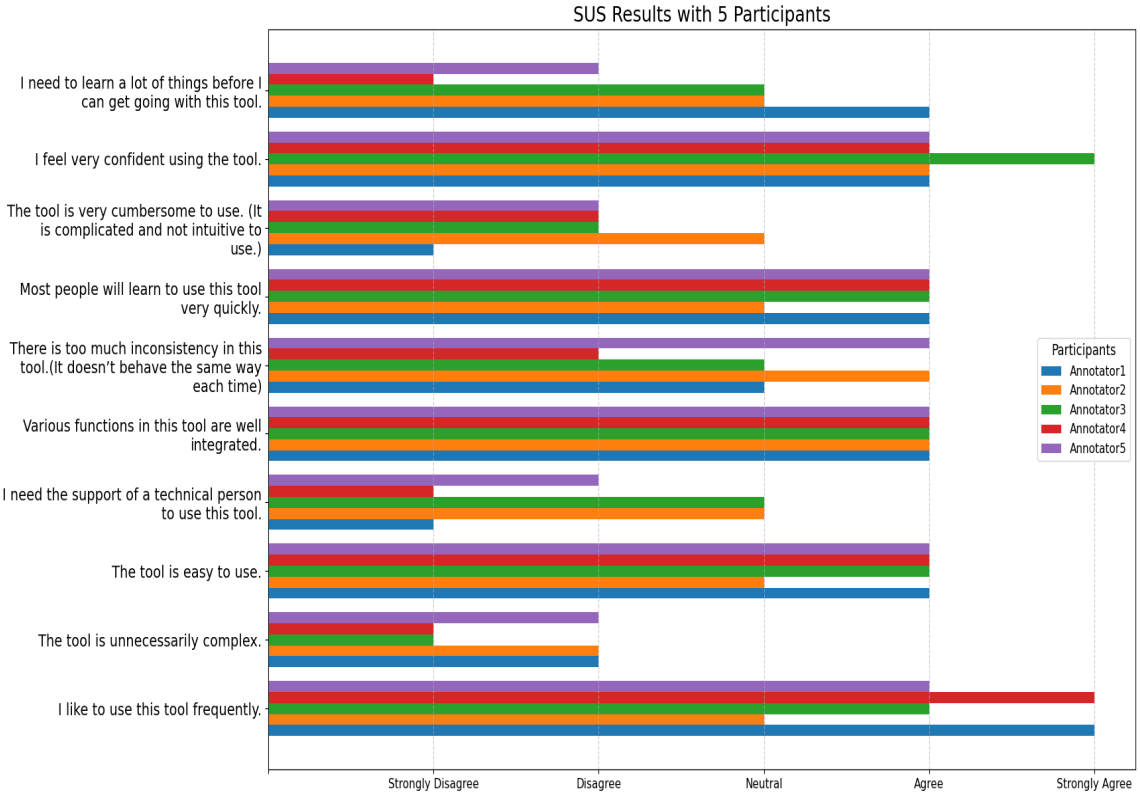


Figure 7: Distribution of System Usability Scale (SUS) responses across five annotators evaluating SAVI.

dation tasks.

Question	VP	P	Avg	G	Ex
Is the tool intuitive to use?	0	1	1	3	0
Is the tool easy to learn?	0	0	1	2	2
Is editing easy?	0	0	1	4	0
Is the tree/graph view helpful for editing?	0	0	1	3	1
Are features like search or shortcuts helpful?	0	0	1	3	1
Overall satisfaction with the tool	0	0	0	5	0

Table 4: User responses obtained from the usability survey on SAVI (VP=Very Poor, P=Poor, Avg=Average, G=Good, Ex=Excellent).

Technical performance responses indicated stable system behavior with predominantly instantaneous saving operations, as summarized in Table 5. Users reported moderate load times and fast response during editing operations.

Metric	Slow	Moderate	Fast
Load Time	0%	100%	0%
Saving Changes	0%	20%	80%

Table 5: User-reported technical performance of SAVI.

Early deployment showed minor stability issues, which were resolved through backend optimization and improved session handling. Overall, users reported that SAVI enables efficient multilayer validation while maintaining consistent annotation control.

6. Conclusion

This paper presented **SAVI**, a web-based interface for structured validation of Universal Semantic Representation (USR). Unlike conventional annotation tools, SAVI is specifically designed to support synchronized multilayer validation across lexico-conceptual, constructional, relational, discourse, and co-reference layers with enforced cross-layer constraints and dictionary-governed concept consistency.

Evaluation results demonstrate that SAVI supports controlled multilingual lexicon growth, reduces validation effort through a structured interface, and maintains multilayer structural consistency during annotation. Usability findings further indicate that tab-based separation and integrated visualization facilitate efficient and cognitively manageable validation workflows.

These results position SAVI as a structured validation framework suitable for scalable semantic corpus construction, particularly in low-resource and multilingual settings. Due to its modular and configurable architecture and a layered backend

design in which annotation types, tagsets, and validation constraints are defined independently of the interface logic, it can be adapted to other annotation schemes, also with minimal changes.

SAVI will be made publicly available as an open-source project under the GNU General Public License (GPL v3). The source code, along with detailed documentation and setup instructions, is available at: https://github.com/LC-Platform/Language_Communicator.git

Future work includes expanding AI-assisted validation mechanisms, improving automated structural suggestion modules, and extending support to additional languages and larger discourse-level corpora.

Limitations

The following are some of the limitations of our tool: (1) a preprocessing pass is required for the languages and scripts with complex segmentation, (2) specialized fonts or non-Unicode scripts may need normalization, (3) very large graphs (e.g., long-document discourse structures) may require chunked views and progressive rendering, and (4) for documents beyond the range of 5,000 tokens, increased memory consumption—primarily due to graph visualization and frontend rendering—can lead to slower response times.

Ethics Statement

SAVI is designed to support linguistic annotation and validation tasks and does not involve the collection of personal or sensitive user data. The system operates on textual datasets curated for research purposes, and no identifiable personal information is processed.

However, as with any language technology, biases present in the underlying data or annotation schemes may influence the outputs. Efforts should be made to ensure diversity and representativeness in annotated datasets. The system is intended for research use, and users are encouraged to critically assess outputs in downstream applications.

Acknowledgment

We are grateful to the annotators, especially Bidisha Bhattacharjee, Isma Anwar, Mohan, Rajni, Sabharaj, Satyaprakash, Sakshi, Muskan, Sanchari, Saumini, and Vandana, for their contributions to data preparation and experiments. This work forms part of the project titled "Sanskrit Knowledge Accessor" funded by MeitY, GoI, under the NLTM: BHASHINI scheme.

7. Bibliographical References

Omri Abend, Shai Yerushalmi, and Ari Rappoport. 2017. UCCAApp: Web-application for syntactic and semantic phrase-based annotation. In *Proceedings of ACL 2017, System Demonstrations*, pages 109–114, Vancouver, Canada. Association for Computational Linguistics.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik Van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. *arXiv preprint arXiv:1702.03964*.

Explosion AI. 2020. spacy: Industrial-strength natural language processing in python. https://spacy.io/.

Mike Bayer and contributors. 2006. Sqlalchemy: The database toolkit for python. https://www.sqlalchemy.org/. Version used in project may vary.

John Brooke. 1995. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.

John Brooke. 2013. Sus: a retrospective. *Journal of Usability Studies*, 8:29–40.

Benoît Chesneau and Gunicorn Developers. 2025. Gunicorn: Green unicorn {WSGI HTTP Server}. https://docs.gunicorn.org/. Accessed: 2025-08-28.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

John Ellson, Emden Gansner, Yifan Hu, Eleftherios Koutsofios, and Stephen North. 1991. Graphviz - graph visualization software. https://graphviz.org/.

Kirti Garg, Soma Paul, Sukhada Sukhada, Fatema Bawahir, and Riya Kumari. 2023. Evaluation of universal semantic representation (USR). In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 13–22, Nancy, France. Association for Computational Linguistics.

Google. 2017. Flutter: Ui toolkit for building natively compiled applications. https://flutter.dev/.

J. Klensin. 2008. Simple mail transfer protocol (smtp). Technical Report RFC 5321, Internet Engineering Task Force.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Jivri Mirovsky, Lucie Mladova, and Zdenvek vZabokrtsky. 2010. Annotation tool for discourse in PDT. In *Coling 2010: Demonstrations*, pages 9–12, Beijing, China. Coling 2010 Organizing Committee.

Ines Montani and Matthew Honnibal. 2018. Prodigy: A new annotation tool for radically efficient machine teaching. https://prodi.gy. Accessed: 2025-04-10.

Soma Paul, Sukhada Sukhada, Bidisha Bhattacharjee, Kumari Riya, Sashank Tatavolu, Kamesh R, Isma Anwar, and Pratibha Rani. 2025. Indian grammatical tradition-inspired universal semantic representation bank (USR bank 1.0). In *Proceedings of the 1st Workshop on Benchmarks, Harmonization, Annotation, and Standardization for Human-Centric AI in Indian Languages (BHASHA 2025)*, pages 11–22, Mumbai, India. Association for Computational Linguistics.

PostgreSQL Global Development Group. 1996. Postgresql: The world's most advanced open source database. https://www.postgresql.org/.

Python Software Foundation. 1991. Python programming language. https://www.python.org/. Version 3.x.

Armin Ronacher and the Pallets team. 2010. Flask: A lightweight wsgi web application framework. https://flask.palletsprojects.com/. Version used in project may vary.

Axel Sorensen, Siyao Peng, Barbara Plank, and Rob Van Der Goot. 2024. EEVEE: An easy annotation tool for natural language processing. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 216–221, St. Julians, Malta. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Sukhada Sukhada, Sirisipalli Veera Hymavathi, and Soma Paul. 2023. Generation of mrs abstract predicates from paninian usr. In *Proceedings of the 30th International Conference on Head-Driven Phrase Structure Grammar, University of Massachusetts Amherst*, pages 122–142, Frankfurt/Main. University Library.

Sukhada Sukhada and Soma Paul. 2023. Theory of sāmarthya in Indian Grammatical Tradition: The foundation of Universal Semantic Representation. In *Int J Sanskrit Res*, pages 17–22.

Francis M. Tyers, Maria Sheyanova, and Jonathan Washington. 2018. Ud annotatrix: An annotation tool for universal dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16)*, pages 10–17, Prague, Czech Republic. Association for Computational Linguistics. Distributed under a CC-BY 4.0 licence.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. WebAnno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

Jin Zhao, Nianwen Xue, Jens Van Gysel, and Jinho D. Choi. 2021. UMR-writer: A web application for annotating uniform meaning representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 160–167, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Finding Meaning in Embeddings: Concept Separation Curves

Paul Keuren[*†], **Marc Ponsen** [†], **Robert A. Bagheri**[*]
[*]Utrecht University
[†]Statistics Netherlands
p.j.g.keuren@uu.nl

Abstract

Sentence embedding techniques aim to encode key concepts of a sentence's meaning in a vector space. However, the majority of evaluation approaches for sentence embedding quality rely on the use of additional classifiers or downstream tasks. These additional components make it unclear whether good results stem from the embedding itself or from the classifier's behaviour. In this paper, we propose a novel method for evaluating the effectiveness of sentence embedding methods in capturing sentence-level concepts. Our approach is classifier-independent, allowing for an objective assessment of the model's performance. The approach adopted in this study involves the systematic introduction of syntactic noise and semantic negations into sentences, with the subsequent quantification of their relative effects on the resulting embeddings. The visualisation of these effects is facilitated by Concept Separation Curves, which show the model's capacity to differentiate between conceptual and surface-level variations. By leveraging data from multiple domains, employing both Dutch and English languages, and examining sentence lengths, this study offers a compelling demonstration that Concept Separation Curves provide an interpretable, reproducible, and cross-model approach for evaluating the conceptual stability of sentence embeddings. The code is open source and located on github, and a live interactive demo is available at streamlit.

Keywords: sentence embedding, embedding evaluation, conceptual representation, concept separation curves, large language models

1. Introduction

How can we be certain that a Large Language Model (LLM) is capable of distinguishing between concepts? Oftentimes, this is tested by prompting an LLM and inspecting the agreement of the result (Kıyak and Emekli, 2024; Yetisensoy, 2025). The problems with this approach are twofold. Firstly, it is reliant on costly human annotations (Wang et al., 2021). Secondly, it makes the implicit assumption that correct answers necessarily reflect genuine understanding, even though models may succeed without representing the underlying concepts(Adi et al., 2017; Bender and Koller, 2020). The annotations become even more difficult when such an answer cannot be evaluated by hand, as is the case with the output from sentence encoders (Vaswani et al., 2017).

In this research, we propose a method which resolves the annotation issue and sheds light on the understanding. Our method, summarised in Figure 1, is applied to different sentence embedding methods. By taking a corpus and performing perturbations, we can derive a measure of concept separability by the embedding method. The perturbations are key, as they are required to embody different concepts. A concept is defined as a thought or

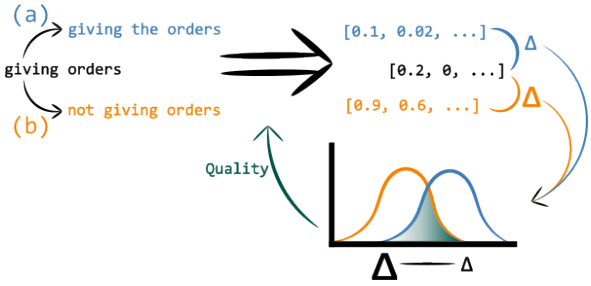


Figure 1: Concept Separation Curves. This example has been translated from the Dutch sentence "bevelen geven" (giving orders), which originates from the CompetentNL dataset. Initially, a set of perturbations is computed for a given sentence: a) surface-level perturbation, and b) semantic change. Following the process of embedding each sentence, the difference per vector is measured. The application of this process to the entire corpus provides insight into the quality of the embedding, as demonstrated by the overlap between the curves.

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 91–101
11 May 2026.

idea (Vocabulary.com, 2025), and as such, it can be difficult to generate sentences with different concepts automatically. Due to this, we define a set of rules for the perturbation and implement an example in the form of negation.

Unlike existing evaluation approaches((Adi et al., 2017)), our method does not rely on annotated datasets or classifiers; instead, it isolates semantic changes (such as negations) from surface-level perturbations (such as added noise). This makes it possible to directly assess how sensitively an embedding represents meaning, independent of external modelling choices. Although we use our approach on data in two languages (Dutch and English), it is designed to generalise across others as well.

To make these claims concrete, we make the following contributions:

- We introduce **Concept Separation Curves (CSCs)**, a classifier-independent method for evaluating sentence embeddings by contrasting semantic and surface-level perturbation.
- We propose **an automated, annotation-free perturbation framework** based on controlled alteration.
- We define a geometric overlap measure to quantify separation between semantic and non-semantic effects in embedding space.
- We analyse embedding behaviour across models, languages, and sentence lengths, identifying failure modes related to token position sensitivity and sentence length.

In the rest of the paper, we will delve into related work. Then, we describe our methods, which focus on language analysis instead of expert knowledge. Finally, we discuss the results and implications of our methods.

2. Related work

To give a better overview, we split the related work into two groups: **perturbations** and **concept validity**.

The impact of text manipulation on the resulting embeddings has been a subject of research for some time. One of the most relevant to this paper is the one by Wang et al. (Wang et al., 2022). They make a distinction between different types of automated negations of text to train an embedding. As such, it is closely related to both the Mission Impossible Languages by Kallini et al. (Kallini et al., 2024) method and GPT understanding by Liu et al. (Liu et al., 2024). All these methods alter input text to train LLMs; the latter two (by Kallini and Liu) focus on GPT, the first focuses on embedding. Our method also has a focus on embeddings, yet we do not train on the generated data. Furthermore, we also insert terms to create noise to compare with the negations.

Concept validation of embeddings in NLP has been studied in various contexts, including semantic similarity, interpretability, and alignment with human judgments. One of the studies in this area was conducted by Fang et al. (Fang et al., 2022). In this study, they describe how they alter and measure the ESS questionnaire texts. They use a manual approach to generating texts, which ought to be more or less similar, describing the same properties. The difference between similar and dissimilar is then used as the basis for a boxplot. In our research, we do not alter any text manually, nor do we annotate the expected outcome. We base ourselves purely on grammar, thus reducing bias in the generated output. This is not to say that our method has no bias, but its impact is expected to be reduced (Dror, 2020).

3. Methods

To measure whether a language model's capability of discerning between concepts, we introduce CSCs. CSCs quantify how sensitively an embedding model reacts to semantic variations (meaning changes) compared to non-semantic variations (surface-level perturbations). This allows us to assess whether embeddings preserve conceptual meaning when sentence form changes. The curves visualise the difference between the two types of alteration. Specifically, each sentence is modified through two parallel processes: Fuzzing, which introduces non-semantic variations, and Negation, which introduces semantic vari-

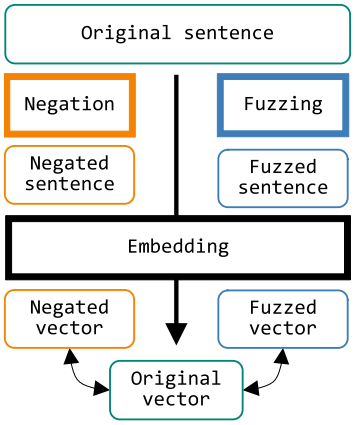


Figure 2: Approach setup, square components are algorithmic processes. This setup summarises the pipeline: from text alteration to embedding and similarity computation.

ations. In this research, we use the terms Fuzzing and Negation as this makes the distinction between the different alterations easier to tell apart. Our framework is not limited to strict Fuzzing and Negating, nor is the Negation a full opposite of the original sentence. The embeddings of the altered and original sentences are then compared, and when applied across an entire corpus, the resulting distribution patterns form the CSCs. The overview of the process is shown in Figure 2.

3.1. Data

The data used in this paper is intended to show the broad applicability of our method. Furthermore, we wish to evaluate whether our method indeed works across domains and languages. To evaluate our method, we aim to reduce the impact of other variables. The two variables we wish to isolate in terms of effect are language and sentence length. The amount of data available in a language has been shown to impact the quality of embedding algorithms (Koto et al., 2024). Hence, we use sources from two languages with differences in the amount of available content: Dutch and English. For estimating the differences in available content, we used the number of available Wikipedia articles as a heuristic (Wikipedia, 2025a). This source was selected as it is commonly used for training many of the state-of-the-art LLM encoders. In terms of difference, at the time of writing, the number of English pages is $62.3 * 10^6$ with a total of $48.0 * 10^8$ words (Wikipedia, 2025b). This contrasts with the Dutch language, totalling $4.68 * 10^6$ pages and $5.30 * 10^8$ words (Wikipedia, 2025c). Such a large difference between the languages might be reflected in the performance of the algorithms. The sentence length might also significantly impact the results. As previously mentioned, the research by Liu et al. (Liu et al., 2024) shows the impact of inserting a singular term. In our method, we also describe the process of adding a word to a sentence. The impact in terms of volume strongly depends on the number of words present. Altering a singular word in a five-word sentence can have a larger impact than on a twenty-word sentence. We selected the corpora based on finding and detecting any of these issues.

The corpora we selected are CompetentNL (CNL), ESS Questionnaire (ESS), and Paracrawl (PC) (Bañón et al., 2020). CNL is a short sentence Dutch corpus describing skills. For this research, we define a short sentence corpus as one where the majority of the sentences contain fewer than 10 tokens. The ESS is the same as used by Fang et al. (Fang et al., 2022), it comprises a set of English questions of varying length. PC is a corpus of both Dutch and English texts from web pages. These crawled texts have a large variety of sentence lengths. To give a better overview of the statistics per source, an overview is given in Table 2. The main difference between PC and the others is that it is not as strongly bound to a singular domain. CNL limits itself to short skill descriptions, and ESS specifically to questions from a singular questionnaire. Hence, both can be labelled as narrow datasets. The PC, however, is selected not to test on the domain, but to serve as a test of both language and sentence length. The relationship between the sources regarding the main identified variables is depicted in Table 1. In this Table, we also introduce the PC_filtered set. This is a subset of the PC dataset, which is reduced to sentences with the same length as the CNL dataset. This combination of sources is expected to give a good impression of the described variables.

	Sentence length	
	Short	Long
Dutch	CNL, PC_filtered	PC
English	PC_filtered	ESS, PC

Table 1: Overview of the different data sources used and key properties.

# Tokens	CNL	ESS	PC_EN	PC_NL
0-10	99.26%	5.32%	32.99%	33.41%
10-20	0.72%	30.85%	31.74%	31.21%
20-30	0.02%	26.60%	18.92%	18.71%
30-40	0.00%	26.60%	8.96%	8.85%
40-50	0.00%	10.64%	3.73%	4.05%
≥50	0.00%	0.00%	3.66%	3.77%
Sentences	4738	94	2560472	2560472

Table 2: This table shows, per source, the percentage of sentences with a token count in a given range. Below the percentages, the total number of sentences from which the tokens were extracted is shown.

3.2. Fuzzing and Negation

Each of the sentences from the sources goes through two modification processes: Fuzzing and Negation, as shown in Figure 2.

Fuzzing serves as a control condition that introduces minimal, non-semantic textual perturbations, allowing us to test how stable an embedding remains when surface form changes but meaning is preserved. We define Fuzzing as the alteration of a sentence without altering its concept.

Negation, in contrast, represents a targeted semantic perturbation: it minimally changes the surface form while significantly altering the sentence's conceptual meaning. This allows us to examine whether embeddings are sensitive to genuine semantic shifts rather than superficial textual ones.

Furthermore, we limit the Negation to change an equal number of tokens as the Fuzzing. This restriction on the Negation is to ensure that any measured effect can only be explained by the contents of the change, not by differences in the number of inserted tokens.

In this research, we implement both Fuzzing and Negation through token addition. For Fuzzing, we insert articles in the respective languages, "de" or "het" in Dutch, and "a" or "the" in English. This simple insertion strategy aligns with many embedding methods that operate at the token level (Vaswani et al., 2017; Devlin et al., 2019a). Accordingly, we use a single-token insertion to introduce controlled surface-level variation. For Negation, we insert a single negative particle: "niet" in Dutch and "not" in English. This addition minimally changes the sentence's structure while reversing its propositional meaning.

The insertion procedure is visualised in Figure 3. The full procedure works as follows: First, all viable locations for insertion are detected (displayed in red under the text). Viable locations are defined as in front of any word in a sentence. Second, given the available insertion terms (in green), create a list of all possible combinations of terms and locations (the list in blue). This combination is randomly shuffled. Finally, select and perform up to X options (red circles surrounding items in the blue list). Each option results in a new sentence (as depicted at the bottom of the figure). This procedure was chosen to be able to limit the data generation. Short sentences do not support the same number of possible perturbations. As such, longer sentences could be overrepresented in the curves if the X value is set too high. There is also the increase in computations for higher X values. As the impact of X is described by: $2X * n$ with n being the size of the corpus. Combining these factors, for our research we chose an X value of 3. This data generation procedure works for both Fuzzing and Negation.

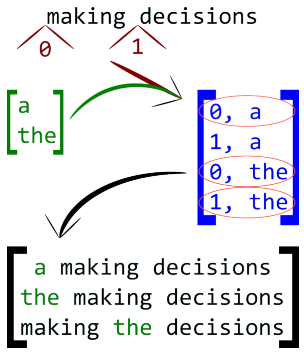


Figure 3: Depiction of the sentence generation process for the Fuzzing. Parts not visualised are the random shuffling. This sentence is translated from "beslissingen maken" from the CompetentNL source.

Although this algorithm is identical for Neg-

ating and Fuzzing, the number of sentences it returns for both is not guaranteed to be the same. As depicted in Figure 3, the insertion locations are base on the sentence and as such do not differ between Negation and Fuzzing. The insertable items (shown in green in the figure) could differ between the operations. Thus, the possible number of generated sentences (based on the length of the list in blue) could differ as well, given that each pair corresponds to an output sentence. This difference is handled by normalisation in the comparison step (explained in subsection 3.4). For now, it is important to note that the chosen Fuzzing and Negation, although identical in their algorithm, are not identical in the expected output volume.

3.3. Models

In this research, we focus on understanding embedding methods. Any model fitting the capabilities of turning a text into a vector (as depicted in Figures 2 and 1) can be used. As such, a non-Dutch model could be applied to the Dutch language. However, such a mismatch ought to result in worse outcomes. To test this, we apply every model to every source, even if there is a mismatch between the model's supported language and the source language.

Firstly, for a baseline, we use **Term Frequency Inverse Document Frequency (TFIDF)** (Pedregosa et al., 2011). This is a simple yet effective method which does not consider word order and is not trained on a large corpus. It is common to use a stopword removal step before applying this algorithm. Yet, for our method to work, it is critical not to remove stopwords which might be added in the Fuzzing or Negation steps. Therefore, instead of curating a stopword list, we decided to omit the stopword removal altogether.

The second approach we use is **Fasttext** (Joulin et al., 2016). This model is available in both Dutch and English in a pre-trained format. Although it is an older model, it is a method which can embed sentences, making it a suitable baseline for embedding-based methods.

Finally, we use multiple **state-of-the-art sentence embeddings**; GroNLP (de Vries et al., 2019), MPNET (Song et al., 2020), RobBERTa (Delobelle et al., 2020) and LaBSE (Feng et al., 2022). These models have been pre-trained on different sources. GroNLP is specifically trained on Dutch corpora, although it might contain some English terms. The MPNET and RobBERTa models have been trained on English corpora, while LaBSE is designed for cross-lingual applications using a variety of languages. All of these models are based on the original BERT (Devlin et al., 2019b), yet each is configured or trained differently to suit specific needs.

3.4. Concept Separation Curves

The goal of Concept Separation Curves is to illustrate the understanding of a text embedding model without annotations. This method does so by generating at least three vectors for one text: the original text vector, the fuzzed vector, and the negated text vector. The hypothesis for our method centres around two different observables:

- The Fuzzed Vectors should stay close to the original embedding. The meaning is the same, but a change was made regardless.
- The Negated Vectors should show a difference with the original, which is larger than the Fuzzed Vector differences. The meaning differs from the original; as such, the impact on the vector ought to be greater.

To perform this comparison, we propose **Concept Separation Curves** (CSCs). CSCs are a visualisation of how a text embedding technique responds to concept alteration compared to textual alteration. The response to concept alteration and textual alteration impact curves are both plotted in the same graph. These curves are made by comparing the vectors of the original sentence with both the negated and fuzzed vectors. This results in two curves, one for the negations and one for the fuzzing. These curves are intended to show the sensitivity to Fuzzing and how different the Negation impacts the resulting vectors. To improve the readability of the curves and focus on the underlying distribution, we perform a Gaussian kernel density estimation function (Virtanen et al., 2020) on the raw data.

The resulting curves can differ wildly as the Negation and Fuzzing process do not guarantee an equal output volume (as explained in subsection 3.2). To this end, we perform the surface normalisation as defined in equation 1.

$$norm(d, i) = \frac{d_i}{\sum_{j=-1}^{1} d_j} \quad (1)$$

In this normalisation, let d represent the density and i the inspected value within the range [-1,1], and d_i the density at i. Just like i, j also iterates over the values in d, meaning that for both there is an overarching resolution. The meaning of this resolution is the number of steps to inspect in this [-1, 1] range. Intuitively, this normalisation ensures that the total area under each density curve equals 1, allowing a fair comparison between the Negation and Fuzzed distributions regardless of their differing volumes. The resulting values are plotted in a line plot for both the negated and fuzzed densities.

One aspect which can be difficult to spot in the plot is the shared surface between the curves. Due to possible small differences across all similarities, these differences can add up without being easy to spot. To circumvent this, we also compute the shared surface using the equation shown in equation 2.

$$\sum_{i=-1}^{1} min(norm(fuz, i), norm(neg, i)) \quad (2)$$

In this equation, fuz stands for the fuzzed similarity density distribution and neg for the negated. The function describes how we normalise these density distributions of f and n, resulting in each adding up to 1. The normalisation is required to compensate for the potential difference in volume. This difference in volume is expected due to potential differences in negated terms and fuzzed terms (as explained in subsection 3.2). Then we are interested in the overlap between these values, and go through values i from -1 to 1, taking the minimum value between fuz and neg. This results in an overlap score ranging from 0 to 1, where 0 indicates no overlap and 1 is a perfect overlap.

Finally, there is a chance that some alterations in the vector are due to the text being out of distribution. Because our generated sentences are not expected to always have correct grammar, it might be that embedding models like BERT generate wildly different vectors. To check for this, we use a method based on Wang et al. (Wang et al., 2022) to generate negations with more grammatically correct negations. In their original method, they add a negation if none is present and add one if not present. We adjust this by only allowing the addition. Furthermore, this method is specific to the English language. Thus, we only apply it to the English corpora.

4. Results

In this section, we present the main patterns observed in the CSCs, illustrating both desirable and failure-case behaviours, followed by a quantitative analysis of their overlap. First, we start with an expected and desired curve, followed by different types of negative results.

An example of good concept separation is shown in Figure 4. In this figure, CNL data is used in combination with the GroNLP model. This plot shows a nice separation between concepts. The fuzzed curve is close to 1, thus showing a high similarity to the original. The negated sentence curve is to the left of the fuzzed curve, showing an increased dissimilarity to the original sentence. As such, it displays that the encoded vector governs a different concept. The overlap is indicative of two groups: fuzzed sentences where the Fuzzing had a large impact, and Negations with a small impact. As this overlap is small and the Negations are more dissimilar to the original compared to the fuzzed sentences, this displays a good concept separation.

Not all embeddings perform as nicely as the already shown GroNLP. For instance, the FastText embedding, as shown in Figure 5. Here, the Negations are more similar to the original sentence than the fuzzed sentences. As such, this algorithm cannot be stated to have encoded the concept of a sentence. Another example of a less-than-desirable result is the one shown by sBERT MPNET in Figure 6. Although the fuzzed sentences are slightly more similar to the original, there is a second peak near -1. This can be seen across datasets and

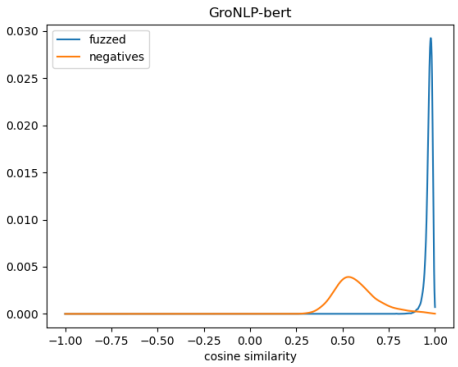


Figure 4: Concept Separation Curves using Gaussian kernel density estimation on the CNL data and the GroNLP embedding model. This graph shows an overlap of 0.0221.

languages. Given that the only constant in our alteration is the addition of a token, we have to conclude that this algorithm reacts heavily to the change of token position. As such, it behaves more akin to a hashing algorithm rather than a concept embedding. Both these patterns show that the used embedding has difficulty discerning concepts.

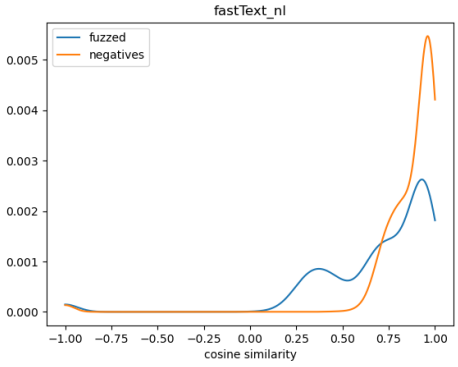


Figure 5: CNL data with FastText embedding. The overlap is 0.6652

The final pattern of note we discovered in our results is the sentence length effect. This effect is visible in Figure 7. The unfiltered and filtered data are from the same domain and language. The only difference is the length of the sentences. The curves of the unfiltered, therefore longer texts, are less pronounced and thus more difficult to perceive. With even longer sentences, this effect is expected to worsen. As such, we see that with increased sentence length, our method decreases in its detection capabilities.

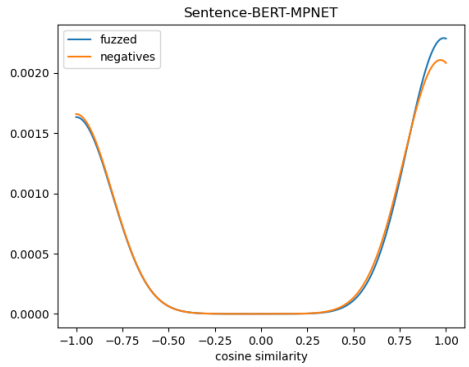
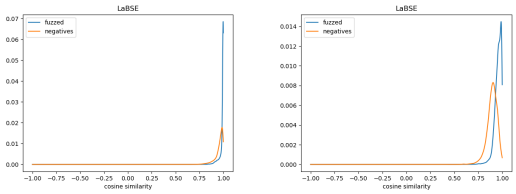


Figure 6: ESS data with sBERT MPNET embedding. The overlap is 0.9810



(a) Unfiltered, 0.5168 (b) Filtered, 0.4632

Figure 7: LaBSE algorithm on PC NL data, in both its raw form and filtered to the short sentence lengths present in the CNL dataset.

In previously shown results, the overlap is included in each plot. The complete overview is given in Table 3. In this Table, there are stark differences between models. The Dutch GroNLP appears to be the best at each Dutch dataset, and the same is the case for LaBSE in English. Furthermore, some models appear to have a near-perfect overlap as the values are close to 1. This holds for longer sentences as well, even though optical inspection of the curves for this data is more difficult.

5. Discussion

In this paper, we demonstrated a method for measuring the concept certainty of embedding algorithms. Unlike other approaches, our approach does not rely on human annotations, nor does it utilise a classifier. This makes our method easy to apply in other languages and even makes it accessible per domain.

Effects of Sentence Length and Token Position. When inspecting the results, the length of a sentence appears to be the strongest limiting factor. This makes sense as we alter a

	CNL	PC NL Filt.	PC EN Filt.	PC NL	PC EN	ESS
TFIDF	0.2993	0.7684	0.6015	0.9449	0.8614	0.4516
GroNLP	**0.0221**	**0.0861**	0.8056	**0.2925**	0.9449	0.8767
LaBSE	0.1984	0.4632	**0.3588**	0.5168	**0.5635**	**0.3511**
Fasttext NL	0.6652	0.6825	0.8809	0.7402	0.8692	0.8072
Fasttext EN	0.7764	0.7977	0.7652	0.8804	0.9405	0.9549
MPNET	0.9756	0.9813	0.9496	0.9874	0.9523	0.9810
RobBERTa	0.9605	0.9726	0.9488	0.9905	0.9410	0.9773

Table 3: Normalised area overlap of Fuzzed and Negated curves.

smaller percentage compared to the size of the sentence. Altering in a significant manner for such text would either involve larger insertions or shortening the sentences first. The problem with the larger insertions is that with each insertion, some meaning could be altered. For instance, in short sentences, there is already some impact with the Fuzzing, as there might be a difference between "a bike" and "the bike". This effect is cumulative; thus, when applied in larger volumes, the concept is nearly bound to change. As such, we expect that it might be beneficial for our method to reduce sentences to a shorter format.

Next to the sentence length, some algorithms struggled with the position of words. Especially in short sentences, the MPNET and RobBERTa algorithms changed the vectors drastically for both Negation and Fuzzing. Whilst the impact of Fuzzing and Negation ought to be different, they were nearly negligible. We believe this is due to the position of the tokens in the sentence. These results are more akin to a positional hashing algorithm, rather than a content embedding. Given that both insertion algorithms do not append at the end of a sentence, this might result in the findings for both algorithms.

Implications for Embedding Evaluation. Finally, the impact of language on the task appeared to be present, yet minor. When inspecting Table 3 and the Figures, a pattern emerges that certain models perform best for a specific language. Other than the best, there is a difference between the languages in terms of the spread of overlap. When looking at the statistics between the English data and Dutch data, we see that Dutch has an average of 0,6668 (σ 0,2658) and English 0,7992 (σ 0.1622). This difference is not completely fair, as there are more English models than Dutch models in our test. However, it is a rather striking difference, which could be explained by differences between languages.

6. Conclusion

In this research, we looked into a method for finding the concept validity of embedding models. Our method encompassed a difference analysis of Fuzzing and Negation to accomplish this. We found that there are differences between models in terms of validity. The proposed Concept Separation Curves visualise and quantify how much a sentence embedding cares about meaning. Furthermore, some models were unable to distinguish between the same and different concepts, whereas others were not. Additionally, we found that some algorithms reacted heavily to the position of tokens. Given the range of applied datasets, we expect this effect is not tied to the used data in terms of length and language. As such, it appears these models do not capture the underlying concepts.

7. Limitations

A potential limitation of the present study is its reliance on the incorporation of terms into a sentence. However, it should be noted that this method may not be universally applicable, as in some languages, addition cannot be used to obtain a negative value. Consequently, a potential avenue for future research could involve exploring alterations as a substitute for additions

An additional constraint is that the available resources for the languages do not encom-

pass antonyms and synonyms. Instead of inserting terms for both Fuzzing and Negation, it would be possible to use synonyms and antonyms for these operations. An effort was made to use publicly available antonyms and synonyms. However, analysing the number of synonyms and antonyms in our dataset showed that, on average, each sentence had fewer than 1 word from either, with a negligible number of sentences containing both a synonym and an antonym. As such, it was deemed not viable, for it would not alter enough sentences. If a more complete set of synonyms and antonyms were available, this would be of interest to compare against the obtained results from this research.

An important note to the results presented in this research is that our measure is not directly related to its capabilities in other tasks. The simplest example would be a list comprising all search terms in a domain. When looking for an item, it can be retrieved quickly and precisely, even while the user is still typing the full query. Such a system would not have high concept validity. As the Fuzzing might completely offset its results, a precise position might be relevant. Thus, our research does not indicate the performance of other tasks It can be used, however, in domains where certainty plays a key role. The implications of our research for the results in other fields, however, are a topic which might warrant further research.

Another point of interest is the extension of the proposed method. In the results, we highlighted different stereotypical curves. These effects are not guaranteed to be visible from the total overlap. An example would be an embedding where the Fuzzing has a peak near -1 and a Negation near 1. Such a curve would display that the method can make a distinction between concepts and correctly have an overlap value of 0. However, the problem would be that there might be some underlying effect causing such an abnormality. Possibly due to token order sensitivity and not filtering the negated words. We did not encounter this effect, but the method could be extended to include measurements to check for such an effect.

The measurement employed, the cosine similarity, is another potential avenue for enhancement. The usage of the cosine similarity reduces the comparison to a single number, whilst the true change of the perturbation might be too local for a significant change. As such, a model with a higher dimensionality (for instance, 700+), reacting to our method on just 1 dimension, would perform worse compared to a lower-dimensional model. To correct for this, a more thorough comparison of the vectors could be devised.

Finally, it is possible to extend our method to different types of input alteration. These alterations could involve altering a noun or verb, compared to changing a domain-specific irrelevant term. An example would be: "Applicant should be able to drive a car", Fuzzing it into "He should be able to drive a car" and the alteration being "Applicant should be able to drive a boat". The problem we found with such alterations is that they are more domain-bound and require more complex algorithms to automatically construct (if at all possible). The alterations themselves could be done in a more fine-grained approach tailored to a specific domain.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-Scale Acquisition of Parallel Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of*

the association for computational linguistics, pages 5185–5198.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. arXiv:1912.09582.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: A Dutch RoBERTa-based Language Model.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Itiel E. Dror. 2020. Cognitive and Human Factors in Expert Decision Making: Six Fallacies and the Eight Sources of Bias. *Analytical Chemistry*, 92(12):7998–8004.

Qixiang Fang, Dong Nguyen, and Daniel L. Oberski. 2022. Evaluating the construct validity of text embeddings with application to survey questions. *EPJ Data Science*, 11(1):39.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible Language Models.

Yavuz Selim Kıyak and Emre Emekli. 2024. Chatgpt prompts for generating multiple-choice questions in medical education and evidence on their validity: a literature review. *Postgraduate Medical Journal*, 100(1189):858–865.

Fajri Koto, Tilman Beck, Zeerak Talat, Iryna Gurevych, and Timothy Baldwin. 2024. Zero-shot sentiment analysis in low-resource languages using a multilingual sentiment lexicon. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 298–320, St. Julian's, Malta. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2024. GPT understands, too. *AI Open*, 5:208–215.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors.

2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Vocabulary.com. 2025. concept - dictionary definition. Accessed: 2025-11-20.

Hao Wang, Yangguang Li, Zhen Huang, Yong Dou, Lingpeng Kong, and Jing Shao. 2022. SNCSE: Contrastive Learning for Unsupervised Sentence Embedding with Soft Negative Samples.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wikipedia. 2025a. List of wikipedias. https://en.wikipedia.org/wiki/List_of_Wikipedias. Accessed: 2025-1-23.

Wikipedia. 2025b. Wikipedia statistics. https://en.wikipedia.org/wiki/Special:Statistics. Accessed: 2025-1-23.

Wikipedia. 2025c. Wikipedia statistieken. https://nl.wikipedia.org/wiki/Speciaal:Statistieken. Accessed: 2025-1-23.

Okan Yetisensoy. 2025. Validity challenge in genai models: Evaluating the validity of content generated by text-to-image models in the context of social studies education. *Journal of Pedagogical Research*, 9(4):81–101.

Extracting First Order Logic formulas from graphical semantic representations

Rémi de Vergnette, Vincent Tourneur, Maxime Amblard
LORIA, UMR 7503, Université de Lorraine, CNRS, Inria
54000 Nancy, France
remi.de-vergnette@loria.fr, vincent.tourneur@loria.fr, maxime.amblard@univ-lorraine.fr

Abstract

In this paper, we present a method for interpreting YARN structures as logical formulas in a modal first order logic with temporality. YARN is a recent semantic formalism that aims to bridge the gap between graph-based and logic-based semantic representations, providing a flexible and expressive framework for capturing the meaning of natural language utterances. Our approach translates the elements of YARN structures such as predicates, features, into corresponding logical constructs, allowing for an interpretation of the represented meaning. Given that YARN allows underspecified scope, we associate to each YARN structure a set of possible logical interpretations. We account for a range of semantic phenomena, extending beyond ambiguity to capture aspects of dynamic quantification as well. This work contributes to the understanding of the expressive power of graphical semantic representations and their relationship to formal logic.

Keywords: Semantics, Logic, Computational semantics, FOL, modal semantics

Introduction

Semantic formalisms represent a symbolic and explicit approach to capturing and manipulating the meaning of natural language utterances. In contrast to task-oriented representations of semantics, which are generally learned end-to-end using a specific or generic learning task (in the context of transfer learning) and are naturally embedded in continuous vector spaces, semantic formalisms provide explicitly structured representations that aim to capture meaning in a systematic and interpretable way.

Semantic formalisms for Natural Language Processing (NLP) include several advantages. First, their support for compositionality, where the meaning of complex expressions can be derived from the meanings of their parts (Frege, 1891, 1893). Secondly, they allow enhanced explainability, as the structured intermediate representations allows to formalise reasoning in explicit terms, making part of the process more transparent (Nguyen et al., 2025). They also provide easier debiasing, since explicit representations can be made "blind" to certain aspects of the data more systematically (Lobo et al., 2023), and increased frugality, as on one hand semantic representations can be reused across multiple tasks rather than requiring task-specific learning from scratch (Wein and Opitz, 2024), and on the other hand, they tend to build on logical frameworks that enable the use of efficient algorithms for reasoning and inference. Third, semantic annotations have been shown to provide a useful complementary signal to LLMs, for instance in under-resourced languages (Wein, 2025) situations.

The expressivity of structures, relating to what they can represent, provides the absolute bound on the utility of any meaning representation. For instance, when using AMR (Banarescu et al., 2013) alone, which does not fully represent scope, automated natural language inference becomes problematic. Indeed, since two formulas with different scope configurations may have the same representation, it is impossible to process them differently when using AMR alone.

A tension emerges: the more popular formalisms (which are also the ones for which efficient parsers exist) tend to be the least expressive (Pavlova et al., 2023b; Crouch and Kalouli, 2018), while more expressive formalisms, often based on extensions of lambda calculus (Maršík et al., 2021; Amblard and Retoré, 2014) or formal logic (Bos, 1996), are more difficult to parse because of strong structural properties. This implies that obtaining annotations ffor such formalisms generally requires more work.

The introduction of the YARN formalism (Pavlova et al., 2024) is an attempt at resolving this tension by providing a way to represent high-quality structures in terms of semantic, logical, and even syntactic and discursive properties, while maintaining a permissive, scalable, and easy-to-annotate framework for humans. YARN is not alone in extending and refining AMR: UMR (Bonn et al., 2024) represents another such approach, similarly employing a second annotation level to capture higher-level linguistic phenomena. We will be interested in the theoretical properties of YARN structures, more specifically in their expressive power.

Grounding graphical semantic formalisms in formal logic has been explored for related formalisms like AMR (Bos, 2016; Lai et al., 2020), using lambda calculus. (Pavlova et al., 2023a) uses Graph Rewriting Systems for mapping two different semantic

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 102–112
11 May 2026.

formalisms, DRT and AMR. The intermediate scope constraint system we present here is similar to the semantic formalism introduced by (Bos, 1996).

In this paper, we propose how to transform a YARN structure into a set of logical formulas. This provides a way to interpret YARN structures in a logical framework, allowing us to bridge the gap between representations of meaning in YARN and logic formalisms. For the most logically inspired formalisms like DRT (Kamp and Reyle, 1993), there are direct conversions to first order logic. However, given the possibility for underspecification and ambiguity in YARN structures, and the far less constrained structural properties, we will interpret YARN structures as encoding a set of constraints on admissible logical representations.

This is similar to other semantic frameworks such as MRS (Copestake et al., 2005) or Predicate Logic Unplugged (Bos, 1996). Note that since AMR does not encode many logic relevant aspects such as scope, it can be considered an underspecified formalism too. YARN is a superset of AMR and the added features allow more control over the possible interpretations.

After introducing the YARN formalism in Section 1 and the logical framework we use in Section 2, we present in Section 3 our main contribution: we propose an explicit translation from a YARN structure Y to an intermediate representation consisting of a pair (F, R), where F encodes scope dependencies as a labelled forest and R encodes flat predicate-argument relations. We then define a constraint system $C(F, R)$ whose solutions characterize the set of admissible trees compatible with the original YARN structure, thereby making underspecification explicit. For each admissible tree, we provide a compositional interpretation function that yields a modal first-order logic formula with temporality, grounding YARN structures in a logical semantics. We illustrate how the resulting framework captures non-trivial scope interactions and dynamic quantification phenomena on representative examples.

1. YARN

There appears to be a fine line between formalisms that are harder to produce (logic-based) and others that are harder to use for precise tasks (graph-based). Bridging the gap between logic and graphs for semantic annotation is the objective of YARN (Pavlova et al., 2024; Pavlova, 2025).

From graph-based formalisms, especially AMR, YARN adopts the absence of explicit variables in the graphical form of the formalism, a visual approach to annotation, and the use of a graph structure to represent the predicate structure of a sentence. From logical formalisms, YARN adopts the explicit representation of scope through tree-like structures, and the use of features to represent linguistic phenomena and interaction with the predicate part.

YARN adopts a layer-based approach to semantic representation, where there is no single "right" annotation for a sentence, but rather a set of possible annotations, depending on the features modeled. Layers (like tense, aspect, quantification, etc.) can be added or removed depending on the linguistic phenomena of interest. This allows for a flexible representation of meaning that can adapt to different tasks and requirements, as well as to annotator expertise.

YARN does not make assumptions about the underlying modeling of these phenomena and does not commit itself to one particular logical framework: in this sense, it is a "universal" formalism, like DRT.

We first present the formal definition of YARN. As a linguistically inspired framework, YARN encodes linguistic information that goes beyond semantics, providing ways to model information that is irrelevant to the main concern of this paper. We thus focus on a specific subset of elements of YARN structures, which we define below.

1.1. Formal Definition

Following (Pavlova, 2025), a YARN structure is defined as a nine-tuple:

$$\langle S, V, F, D, E, C, L, H, I \rangle$$

where:

- S is a set of vertices representing elementary events
- V is a set of vertices representing predicates and concepts
- F is a set of vertices representing features (tense, aspect, quantification, etc.)
- D is a set of directed edges between pairs of vertices $s_1, s_2 \in S$
- E is a set of directed edges between pairs of elements $v_1, v_2 \in V$
- C is a set of directed edges from a vertex $v \in V$ to a vertex $s \in S$
- L is a set of directed edges from a feature $f \in F$ to a vertex $v \in V$ or an edge $e \in E$
- H is a set of directed edges meeting one of two conditions:
  - start at an F vertex and end in an L or H edge
  - start at an L or H edge and end in a V vertex or an E edge

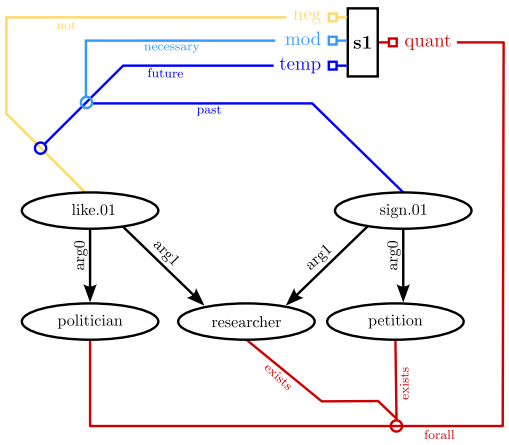


Figure 1: A YARN representation of sentence 1

- I is a set of edges between pairs of elements $v_1, v_2 \in V$

Intuitively, V, E give an AMR-like predicate-argument graph, and F, L, H build a scope-bearing feature structure that can be read as (a set of) operator trees over that graph. D, C specify a higher-level discourse structure. The V and E elements represent the basic predicate structure of a sentence, using PropBank frames for predicates in AMR fashion, and English words for concepts. S elements represent elementary events, linked through discourse relationships modeled by D edges, or linked to V nodes by C edges representing clauses (as in "He said he loved it"). F elements are features allowing the representation of linguistic phenomena like tense, aspect, and quantification. L and H elements allow features to modify predicative elements, with the iterative construction in H enabling scope representation (an H element represents a modification of the element represented by its target, inside the scope of the element represented by its source). I edges provide constraints on the interpretation of V and E elements by specifying set-membership relationships.

1.2. Example

To illustrate YARN's capabilities, let us annotate the sentence:

(1) In the future, politicians won't like researchers who signed a petition.

YARN structures are quite complex when decomposed and described in terms of their atomic elements, but they admit a more readable graphical representation, as shown by Figure 1 (which represents sentence 1).

In this representation, there is one main event taking place, represented by the S node s_1, at the top of the structure, which makes it the root. The predicative structure represents the core meaning of the sentence, consisting of V and E elements. It corresponds to the black graph structure of the figure. A reader trained in AMR will recognize the structure as very similar to the graphical representation of AMR. The predicative structure can be decomposed into V nodes and E edges, where V nodes represent concepts (e.g. "like.01"), and E edges represent relations (e.g. "arg0").

We now restrict the structure to express linguistic phenomena through features and their interactions with the predicative structure. This corresponds to the colored edges and nodes in Figure 1. For features, we show only temporality, negation, modality, and quantification layers. The set of available layers is larger and can be found in (Pavlova, 2025).

The L and H elements specify feature interactions. An L element directly links a feature to the element it modifies, and has a label expressing the kind of modification it performs. For instance, the L edge linking the quantification feature to the "politician" node expresses universal quantification over a variable ranging over politicians. Note that, even though we do not express it explicitly in the graphical representation, we associate a unique identifier to each node, allowing us to introduce a variable for each quantified element.

H elements work in the same way; however, they introduce scope. An H element either:

- links an existing L or H element to an element of the predicative part, such as the edge labelled "exists" in Figure 1.
- links a feature to an existing L or H element. This represents linguistic phenomena taking wide scope over other phenomena. Here, the chain $\xrightarrow{\Box}$ $\xrightarrow{\text{future}}$ $\xrightarrow{\text{not}}$ shows the order in which the modifications should be applied. This ordering represents an event that necessarily will not happen in the future, i.e. an impossible event. A different ordering of the same elements, starting with the "not" feature, represents an event that will not necessarily happen in the future, i.e. an event that might not happen.

As noted earlier, YARN structures can be complex, and not all elements are relevant for the purpose of this paper. We focus on a subset of YARN structures, restricting the features considered to those relevant for logical representation: tense, modality, negation, and quantification. Similarly, we do not treat I elements in this work: as they represent another type of relation between V nodes, we assume we can treat them as E elements. A more

precise treatment would require additional assumptions about the semantics we consider, which is out of the scope of this paper. Another feature of YARN that we do not treat in this paper is the possibility to have H edges target E edges, effectively quantifying over relations. We leave this to future work. The annotation layers we consider are sufficiently rich to represent a wide range of linguistic phenomena, but are not the only ones available in YARN. Crucially, we do not treat the aspect layer, which gives information about the internal temporal structure of events, however, the $\prec$ and $\mathcal{O}$ relations we introduce in the logical framework of Section 2 are sufficient to represent such semantic information (Kamp, 2017) , and thus the work presented in this paper can be extended to treat aspect as well. Finally, for simplicity, we only give a complete treatment of annotations with a single root, but as we will see in 3.5, our approach can be easily extended to structures with multiple roots. Even though we do not treat all features and phenomena that can be represented in YARN, the ones we do treat are sufficient to represent a wide range of linguistic phenomena and are the heart of the logical expressivity of YARN and its specification of quantification, which is the main concern of this paper.

2. Modal first order logic with temporality

We now introduce the logical framework we use as a proxy for the semantics of YARN structures.

We take into account modalities, temporality, negation and quantification features. We base ourselves on standard first order logic, adding modal operators $\Box$ and $\Diamond$ for necessity and possibility respectively, following the syntax presented in (Braüner and Ghilardi, 2007). Formal interpretation on modal subordination for semantics have been long studied (Davidson and Harman, 2012; Blackburn and Bos, 2003; Blackburn and Van Benthem, 2007; Qian et al., 2016).

The set of formulas we produce is defined as:

$$\phi ::= A \mid \top \mid \bot \mid p \mid \neg\phi \mid (\phi_1 \wedge \phi_2) \mid (\phi_1 \vee \phi_2) \mid \Diamond(\phi) \mid \Box(\phi) \mid \forall x\ \phi \mid \exists x\ \phi$$

With A the set of atomic formulas defined as:

$$t ::= x \mid c$$
$$A ::= R(t,t) \mid P(t)$$

Where R represents a binary relation symbol, P a predicate symbol, c a constant and x a variable.

In order to treat temporality, we add to the set of binary relations symbols two special temporal relations following the standard approach studied in (Kamp and Reyle, 1993), inspired by (Allen, 1983): $\prec$ for total temporal precedence and $\mathcal{O}$ for temporal overlap, with the following axioms:

- A1: $\forall x \forall y (x \prec y \rightarrow \neg(y\ \mathcal{O}\ x))$
- A2: $\forall x \forall y \forall z (x \prec y \wedge y \prec z \rightarrow x \prec z)$
- A3: $\forall x (x\ \mathcal{O}\ x)$
- A4: $\forall x \forall y (x\ \mathcal{O}\ y \leftrightarrow y\ \mathcal{O}\ x)$
- A5: $\forall x \forall y (x \prec y \rightarrow \neg(x\ \mathcal{O}\ y))$
- A6: $\forall x \forall y \forall z \forall t (x \prec y \wedge y\ \mathcal{O}\ z \wedge z \prec t \rightarrow x \prec t)$
- A7: $\forall x \forall y (x \prec y \vee x\ \mathcal{O}\ y \vee y \prec x)$

The variables in these axioms represents time points or intervals. In order to model the relation between events and the present moment, we also introduce a special constant symbol now, representing the present moment.

This way we can represent temporality in a way compatible with YARN structures, where temporality features are anchored to events.

We now give two examples of formulas representing every day sentences using this framework:

- "Maybe it will rain": $\Diamond(\exists e,\ now \prec e \wedge \text{rain}(e))$
- "John did not see any friend": $\neg\exists e, x, y, b\ e \prec now \wedge \text{see-01}(e) \wedge \text{john}(x) \wedge \text{person}(y) \wedge \text{befriend}(b) \wedge \text{arg0}(e,x) \wedge \text{arg1}(e,y) \wedge \text{arg0}(b,x) \wedge \text{arg1}(b,y)$

Note that we are using a Neo-Davidsonian (Davidson and Harman, 2012) framework, which comes naturally from the fact that in YARN, we modify each event with individual semantic phenomena.

3. Defining interpretations for YARN structures

3.1. Overview

We now turn to the main goal of this paper: obtaining interpretations as logical formulas for YARN structures. As is the case with our example 1 annotated with the structure presented in Figure 1, many YARN structures are ambiguous, as they do not encode all scopal specifications. We represent meaning as set of possible denotations, in line with (Poesio, 1994).

Our approach works as follows: let Y be a YARN structure. We transform Y into a pair (F, R), where F is a labelled forest (i.e. a set of labelled trees) and R is a set of relations over concept nodes identifiers. This allows us to separate two independent structures: scope dependencies (represented by F) and flat relations between entities (encoded by R). Now R and F give rise to a set of constraints $C(F, R)$ over the set $\mathbb{T}_{\text{all}}$ of trees with the same nodes as F.

We then obtain a set $\mathbb{T}$ of possible tree denotations as trees over nodes of F satisfying $C(F, R)$. Finally, we define an interpretation function, depending on R allowing us to convert every tree of $\mathbb{T}$ to a first-order logical formula with modalities.

For a given YARN structure Y, each element of the intermediate representation $\mathbb{T}$ unambiguously defines a formula. The set of constraints $C(F, R)$ represents both compatibility conditions for preserving scope indications that are already present in Y as well as coherence and linguistically motivated conditions that we shall explicit in 3.3.

3.2. YARN into the forest

Let Y be a YARN structure. We transform Y into a labelled forest F representing every element of the feature structure of Y.

First we introduce a unique identifier for each node of the predicative part (V elements), which will be used as variables in the logical formulas we produce. Here, for each node, we take the first letter of its label, disambiguating "politician and petition" by associating q with "petition" but any unique identifier would do.

This allows to represent the relation part R as follows: we simply extract the set of relations between concept nodes, which are represented by V elements in YARN. We represent them as a set of tuples (s, e_1, e_2) where s is the label of the relation (e.g. "arg0"), and e_1 and e_2 are the unique identifiers of the source and target nodes of the relation respectively. Here $R = \{\text{arg0}(l, p), \text{arg1}(l, r), \text{arg0}(s, r), \text{arg1}(s, q)\}$.

Now to build F we reifiy every H and L elements, representing them as nodes. An H or L element that has a V node as a target is represented as a node with label $Q_{\text{label}}\ e\ p$ where e is a variable representing the target node, label the label of the H or L element, and p the label of the target node. In that case we say that Q_{label} introduces the variable e. For instance, the L element linking the quantification feature to the "politician" node is represented as a node with label $Q_\forall\ p\ politicians$, which introduces variable p. An H element that has a H or L target is represented as a node with label Q_{label} where label is the label of the H or L element. For instance, the H element (blue in Figure 1) linking the "temporality" feature to the negation feature L element (yellow in Figure 1 is represented as a node with label Q_{future}.)

Finally, we maintain information about the source and target of H and L elements by representing an H element h_1 that has another H or L element h_2 as a source as a child node of the node representing h_2, and an H or L element h_1 that is the target of another H or L element h_2 as a child node of the node representing h_2. For instance, the

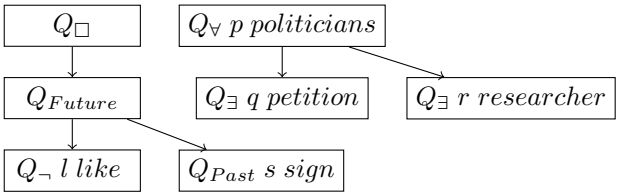


Figure 2: The forest F obtained from the YARN structure of Figure 1

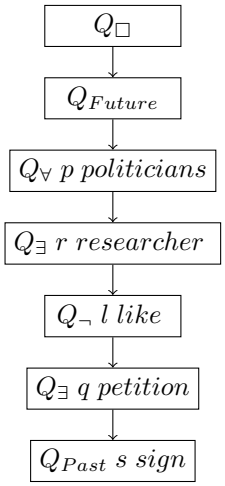


Figure 3: A possible merging for the forest represented in Figure 2

chain $\xrightarrow{\square}\ \xrightarrow{\text{future}}\ \xrightarrow{\text{not}}$ is represented as a tree with root node labelled $Q_\square$, which has a child node labelled Q_{future}, which itself has a child node labelled $Q_\neg\ l\ like$.

Note that we are conserving the scope information already present in the YARN structure, as the resulting forest is a direct encoding of the relations between H and L elements of the original structure. This achieves the separation of the two structures we mentioned in Section 2, with F representing scope information and R representing relations between entities. Figure 2 shows the forest F obtained from the YARN structure of Figure 1.

3.3. From one forest to many trees

Remember that F encodes general quantification and scope information, and R encodes relations. We now introduce a set of constraints $C(F, R)$ over the set $\mathbb{T}_{\text{all}}$ of trees with the same nodes as F. This allows to describe how we can unify the different trees in F to make a single tree, that we will be able to interpret as a formula. This step is necessary because the original YARN structure does not encode all scopal specifications, and thus we need to introduce constraints to obtain a set of possible

trees that are compatible with the original structure.

Let T be a tree in $\mathbb{T}_{\text{all}}$, the set of trees on $\mathbb{V}(F)$, the nodes of F. We say that T satisfies the constraints $C(F, R)$ if and only if the following conditions are satisfied:

- (Compatibility of scopes) For every pair of nodes (e_1, e_2) such that e_2 is a child of e_1 in F, e_2 is a descendant of e_1 in T.
- (Locality of features) In T, every child of a node that doesn't introduce a variable is a child of the same node in F.
- (Participant before event) For every relation $r(e_1, e_2)$ in R, if e_1 is introduced by a node n_1 and e_2 is introduced by a node n_2, n_2 is an ancestor of n_1 and every node that is on the path between n_2 and n_1 and that introduces a variable e_3 is such that there exists a relation $r'(e_3, e_2)$ or $r'(e_1, e_3)$ in R.

(Compatibility of scopes) condition plainly describes that we may not break the scope relations already present in F when building a tree T.

(Locality of features) condition describes that feature acting over other features do not modify wider scope features.

(Participant before event) condition is a linguistically motivated constraint to restrict the number of possible trees, and it may be relaxed. It says that we introduce events after their participants, and that we do not introduce any variable in the scope of the event's participants before introducing the event itself, unless it is another event relating the same participants.

$\mathbb{T}$ can be constructed by enumerating all possible trees that satisfy the (Compatibility of scopes) and (Locality of features) conditions. During the enumeration, at each step where we add a node, we check if its ancestors satisfy the constraints (Participant before event), and if it is not the case, we discard all the solutions that can be obtained from the current state. Figure 3 shows a possible tree obtained by merging the two trees of the forest F while respecting the constraints we just introduced.

3.4. Interpretation of trees over formulas

Having obtained the set $\mathbb{T}$ of trees over $\mathbb{V}(F)$ satisfying $C(F, R)$, we now define an interpretation function allowing us to convert every element T of $\mathbb{T}$ to a first order logical formula with modality.

Since T only represents the structure of a possible formula, we first need to enrich it with elements of R in order to inject the full semantic content of the structure. For each node of T, we compute the list of relations in R that are between variables introduced by this node or its ancestors. If this list is not empty, we append a child node that contains this list. Figure 4 shows the tree obtained after adding these relation nodes. We write the result of this operation Join(T, R).

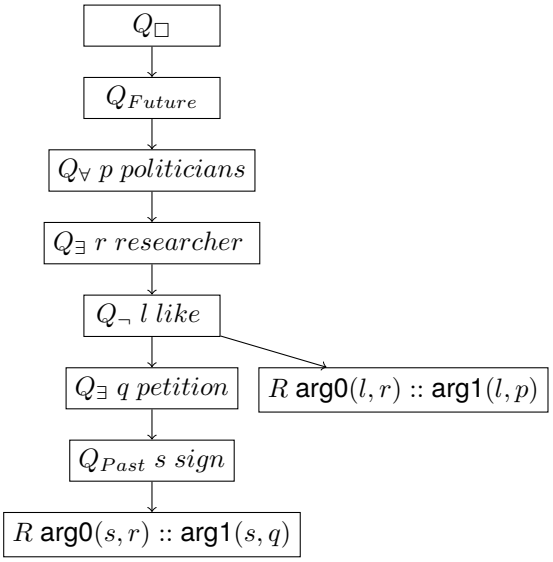


Figure 4: The merging of Figure 3 enriched with the relations in R

Then we can define an interpretation function $[\![\cdot]\!]_\tau$ that allows us to convert every element of $\mathbb{T}$ to a first order logical formula with modality. To achieve this, we need a linear representation of the trees, which will be the input of the function $[\![\cdot]\!]_\tau$. This representation is defined in Figure 5. **S** represents a tree, and each possible rule encodes the type of the root node. For example, the root node of the tree in Figure 4 is encoded by rule 5 ($Q_\square$ **F**). Each possible node type of **S** includes an **S**, which is a list of trees, to represent all child trees (except nodes created with rule 2, since they are always leaf nodes). **A** is a list of relations, which is used by nodes of type R (rule 2). The symbol :: serves as a separator between elements in both lists. The full translation of the tree in Figure 4 is Join$(T, R) =$
$(Q_\square\ (Q_{Future}\ (Q_\forall\ p\ politicians\ (Q_\exists\ r\ researcher$
$(Q_\neg\ l\ like\ (R$ arg0(l, r) :: arg1(l, p) :: $[])$
$(Q_\exists\ q\ petition$
$(Q_{Past}\ s\ sign\ (R$ arg0(s, r) :: arg1(s, q) :: $[])$
:: $[])$ :: $[])$ :: $[])$ :: $[])$ :: $[])$ :: $[])$ :: $[])$

The function $[\![\cdot]\!]_\tau$ is then defined for each rule of the tree grammar. The full definition is given in Figure 6. τ represents the temporal context in which the formula is evaluated, and it is used to interpret temporal operators. This provides a unique interpretation of every tree in $\mathbb{T}$, and thus for the original YARN structure Y given the solution to the constraints $C(F, R)$ provided by $\mathbb{T}$, by computing $[\![$ Join(T,R) $]\!]_{now}$.

The result for the tree of Figure 4

$$
\begin{aligned}
\mathbf{S} &::= (R\ \mathbf{A}) && (2)\\
&\mid (Q_{\forall}\ e\ p\ \mathbf{F}) && (3)\\
&\mid (Q_{\exists}\ e\ p\ \mathbf{F}) && (4)\\
&\mid (Q_{\Box}\ \mathbf{F}) && (5)\\
&\mid (Q_{\Box}\ e\ p\ \mathbf{F}) && (6)\\
&\mid (Q_{\Diamond}\ \mathbf{F}) && (7)\\
&\mid (Q_{\Diamond}\ e\ p\ \mathbf{F}) && (8)\\
&\mid (Q_{\neg}\ \mathbf{F}) && (9)\\
&\mid (Q_{\neg}\ e\ p\ \mathbf{F}) && (10)\\
&\mid (Q_{Past}\ \mathbf{F}) && (11)\\
&\mid (Q_{Past}\ e\ p\ \mathbf{F}) && (12)\\
&\mid (Q_{Present}\ \mathbf{F}) && (13)\\
&\mid (Q_{Present}\ e\ p\ \mathbf{F}) && (14)\\
&\mid (Q_{Future}\ \mathbf{F}) && (15)\\
&\mid (Q_{Future}\ e\ p\ \mathbf{F}) && (16)\\
\mathbf{F} &::= [\ ]_F && (17)\\
&\mid \mathbf{S} :: \mathbf{F} && (18)\\
\mathbf{A} &::= [\ ]_A && (19)\\
&\mid (s, e, f) :: \mathbf{A} && (20)
\end{aligned}
$$

In all rules where they appear, e and f are variable names, p is a predicate, and s is a relation.

Figure 5: The grammar which defines the linear representation of trees in $\mathbb{T}$

$$
\begin{aligned}
[\![\ [\ ]_A\ ]\!]_\tau &\rightarrow \top && (21)\\
[\![\ (s, e_1, e_2) :: al\ ]\!]_\tau &\rightarrow s(e_1, e_2) \wedge [\![\ al\ ]\!]_\tau && (22)\\
[\![\ [\ ]_F\ ]\!]_\tau &\rightarrow \top && (23)\\
[\![\ t :: f\ ]\!]_\tau &\rightarrow [\![\ t\ ]\!]_\tau \wedge [\![\ f\ ]\!]_\tau && (24)\\
[\![\ R\ al\ ]\!]_\tau &\rightarrow [\![\ al\ ]\!] && (25)\\
[\![\ Q_{\forall}\ e\ p\ f\ ]\!]_\tau &\rightarrow \forall e, p(e) \Rightarrow [\![\ f\ ]\!]_\tau && (26)\\
[\![\ Q_{\exists}\ e\ p\ f\ ]\!]_\tau &\rightarrow \exists e, p(e) \wedge [\![\ f\ ]\!]_\tau && (27)\\
[\![\ Q_{\Box} f\ ]\!]_\tau &\rightarrow \Box([\![\ f\ ]\!]_\tau) && (28)\\
[\![\ Q_{\Box}\ e\ p\ f\ ]\!]_\tau &\rightarrow \Box(\exists e, p(e) \wedge [\![\ f\ ]\!]_\tau) && (29)\\
[\![\ Q_{\Diamond}\ f\ ]\!]_\tau &\rightarrow \Diamond([\![\ f\ ]\!]_\tau) && (30)\\
[\![\ Q_{\Diamond}\ e\ p\ f\ ]\!]_\tau &\rightarrow \Diamond(\exists v, p(v) \wedge [\![\ f\ ]\!]_\tau) && (31)\\
[\![\ Q_{\neg} f\ ]\!]_\tau &\rightarrow \neg([\![\ f\ ]\!]_\tau) && (32)\\
[\![\ Q_{\neg}\ e\ p\ f\ ]\!]_\tau &\rightarrow \neg(\exists e, p(e) \wedge [\![\ f\ ]\!]_\tau) && (33)\\
[\![\ Q_{Past}\ f\ ]\!]_\tau &\rightarrow \exists e, e \prec \tau\ \wedge [\![\ f\ ]\!]_e && (34)\\
[\![\ Q_{Past}\ e\ p\ f\ ]\!]_\tau &\rightarrow \exists e, p(e) \wedge e \prec \tau\ \wedge [\![\ f\ ]\!]_e && (35)\\
[\![\ Q_{Present}\ f\ ]\!]_\tau &\rightarrow \exists e, \tau\ \mathcal{O}\ e\ \wedge [\![\ f\ ]\!]_e && (36)\\
[\![\ Q_{Present}\ e\ p\ f\ ]\!]_\tau &\rightarrow \exists e, p(e) \wedge \tau\ \mathcal{O}\ e\ \wedge [\![\ f\ ]\!]_e && (37)\\
[\![\ Q_{Future}\ f\ ]\!]_\tau &\rightarrow \exists e, \tau \prec e \wedge [\![\ f\ ]\!]_e && (38)\\
[\![\ Q_{Future}\ e\ p\ f\ ]\!]_\tau &\rightarrow \exists e, p(e) \wedge \tau \prec e \wedge [\![\ f\ ]\!]_e && (39)
\end{aligned}
$$

In rules 34, 36 and 38, e is a fresh variable name.

Figure 6: The definition of function $[\![\ \cdot\ ]\!]_\tau$ for every rule of the tree grammar

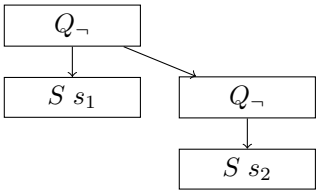


Figure 7: A consequence relation between two S nodes modelled using $Q_\neg$ nodes

is $[\![$ Join(T,R) $]\!]_{now} = \Box(\exists e, now \prec e \wedge (\forall p, politician(p) \Rightarrow (\exists r, researcher(r) \wedge (\neg(\exists l, like(l) \wedge (\exists q, petition(q) \wedge (\exists s, sign(s) \wedge e \prec s \wedge$ arg0$(s, r) \wedge$ arg1$(s, q)) \wedge$ arg0$(l, r) \wedge$ arg1$(l, p)))))))$

The other possible tree gives rise to the reading where the "petition" q variable is introduced before the "researcher" r variable, which corresponds to the reading where there is a specific petition whose signing makes politicians dislike researchers, which is a weaker but valid reading of the original sentence. We now follow the approach of (Bos, 1996) and define a semantic for YARN by defining the interpretation of Y as the set of all pairs consisting of a tree T in $\mathbb{T}$ and the truth value of $[\![$ Join(T, R) $]\!]_{now}$.

In plain words, we interpret a YARN structure as a set of possible interpretations together with their truth values.

3.5. Extension to complex structures

In this section, we show how to extend the approach we just described to YARN structures involving several S nodes, linked together by elements of type D. This is important, as many YARN structures involve several S nodes, and we want to be able to obtain interpretations for such structures as well. Remember that S nodes represent main events, and that D represent discourse relations. Different YARN substructures may share the same V nodes, thus we can't treat them as independent structures, and we need to take into account the relations between them when building the forest F and the set of relations R. We will show how to treat the "consequence" relation, which we model as implication.

We extend the forest building step of 3.2: we first build a forest for each S node independently, and then add a new node labeled $Svar$ for each S node, which introduces a variable representing the event corresponding to this S node. Writing $a \Rightarrow b$ as $\neg(a \wedge \neg b)$, we add two $Q_\neg$ nodes, in the configuration shown in Figure 7. Note that if we wished to consider that the consequence relation induces a temporal precedence phenomenon, we could have done so by the mean of an additional

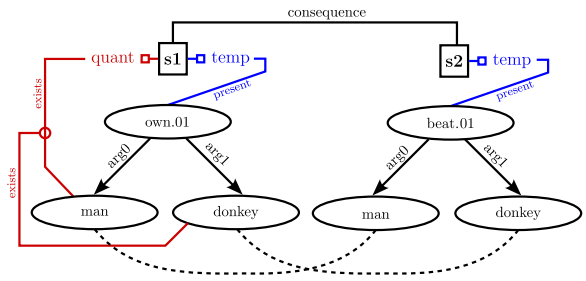


Figure 8: A possible YARN structure for the sentence "Every farmer who owns a donkey beats it"

Q_{Future} node.

We can apply the same steps as before to obtain a set of trees $\mathbb{T}$, and then apply the interpretation function to obtain a set of formulas. We only need to update the grammar 5 to allow for the new elements we introduce, and provide a suitable interpretation for them: $\mathbf{S} ::= (S\ e\ \mathbf{F})$ and $[\![S\ e\ f]\!]_\tau \rightarrow [\![f]\!]_e$. As an example of the expressivity of this system, we show how this allows us to obtain a correct interpretation for donkey style structures. Such constructions are well known to pose profound challenges related to compositionality, quantification and coreference (Kamp, 1981; Partee et al., 1984; Kanazawa, 1994). Since YARN does not encode a coreference resolution mechanism, the primary aim of this example is to demonstrate that dynamic quantification phenomena are naturally handled by our approach, yielding correct interpretations for such sentences without requiring any additional machinery.

Consider the following sentence:

(40) If a man owns a donkey, he beats it.

We propose a YARN annotation for this sentence, which is shown in Figure 8. For readability purposes, we have duplicated the nodes corresponding to "man" and "donkey", but there is only one occurrence of each node in the actual structure. This is represented by the dashed lines in the bottom of Figure 8.

Applying the approach presented in 3.2, we obtain a forest F, represented graphically in Figure 9. As for relations, we have $R = \{\text{arg0}(o,m), \text{arg1}(o,d), \text{arg0}(b,m), \text{arg1}(b,d)\}$, where we introduced the variables m and d to represent the "man" and "donkey" nodes respectively, and o and b for the "own" and "beat" nodes respectively.

Then, after applying the approach of 3.3, we obtain a set of trees $\mathbb{T}$, which is reduced to a single tree T in this particular case. This tree is represented graphically in Figure 10.

Finally, using the rewriting system of Subsection 3.4, we obtain the following formula: $[\![\text{ Join}(T,R)]\!]_{now} = \neg\exists x,y,o\Big(man(x) \wedge donkey(y) \wedge own(o) \wedge o\ \mathcal{O}\ now \wedge \text{arg0}(o,x) \wedge \text{arg1}(o,y) \wedge \neg\exists b\big(beat(b) \wedge b\ \mathcal{O}\ o \wedge \text{arg0}(b,x) \wedge \text{arg1}(b,y)\big)\Big)$ which says "There is no man and no donkey such that that man owns that donkey and does not beat it", which is a correct interpretation for the sentence 40.

Using the equivalences $\neg\exists x, P(x) \equiv \forall x, \neg P(x)$ and $\neg(A \wedge \neg B) \equiv A \Rightarrow B$, we can rewrite this formula in a more natural form as follows: $\forall x,y,o\Big(man(x) \wedge donkey(y) \wedge own(o) \wedge o\ \mathcal{O}\ now \wedge \text{arg0}(o,x) \wedge \text{arg1}(o,y) \Rightarrow \exists b\big(beat(b) \wedge b\ \mathcal{O}\ o \wedge \text{arg0}(b,x) \wedge \text{arg1}(b,y)\big)\Big)$

Conclusion

YARN is a linguistic formalism grounded in logic, both in its underlying motivations and in its formal design. We have shown that YARN structures admit a systematic translation into logical formulas. This suggests that YARN can be regarded as a semantic representation in the strict sense, in so far as truth conditions can be extracted from its structures.

While the presence of structural ambiguity may appear problematic at first, it can also be viewed as an expressive feature, allowing for the representation of complex quantificational patterns, by describing a set of possible interpretations. This is a natural way to account for the ambiguity of natural language, and it allows to represent complex scopal interaction, however, it is not the only solution, and one may argue that the absence of scope specification between two phenomena is not always a sign of ambiguity, but rather a sign of independence between the two phenomena. In that case, we could consider that a structure denotes a single formula, and that the absence of scope specification between two phenomena is a sign of independence between them, which can be modeled by other logical means (Hintikka, 1979), such as the use of Henkin quantifiers (Henkin and Karp, 1965), or Independance Friendly Logic (Hintikka and Sandu, 1989). This would allow to explore the use of YARN as an interlingua for semantic representation, which offer interesting perspectives for symbolic NLP applications, notably due to its visual characteristics.

The scope constraint system and the conversion algorithm presented here are closely related to previous work, in particular that of (Bos, 1996), but our approach differs in its algorithmic nature, the explicit separation of scope and relation structures, and

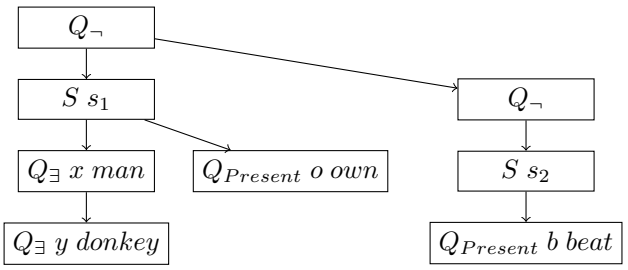


Figure 9: The forest obtained from the YARN structure of Figure 8

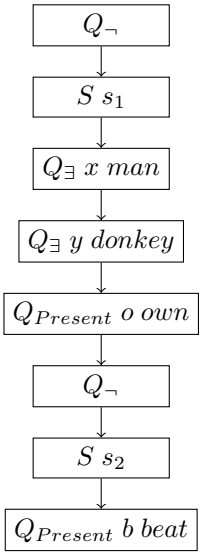


Figure 10: The only possible tree obtained by merging the forest in Figure 9

our commitment to a specific logical framework.

We also introduce a linguistically motivated constraint to restrict the number of possible trees. Note that we are only assuming logical interpretations for the action of features at the very last step: as such, providing alternative interpretations doesn't require any change in the previous steps. For instance, evolving our interpretation for trees by adding a context, as is done in (De Groote, 2006), would only require a change in the definition of $[\![\cdot]\!]_\tau$. This would allow to give a compositional interpretation of YARN structure linked by discourse relation, without needing to merge them into a single structure, assuming they do not share nodes.

Ethical considerations

This study raises no specific ethical concerns beyond those inherent to semantic representation itself. The primary limitations relate to ontological bias in meaning representations, largely attributable to the disproportionate prevalence of English in available linguistic resources and work.

Acknowledgements

We would like to thank Amandine Decker for her attentive and considerate help in proofreading the document. We also thank the reviewers for their suggestions and for helping us broaden our knowledge of previous work on these subjects.

4. Bibliographical References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26:832–843.

Maxime Amblard and Christian Retoré. 2014. Partially Commutative Linear Logic and Lambek Calculus with Product: Natural Deduction, Normalisation, Subformula Property. *IfColog Journal of Logics and their Applications (FLAP)*, 1(1):53–94.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking.

Patrick Blackburn and Johan Bos. 2003. Computational semantics. *Theoria: An International Journal for Theory, History and Foundations of Science*, pages 27–45.

Patrick Blackburn and Johan Van Benthem. 2007. 1 modal logic: a semantic perspective. In *Studies in logic and practical reasoning*, volume 3, pages 1–84. Elsevier.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E. L. Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad

infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italia. ELRA and ICCL.

Johan Bos. 1996. Predicate logic unplugged.

Johan Bos. 2016. Squib: Expressive power of Abstract Meaning Representations. *Computational Linguistics*, 42(3):527–535.

Torben Braüner and Silvio Ghilardi. 2007. 9 first-order modal logic. In Patrick Blackburn, Johan Van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 549–620. Elsevier.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

Richard Crouch and Aikaterini-Lida Kalouli. 2018. Named graphs for semantic representation. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 113–118, New Orleans, Louisiana. Association for Computational Linguistics.

Donald Davidson and Gilbert Harman. 2012. *Semantics of natural language*, volume 40. Springer Science & Business Media.

Philippe De Groote. 2006. Towards a montagovian account of dynamics. In *Semantics and linguistic theory*, pages 1–16.

Gottlob Frege. 1891. Function and concept. In Peter Geach and Max Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*, pages 21–41. Basil Blackwell, Oxford. Original work published 1891.

Gottlob Frege. 1893. *Grundgesetze der Arithmetik*, volume 1–2. Hermann Pohle, Jena.

Leon Henkin and Carol R. Karp. 1965. Some remarks on infinitely long formulas. *Journal of Symbolic Logic*, 30(1):96–97.

Jaakko Hintikka. 1979. Quantifiers vs. quantification theory. In *Game-Theoretical Semantics: Essays on Semantics by Hintikka, Carlson, Peacocke, Rantala, and Saarinen*, pages 49–79. Springer.

Jaakko Hintikka and Gabriel Sandu. 1989. Informational independence as a semantical phenomenon. In Jens Erik Fenstad, Ivan T. Frolov, and Risto Hilpinen, editors, *Logic, Methodology and Philosophy of Science VIII*, volume 126 of *Studies in Logic and the Foundations of Mathematics*, pages 571–589. Elsevier.

Hans Kamp. 1981. A theory of truth and semantic representation. In Jeroen Groenendijk, Theo Janssen, and Martin Stokhof, editors, *Formal Methods in the Study of Language*, volume 135 of *Mathematical Centre Tracts*, pages 277–322. Mathematical Centre, Amsterdam.

Hans Kamp. 2017. Events, discourse representations and temporal reference. *Semantics and Pragmatics*, 10:2–1.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, volume 42. Springer Science & Business Media.

Makoto Kanazawa. 1994. Weak vs. strong readings of donkey sentences and monotonicity inference in a dynamic setting. *Linguistics and philosophy*, 17(2):109–158.

Kenneth Lai, Lucia Donatelli, and James Pustejovsky. 2020. A continuation semantics for Abstract Meaning Representation. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 1–12, Barcelona Spain (online). Association for Computational Linguistics.

Paula Reyero Lobo, Enrico Daga, Harith Alani, and Miriam Fernandez. 2023. Semantic web technologies and bias in artificial intelligence: A systematic literature review. *Semantic Web*, 14(4):745–770.

Jirka Maršík, Maxime Amblard, and Philippe de Groote. 2021. Introducing (| λ |), a λ-calculus for effectful computation. *Theoretical Computer Science*, 869:108–155.

Giang Nguyen, Ivan Brugere, Shubham Sharma, Sanjay Kariyappa, Anh Totti Nguyen, and Freddy Lecue. 2025. Interpretable LLM-based table question answering. *Transactions on Machine Learning Research*.

Barbara H. Partee, Alice ter Meulen, and Robert E. Wall. 1984. *Mathematical Methods in Linguistics*. D. Reidel Publishing Company, Dordrecht.

Siyana Pavlova. 2025. *Tools and methods for semantically annotated corpora*. Ph.D. thesis, Université de Lorraine.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2023a. Bridging Semantic Frameworks: mapping DRS onto AMR. In *Proceedings of The*

15th International Conference on Computational Semantics (IWCS 2023), Nancy, France.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2023b. Structural and global features for comparing semantic representation formalisms. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 1–12, Nancy, France. Association for Computational Linguistics.

Siyana Pavlova, Maxime Amblard, and Bruno Guillaume. 2024. YARN is All You Knit: Encoding Multiple Semantic Phenomena with Layers. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 66–76, Torino, Italia. ELRA and ICCL.

Massimo Poesio. 1994. *Discourse interpretation and the scope of operators*. Ph.D. thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Dernière mise à jour - 2023-07-26.

Sai Qian, Philippe de Groote, and Maxime Amblard. 2016. Modal Subordination in Type Theoretic Dynamic Logic. *Linguistic Issues in Language Technology*, 14((1)):1–39.

Shira Wein. 2025. Can uniform meaning representation help GPT-4 translate from indigenous languages? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 278–285, Vienna, Austria. Association for Computational Linguistics.

Shira Wein and Juri Opitz. 2024. A survey of amr applications. In *EMNLP*, pages 6856–6875.

Meaning Representations as Variational Quantum Circuits

Tilen G. Limbäck-Stokin, Tanishka A. Birdavade, Kin Ian Lo, Mehrnoosh Sadrzadeh
Quantum Learning Labs, University College London
London, United Kingdom
{tilen.limback-stokin.21, tanishka.birdavade.22, kin.lo.20, m.sadrzadeh}@ucl.ac.uk

Abstract

Large language and vision-language models (VLMs) struggle with a 'compositionality gap'. They treat language as a sequence of tokens lacking any structure and thus rely on a large number of parameters, making them computationally expensive. To address these issues, we propose **CCG-VQC**, a quantum framework that unifies statistical distributions with linguistic structure. Guided by Combinatory Categorial Grammar, our model maps syntactic rules into parametrised quantum circuits and models sentences as quantum states. We evaluate **CCG-VQC** on structural VLM benchmarks such as ARO and SVO-Swap. Our experiments show that **CCG-VQC** consistently outperforms a quantum bag-of-words model, as well as classical VLMs such as CLIP and OpenCLIP. **CCG-VQC** achieved 71.19% accuracy on ARO-Attribution, significantly outperforming the parameter-matched MicroCLIP, which struggled to surpass random chance with a maximum performance of 50.85%.

Keywords: Quantum Machine Learning, Syntax, Semantics, Vision-Language Models

1. Introduction

Large language models are thought to be one of the most successful learning algorithms of our age. They are applied to a wide range of domains, from weather forecasting (Li et al., 2025a) and medical modelling (Singhal et al., 2025) to theorem proving (Hubert et al., 2025) and scientific calculations in physics (Pan et al., 2025) and chemistry (Boiko et al., 2023). Although LLMs learn structural patterns present in language implicitly, they treat text as a series of tokens lacking explicit, controllable syntactic and semantic structure (Bender and Koller, 2020). Moreover, this is computed through self-attention, based on the statistics of co-occurrence, which is computationally intensive (Vaswani et al., 2023). This does not take full advantage of the generalisation that arises from the compositional nature of a grammatical rule set (Lake and Baroni, 2018).

Despite achieving high accuracies in generative tasks, they exhibit poor reliability and tend to hallucinate. Their success relies on a large number of parameters, making them energy inefficient and expensive. Moreover, it is not clearly known if LLMs are actually distilling their knowledge to generalise by learning fundamental rules governing the data, such as composition, or if they are just memorising at an immense scale, leading to a problem dubbed as the "compositionality gap" (Press et al., 2023; Li et al., 2025b; Ni et al., 2024).

The behaviour of LLMs is orthogonal to formal computational linguistic models that treat language as sequences of words generated according to the rules of grammar and abiding by structural, semantic and pragmatic constraints (Blackburn and Bos, 2005; Morrill, 2010; Steedman, 2000). The Achilles heel of these methods is their rigidity: they tend to only handle hand-picked examples and are inapplicable to large-scale, naturally occurring data. It is hard to fully describe, let alone formalise, all the rules of the grammar of a language. Formalising and reasoning about semantic and pragmatic constraints remains an open challenge.

A middle ground can be reached by developing meaning representations that unify statistical distributions of data with the linguistic structures embedded in it. Herein, semantic symbolic representations are assigned to syntactic structures via a homomorphic map. The symbolic representations are learnt in context, e.g. via labelled datasets or co-occurrence in corpora of text. Whereas an LLM uses statistical methods and learns vectors for tokens, the unified models learn higher-order linear maps informed by syntactic structure. For instance, the symbolic representations of a transitive sentence are a multilinear map modelling the predicative meaning of the verb, which then applies to the meaning representations of its subject and object, which are vectors.

In finite dimensions, learning a multilinear map is equivalent to learning a tensor. This amounts to training a multi-dimensional array, which suffers from exponential scaling. In quantum computation, tensors are treated as states of quantum systems and are efficiently modelled using variational quantum circuits (VQCs), represented by a handful of parametrised quantum operations known as "gates". This is particularly advantageous for grammar-based learning, where explicit tensor representations grow exponentially with sentence complexity. VQCs mitigate this parameter explosion by accurately approximating these large tensors using unitary rotations, each requiring only

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 113–123
11 May 2026.

a single trainable parameter. Moreover, as quantum computers become larger and more noise-resistant, there is potential to run these models on actual hardware. This natively executes the required linear algebra—since qubits inherently reside in a tensor (Hilbert) space—thereby circumventing the memory bottlenecks of traditional classical computation. In this paper, we present **CCG-VQC**: a meaning representation for natural language using VQCs. Our representations are guided by the rules of Combinatory Categorial Grammar (CCG); we develop a homomorphic map that turns each rule into a series of quantum gates. The meaning of a sentence in this setting is a quantum state. Semantic similarity of two sentences is modelled by *fidelity*, a quantum information theoretic measure for the overlap between two quantum states.

The capabilities of **CCG-VQC** are showcased on a task from vision-language (VLMs). It has been shown that VLMs lack semantic understanding, as a result, struggle to align their embeddings in structural tasks (Koishigarina et al., 2025; Hendricks and Nematzadeh, 2021; Yuksekgonul et al., 2023; Lewis et al., 2024). A variety of datasets are developed to probe for this challenge, working with cases where the correct and the incorrect texts and images are structural renditions of each other. We evaluate our model on two such benchmarks, ARO (Yuksekgonul et al., 2023) and SVO-Swap (Lo et al., 2025), inspired by SVO-Probes (Hendricks and Nematzadeh, 2021).

The performance of our model is compared with a quantum counterpart of a bag-of-words model (QBoW). We also compare our results to CLIP, which is OpenAI's original VLM (Radford et al., 2021) and an open-sourced version of it called OpenCLIP (Ilharco et al., 2021). Our experiments show that firstly, **CCG-VQC** works better than QBoW, and secondly, **CCG-VQC** outperforms both CLIP and OpenCLIP.

To facilitate a fair comparison, we trained a compact transformer model, which we term Micro-CLIP, matching the parameter count of our **CCG-VQC**. Our experiments revealed that this reduced-scale transformer achieves only near-random performance on the ARO benchmarks, whereas **CCG-VQC**, which operates at that parameter scale by design, achieves significantly higher accuracy. This demonstrates that our structurally-informed model is substantially more parameter-efficient than standard attention-based architectures.

Existing Work and Novel Contributions Unified models of statistics and structure exist and are referred to as Compositional Distributional Semantics, sometimes dubbed as DisCoCat (Coecke et al., 2010, 2013; Maillard et al., 2014; Wijnholds et al., 2020; Grefenstette and Sadrzadeh, 2015). DisCoCat learns words very similar to our approach, but it is based on pregroup grammars (Lambek, 1999), which are not widely used in the community. There also exists a translation between DisCoCat and VQCs (Yeung and Kartsaklis, 2021; Lorenz et al., 2021; Wazni and Sadrzadeh, 2023; Kartsaklis et al., 2021; Wazni et al., 2024), but it relies on the theory of categories and has not been tested on large scale structured benchmarks or compared with state-of-the-art technology. A tensor network semantics was defined for CCG and evaluated on VLM tasks (Lo et al., 2025). VQCs have a significantly lower parameter count than tensor networks. Furthermore, tensor networks remain classical objects. Their output is a vector that can be inputted in the `InfoNCE` learning objective of CLIP-like architectures. The output of a VQC is a quantum state, and we design the new `QInfoNCE` objective function for contractive learning.

2. From Syntax to Quantum Semantics

The textual pipeline makes use of grammar as a structural blueprint for designing quantum circuits. Input sentences are processed according to the rules of a CCG. We have chosen to use CCG here as it is a type-driven logic with functional application rules, making it particularly compatible with tensor algebra. We can simply assign a tensor space to each atomic type and treat function applications as contractions. This will become evident in the following explanation from CCG to quantum circuits. Throughout this work we use the Bobcat parser to tag the sentences and generate the parse trees (Clark, 2021). A set of rules, shown in Figure 1, translates the below described CCG trees into quantum circuits, mapping atomic types to qubits, words to variational quantum circuits, and compositions to Bell measurement with post-selection to the Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$. Specifically, each word in the sentence is represented as a parametrised quantum state, referred to as a variational quantum circuit, which is optimised during training to produce the final textual quantum state $|\psi_{\text{txt}}\rangle$.

2.1. Rules of Grammar

CCG consists of a basic and an advanced set of types and inference rules. The basic CCG has the set of atomic types $\{N, NP, S\}$ representing nouns, noun phrases, and sentences, and two function types: forward and backward application. The function type $A\backslash B$ outputs type A given B is to the left and A/B outputs A given B is on the right. The formal definitions of these rules are below.

$$\frac{X/Y \quad Y}{X}\,{<} \qquad \frac{Y \quad X\backslash Y}{X}\,{>}$$

Consider the following example sentence "*Alice likes Bob*". The syntactic structure of this sentence is derivable using both of the inference rules, see below.

$$\dfrac{\dfrac{\text{Alice}}{NP} \quad \dfrac{\dfrac{\text{likes}}{(S\backslash NP)/NP} \quad \dfrac{\text{Bob}}{NP}}{S\backslash NP}\,{<}}{S}\,{>}$$

Basic CCG has other rules such as type raising T and composition B, each with forward and backward variants. The formal definitions of these rules are given below.

$$\frac{X/Y \quad Y/Z}{X/Z}\,{<}B \qquad \frac{Y\backslash Z \quad X\backslash Y}{X\backslash Z}\,{>}B$$

$$\frac{X}{T/(T\backslash X)}\,{<}T \qquad \frac{X}{T\backslash(T/X)}\,{>}T$$

Composition is used for phrase formation to allow for non-standard constituents, such as those found in complex auxiliary structures. As an example, consider the derivation below.

$$\dfrac{\dfrac{\text{Alice}}{NP} \quad \dfrac{\dfrac{\text{might}}{(S\backslash NP)/VP} \quad \dfrac{\dfrac{\text{like}}{VP/NP} \quad \dfrac{\text{Bob}}{NP}}{(S\backslash NP)/NP}\,{>}\mathbf{B}}{S\backslash NP}\,{<}}{S}\,{>}$$

Type raising allows for non-local dependencies such as those seen in relative clause formation. For instance, see the derivation below.

$$\dfrac{\dfrac{\dfrac{\text{The}}{NP/N} \quad \dfrac{\text{food}}{N}}{NP}\,{>} \quad \dfrac{\text{which}}{(NP\backslash NP)/(S/NP)} \quad \dfrac{\dfrac{\dfrac{\text{Mary}}{NP}}{S/(S\backslash NP)}\,{>}\mathbf{T} \quad \dfrac{\text{bought}}{(S\backslash NP)/NP}}{S/NP}\,{>}\mathbf{B}}{NP\backslash NP}\,{>}}{NP}\,{<}$$

In the above, the subscripts $_>$ and $_<$ let us know that an inference rule is forward or backward applied, respectively. Basic CCG has an expressive power equivalent to context-free grammars (Steedman and Baldridge, 2011). CCG can be endowed with a set of additional rules to increase its expressivity. For example, there are rules for type raising T, coordination Φ, and cross composition $B_\times$. Coordination is used for conjunctives such as "and, or, because". Cross-composition is applied to complex situations such as the treatment of phrasal verbs. For further details we refer the reader to a CCG reference, e.g. see (Steedman, 2000).

2.2. Quantum Computing Foundations

While we provide a brief overview of quantum principles here to establish notation, we recognise that the following summary may be challenging for those new to the field. For a complete pedagogical introduction, we refer the reader to the standard text by Nielsen and Chuang (Nielsen and Chuang, 2000).

The fundamental unit of information is the qubit, a state vector $|\psi\rangle$ in a 2-dimensional complex Hilbert space $\mathbb{C}^2$. The states $|0\rangle$ and $|1\rangle$ form an orthonormal basis for this space, known as the computational basis. Unlike classical bits, a qubit can exist in a superposition state defined below:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}$$

The following normalisation constraint holds for a qubit:

$$\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$$

For a system of n qubits, the corresponding state space is the following tensor product of dimension 2^n:

$$\mathcal{H} = \bigotimes_{i=1}^{n} \mathbb{C}^2.$$

The evolution of these states is governed by operators U that are referred to as *unitary*, which belong to the symmetry group $U \in SU(2^n)$. To extract information from the quantum system, we perform a *measurement*. In the computational basis, measuring a qubit collapses the state to $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$.

The above is the Hilbert Space formalism of quantum computing. In practice, it is advantageous to work with a circuit model, which provides a direct path for running the circuits on quantum computers. In the circuit model of computation, these operations are decomposed into a sequence of quantum gates. Single-qubit gates, such as the Hadamard (H) gate, are used to generate superposition:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1}$$

while rotation gates (R_x, R_z) shift the qubit's state by a specific angle θ:

$$R_z(\theta) = \exp\left(-i\frac{\theta}{2}Z\right) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{2}$$

Multi-qubit gates, such as the CNOT, are crucial for creating entanglement between qubits, which is necessary to explore the exponentially large state space, where the state of one qubit cannot be described independently of the others. This entanglement is what allows quantum circuits to model the interactions between words and phrases in a sentence.

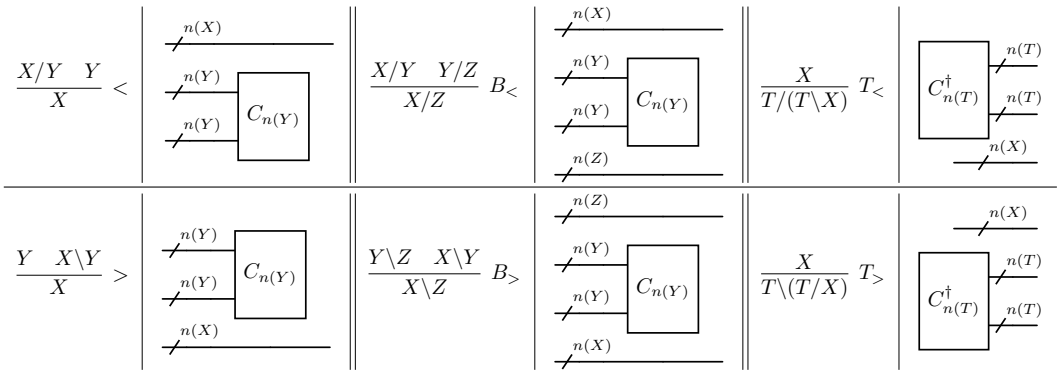


Figure 1: Mapping from the CCG rules to VQCs. The C gates, defined in Figure 2, are either performing Bell measurements mirroring predicate-application, or initialisation Bell states mirroring raising types.

Traditionally, quantum circuits are read from left to right. Each horizontal line represents a qubit (or a bundle of n qubits), and boxes represent gates. A vertical line connecting a filled circle ($\bullet$) to a plus symbol ($\oplus$) denotes a CNOT gate, where the bit of the target qubit is flipped only if the control qubit is in state $|1\rangle$. Triangles at the start of a wire ($|0\rangle$) indicate state preparation, while triangles at the end ($\langle 0|$) represent measurement and post-selection on the 0 outcome.

2.3. Variational Quantum Circuits for CCG Derivations

The conversion from a CCG derivation to a quantum circuit begins with a map n which assigns to each atomic type A a desired number of qubits $n(A)$. The choice of n for the atomic types is a hyperparameter of the model, which can be tuned to balance computational cost and model expressivity. The number of qubits assigned to a composite type X/Y or $X\backslash Y$ is recursively defined as

$$n(X/Y) = n(X\backslash Y) = n(X) + n(Y) \qquad (3)$$

The meaning of a word with type T is represented by an $n(T)$-qubit quantum state $|\psi\rangle$, prepared by a variational quantum circuit $U_{n(T)}(\Theta)$ acting on an initial $|0\rangle^{\otimes n}$ state. These circuits depend on a list of trainable parameters Θ, typically rotation angles. While an arbitrary n-qubit state requires $\mathcal{O}(2^n)$ parameters to specify, VQCs utilise an *ansatz* to explore the high-dimensional Hilbert space using only a polynomial number of parameters.

An *ansatz* is a parametrised circuit template designed to balance expressivity with trainability using a small set of rotation gates and entangling gates. The generalised version for the sentence 'Alice likes Bob' can be seen in Figure 2(a). In our diagrams, wires in parallel represent a tensor product of qubit spaces, while gates spanning multiple wires generate the entanglement necessary to model multilinear mappings. In Figure 2(c) and Figure 2(d), we demonstrate an *ansatz* schema being applied to approximate the space of the learnable state of the word. Application rules correspond to a quantum operation that contract the qubits of two adjacent words or phrases; which is the Bell measurement followed by post-selection (Lorenz et al., 2021), Figure 2(b1) and Figure 2(b2). In this paper, we use a variant of the Sim14 (Sim et al., 2019) ansatz which we call **SAP**, defined by a layer of R_y rotation gates followed by a ring of controlled R_x rotations, then another layer of R_y rotation gates and a ladder of controlled R_x rotations in the opposite direction. A complete example is shown in Figure 3.

3. A Vision-Language Challenge

Vision-language understanding is a key challenge in AI, with applications to image captioning and multimodal retrieval. Models such as OpenAI's CLIP (Radford et al., 2021) have shown that large-scale joint embeddings can effectively connect visual and textual data. However, these models use transformer architectures with dense attention, which often overlook linguistic structure. As a result, there has been an increasing interest in evaluating vision-language models (VLMs) against syntactic and semantic structures such as predicate-argument meaning and word order. Various datasets have been developed for this purpose. In this paper, we consider the Attribution, Relation, and Order (ARO) (Yuksekgonul et al., 2023) and the SVO-Swap datasets. We evaluate our VQCs on these two datasets to determine whether they capture the structural relationships between predicates and nouns and whether they can correctly align text descriptions with images.

VLM architectures consist of two parallel pipelines: one for learning text and another for learning image representations in separate semantic spaces. The two are combined with an objective that aligns them in a shared output space. In our framework, the text pipeline learns quantum mean-

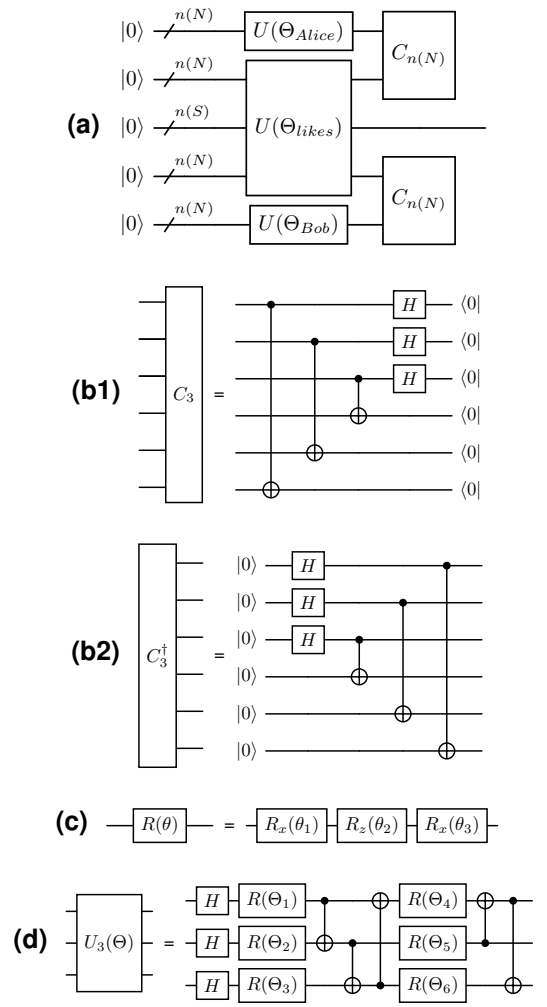


Figure 2: High level VQC for the sentence *Alice likes Bob*. Operator U depends on parameters Θ. C_n is the contractor for n number of qubits.

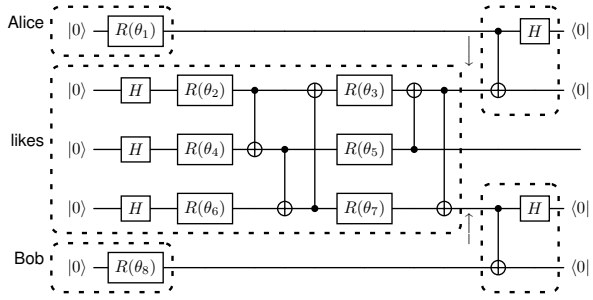


Figure 3: The quantum circuit for the sentence "Alice likes Bob" with $n(N) = n(NP) = n(S) = 1$.

ing representations in a Hilbert space. To align these representations to images, we turn the images into quantum representations using a method known as amplitude encoding. Amplitude encoding turns an m-dimensional image embedding $\overrightarrow{v}_{\text{img}}$ into the amplitudes of a quantum state ψ_{img} that consists of k qubits such that $2^k = m$. The amplitude encoding of $\overrightarrow{v}_{\text{img}} := \Sigma_i a_i \overrightarrow{e}_i$ is

$$|\psi_{\text{img}}\rangle = \frac{1}{\|v_{\text{img}}\|_2} \sum_{i=0}^{m-1} a_i |i\rangle \tag{4}$$

Where $\|v_{\text{img}}\|_2$ is the standard L_2 norm. Images are encoded using CLIP's pre-trained ViT-B/32 image encoder (Radford et al., 2021), which is kept frozen throughout training. It produces a 512-dimensional feature vector, giving $k = 9$ qubits since $2^9 = 512$. For cases where 2^k is not equal to m, the closest exponent is chosen.

The image and text representations are aligned in a unified space by maximising the similarity between matching image-caption pairs and minimising that between mismatched pairs. The objective function used in CLIP is the `InfoNCE` loss (van den Oord et al., 2019). Given a batch of N image-caption pairs $\{I_i, T_i\}_{i=1}^N$, where I_i and T_i are 512-dimensional vectors, t is a temperature parameter, and $s(\cdot, \cdot)$ is a similarity function between an image embedding I (produced by CLIP's encoder) and a caption embedding T (produced by our quantum model), the loss is:

$$\mathcal{L} = -\sum_{i=1}^{N} \log \frac{\exp(s(I_i, T_i)/t)}{\sum_{j=1}^{N} \exp(s(I_i, T_j)/t)} \tag{5}$$

In the above, s is a similarity measure. For vectors in a vector space, it is the cosine function. In our setting, we need to compute the overlap of two quantum states and develop a quantum version of `InfoNCE`, which we call `QInfoNCE`, where the overlap is defined below

$$s(|\psi_{\text{txt}}\rangle, |\psi_{\text{img}}\rangle) = |\langle \psi_{\text{txt}} \mid \psi_{\text{img}} \rangle|^2 \tag{6}$$

This is the inner product between two quantum states, living in the complex Hilbert space, known as *fidelity*. However, using fidelity directly leads to a sharp loss landscape and narrow neighbourhoods of high gradient (Cerezo et al., 2021). To alleviate this obstacle, we use a slightly modified smooth version of it, based on the Fubini-Study metric (Hai and Ho, 2023; Stokes et al., 2020; Haddou and Bennai, 2025), which can be used as a measure of similarity in the form

$$s(|\psi_{\text{txt}}\rangle, |\psi_{\text{img}}\rangle) = \arcsin(|\langle \psi_{\text{txt}} \mid \psi_{\text{img}} \rangle|) \tag{7}$$

This is essentially the quantum equivalent of the geometric distance rather than the plain Euclidean distance. It considers the curved path along a manifold between two points lying on it, instead of cutting straight through it with a straight line.

Once the specific similarity score function is chosen, we define how the model decides between a correct and incorrect caption given an image, making the corresponding classification using the

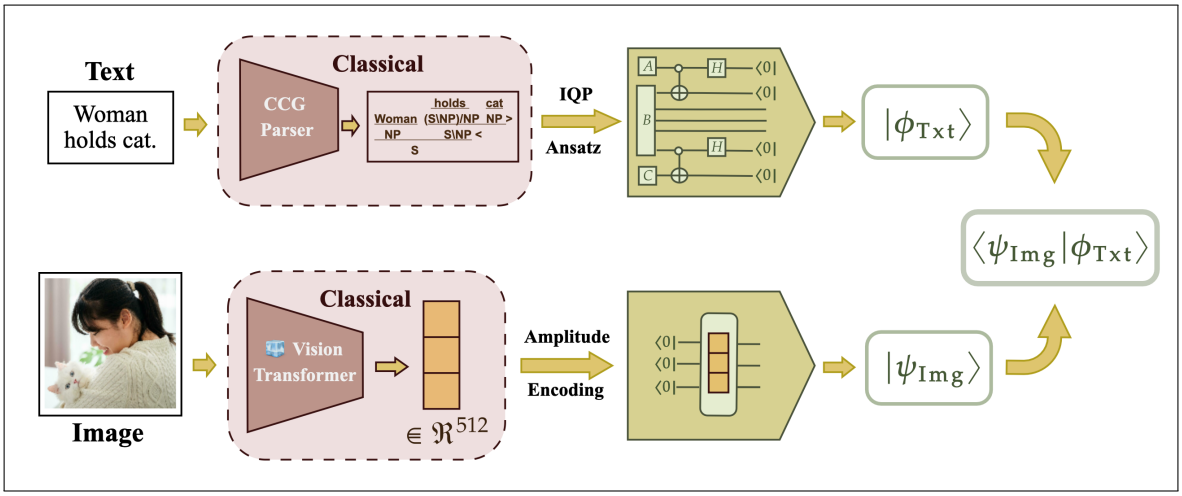


Figure 4: CCG-VQC Vision-Language Pipeline. The architecture maps multimodal inputs into quantum state space. A CCG parser and a quantum circuit are used to encode the input text into the state $|\phi_{\text{Txt}}\rangle$, while a frozen vision transformer and amplitude encoding map the input image into the state $|\psi_{\text{Img}}\rangle$. The similarity between the two modalities is computed via their inner product $\langle\psi_{\text{Img}}|\phi_{\text{Txt}}\rangle$.

'margin'. The margin $\mathcal{M}$ for an image p against a positive text label q^+ and a negative text label q^- is defined by Equation 8, where s is the comparison metric and f and g represent the classical image and quantum text encoders, respectively.

$$\mathcal{M}(p, q^+, q^-) = s(f(p), g(q^+)) - s(f(p), g(q^-)) \quad (8)$$

Therefore, if $\mathcal{M} > 0$ the correct positive label is predicted and if $\mathcal{M} < 0$ the negative incorrect label is selected instead. A higher margin magnitude reflects greater model confidence.

4. Experiments and Results

We evaluated our model on two datasets. The first is the ARO benchmark (Yuksekgonul et al., 2023), which probes noun-attribute and predicate-argument understanding. The noun-attribute subset of the dataset is referred to as **ARO-Attribution** and its predicate-argument subset as **ARO-Relation**. An example of an image entry from ARO-Attribution is the image of a silver fork on a plate with a piece of cake which has dark brown icing. The task is to choose between one of the two pieces of texts: one that describes it as "silver fork and dark brown icing" and another that says "dark brown fork and silver icing". Notably, one piece of text is obtained from the other by swapping the attributes of the nouns. In this case, the nouns are "fork" and "icing", whereas the attributes are "silver" versus "dark brown". Similarly, each entry of ARO-Relation has an image and two pieces of text describing it, where the verb is changed from one text to the other. An example image here is that of a red bus which is to the right of a big building, and the following two pieces of text: "the red bus is to the right of the big building" versus "the big building is to the right of the red bus".

We also evaluate our model on **SVO-Swap** (Lo et al., 2025), a dataset similar to ARO-Relation but inspired by the SVO-Probes benchmark (Hendricks and Nematzadeh, 2021). It works with diverse verb types, whereas the only verb used in ARO is the verb "to be". An example image is that of a woman holding a cat, described by the two captions "A woman holds a cat" versus "A cat holds a woman". SVO-Swap is a small pilot dataset of 95 evaluation pairs, created from SVO-Probes by swapping subjects and objects in its captions.

We trained the model using the `Adam` optimiser (Kingma and Ba, 2015) along with the ReduceLROnPlateau scheduler from the PyTorch python library. We use a batch size of 256 for SVO-Swap and 512 for ARO. We also tested a variety of combinations of qubits and layers for our ansätze. We denote these varieties by pairs (n_q, n_l), where n_q is the number of qubits and n_l is the number of layers. We tested with the following varieties: (1,2), (2,2), (3,2), (4,3), and (5,3). Additionally, we test our CCG-VQC with two different ansatz.

To ensure a fair comparison, we evaluate our approach against two baselines: a VQC made from a bag-of-words model, called *QBoW* and a family of text transformer models sharing the same architecture as the ones used in CLIP but with greatly reduced parameters, termed *MicroCLIP*. In the QBoW each word is translated into an independent ansatz. Words are combined with each other by taking their Frobenius multiplications, their pointwise multiplication. This is a commutative operation so QBoW does not even preserve word order let alone grammatical structure. To ensure

Attribution (Sim: 0.6578, Margin: -0.0198)
C: The tall tree and the green grass.
S: The green tree and the tall grass.

Relation (Sim: 1.0223, Margin: -0.0075)
C: The man is to the left of the person.
S: The person is to the left of the man.

Figure 5: Here 'sim' is the similarity score between the two captions and the 'margin' is the difference between the positive and negative similarities as defined in Equation 8. Here we look at ambiguous cases (high 'sim') from the ARO attribution and relation datasets that the model failed to correctly caption (negative 'margin'). Captions are labelled as Correct (**C**) and Swapped (**S**).

our architectural contributions are evaluated independently of model scale and training volume, we introduce MicroCLIP, a family of models trained from scratch using the same data and protocol as our proposed models. To match the low parameter count of our structured VQCs (10K-100K) and achieve a fair comparison, we drastically shrink the standard CLIP text transformer encoder by reducing the number of layers from 12 to 2, attention heads from 8 to 2, and embedding dimensions from 512 to 8, 16, or 32 (yielding MicroCLIP-8, MicroCLIP-16, and MicroCLIP-32).

Lastly, we also display previous results based on the tensor networks variant of our model before conversion to VQC (Lo et al., 2025).

4.1. Results

Table 1 summarises our performance on the SVO-Swap and ARO datasets. Our fully structured VQC outperforms all others on the ARO-Attribution and ARO-Relation, as well as on the newly developed dataset SVO-Swap, demonstrating the benefit of encoding linguistic structure without training on hard negatives. The bag-of-words version of VQCs performs as expected, reaching only an accuracy of 50%, since it is commutative and ignores the word order. In this model, the quantum circuits of both texts of the entries of each dataset are equivalent up to the parameters.

Despite using 63M parameters, CLIP and its variant OpenCLIP achieve lower accuracy than the **CCG-VQC** across all benchmarks. CLIP achieves 57.89% on SVO-Swap versus 83.16% for **CCG-VQC**; 61.00% versus 71.19% on ARO-Attribution; and 51.53% versus 57.33% on ARO-Relation. OpenCLIP improves over CLIP on SVO-Swap (63.16%), but still falls short of **CCG-VQC** (83.16%), and shows no improvement on ARO.

In Figure 6(a), we show that CCG-VQC outperforms MicroCLIP while using an order of magnitude fewer parameters. As seen in Figure 6(b), the model performed best in SVO-Swap, which shows a positive mean margin and an accuracy of 83.16%. This plot shows that the majority of data had a positive margin, meaning the majority of captions were correctly aligned with images. The outliers below the decision boundary $(\mathcal{M} < 0)$ expose occasional misclassification; this was in cases where the incorrect caption was aligned with an image. A deeper look at the results reveals that in these cases the visual pairing of the subject and the object was ambiguous, i.e. the subject and object looked alike in the image or the captions were very similar, Figure 5. An interesting challenge to note is that as the model achieved 90.1% accuracy

Table 1: Results on SVO-Swap and ARO.

	SVO-Swap	ARO	
		Attribution	Relation
QBoW	50.00	50.00	50.00
CCG-VQC$_{SAP}$	**83.16**	**71.19**	**57.33**
MicroCLIP-8	68.42	50.33	50.11
MicroCLIP-16	69.99	50.27	49.84
MicroCLIP-32	69.68	50.85	51.05
CLIP	57.89	61.00	51.53
OpenCLIP	63.16	59.13	50.71

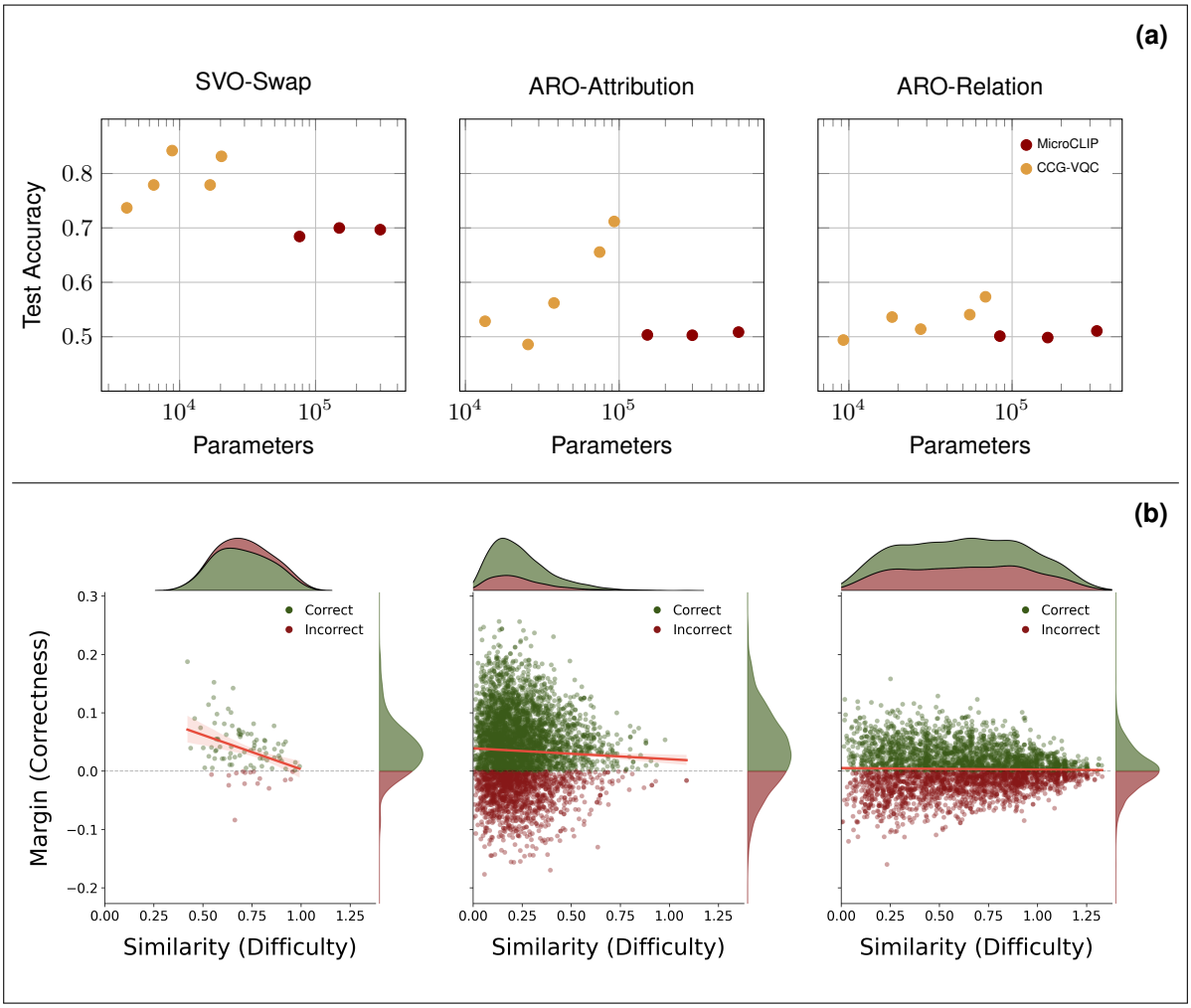


Figure 6: (a) Test accuracy vs. trainable parameters. CCG-VQC achieves competitive or superior accuracy while requiring an order of magnitude fewer parameters than MicroCLIP. (b) Margin of correctness (Equation 8) against caption similarity. Regression lines (red) illustrate performance degradation as lexical similarity (difficulty) increases. Marginal distributions on each axis indicate the density of correct (green) and incorrect (red) classifications relative to the decision boundary (dotted line).

during training on the ARO-Attribution dataset this visually contrasts with the test accuracy of 71.19%. This suggests a generalisation gap when applying learned adjective-noun structures to unseen data. Moreover, for the ARO-Relation dataset both the training accuracy and test accuracy remained near the 50% mark. This suggests a structural collapse when trying to capture spatial relations.

5. Discussion and Analysis

The overall performance of our model across all datasets demonstrates that the use of VQCs and quantum states, in most cases, enabled us to successfully create structurally aware meaning representations capable of capturing the asymmetric behaviour of verbs and other predicates, such as prepositions and adjectives. Here, our **SAP** ansatz did the best and CLIP and OpenCLIP fell short due to their transformer architecture having a lack of explicit structural inductive bias.

This is most evidently seen in the SVO-Swap dataset, where our **CCG-VQC** model achieves an accuracy of 83.16%. SVO-Swap has a primitive linguistic structure; its sentences consist of only three roles: subject, verb, and object. This result demonstrates that when parse complexity is minimised, the model learns it well. This extends to ARO-Attribution, achieving an accuracy of 71.19%, further showing that our model can effectively assign attributes to corresponding objects. On ARO-Relation, which is the hardest of our benchmarks, the model achieves an accuracy of only 57.33%, while low in isolation, many classical models only manage to reach an accuracy of around 50%. This indicates that our model can capture some, albeit minimal, relational dependencies.

This performance shift highlights limitations of

the CCG framework for complex linguistic tasks. Hence after conducting error analysis, in particular investigating the similarity between captions, we noticed the model's lower accuracy seems to arise from the text encoder failing to distinguish between very similar captions. In our failure cases, ARO-Relation text circuits showed high similarity, contrasting with the clear separation observed in ARO-Attribution (see Figure 5). ARO-Relation underperforms compared to SVO-Swap for two main reasons: its verbs lack strong selection preferences, and it contains more complex sentences largely consisting of prepositional phrases. The CCG trees for these phrases are highly nested. This leads to high-rank tensors that disperse semantic information and saturate the representation space, which in turn forces the meaning representations to collapse into a very concentrated region of the Hilbert space. This heavily affects our model's discriminative power. It would be fruitful to explore the relationship between the parse complexity of such models and their resulting performance in terms of accuracy.

There are also structural asymmetries between the text and image modalities. Unlike text, we use a very basic VQC for images, which does not reflect its semantic structure. Further, this encoding is built on the frozen classical vision transformer, which treats the images as a flat grid of patches. As a result, the quantum text encoder is forced to do all the structural heavy lifting, trying to align a highly structured linguistic state to a static and structurally flat visual state, potentially causing the representation to lose structural awareness.

The integration of these modalities introduces further architectural challenges and topological mismatches. Similar to CLIP, our approach relies on late fusion, where the text and image are processed in complete isolation and only interact at the end via similarity. This prevents the text's relational structure from explicitly guiding the visual embeddings or vice versa. Indeed, it has been shown theoretically that no joint embedding space can simultaneously represent concept categorisation, attribute binding, and spatial relationships when cross-modal interaction is reduced to a single scalar similarity score (Kang et al., 2025). Additionally, there is a fundamental topological mismatch between the learnt text embeddings in the complex Hilbert space and the frozen image embeddings originally learnt in Euclidean space. Enforcing alignment on a static Euclidean-informed image manifold crudely projected onto the complex Hilbert space risks stripping the quantum state vectors of their delicate structural representations, further contributing to a potential alignment collapse. Overall, an interesting future direction would be to explore where specifically these bottlenecks lie. If a learnable image component was used with more sophisticated fusion would the text encoder actually suffer from the aforementioned issues or would it perform well in spite of them.

6. Summary and Outlook

In this paper, we introduced **CCG-VQC**, a meaning representation for natural language that uses Variational Quantum Circuits (VQCs) guided by the rules of Combinatory Categorial Grammar (CCG). By mapping syntactic CCG derivation trees into quantum circuits, our model represents words as trainable quantum states and linguistic compositions as entangling operations, encoding sentence structure directly in the circuit topology. Our experiments show that this structural inductive bias leads to consistent gains on compositional benchmarks.

The unstructured quantum bag-of-words baseline (QBoW) collapses to 50% accuracy on every dataset, while **CCG-VQC** reaches 83.16% on SVO-Swap, 71.19% on ARO-Attribution, and 57.33% on ARO-Relation, surpassing both CLIP and OpenCLIP despite using two orders of magnitude fewer parameters. A parameter-matched MicroCLIP baseline confirms that the gains stem from architectural structure rather than scale alone.

Several directions remain open for future work. An interesting area to explore is how different grammar-based topologies handle deep nesting in complex linguistic structures and whether this helps reduce parse dilution. One interesting alternative that could be explored is Universal Dependency (UD) grammars. In multimodality, a two-fold improvement would involve an early-fusion or unified space with a learnable image encoder, allowing structurally aware text embeddings to inform image embeddings and enabling bidirectional alignment that better preserves embedding topology.

Pre-training on larger datasets such as MSCOCO (Lin et al., 2014) and ConceptualCaptions (Sharma et al., 2018) should improve generalisation, particularly for ARO-Relation.

Evaluating **CCG-VQC** on structural NLP benchmarks beyond VLM tasks is another natural future work, as is assessing its robustness to gate noise and running it on near-term quantum hardware.

References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Patrick Blackburn and Johan Bos. 2005. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. Center for the Study of Language and Information.

D. A. Boiko, R. MacKnight, B. Kline, et al. 2023. Autonomous chemical research with large language models. *Nature*, 624:570–578.

M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. 2021. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1).

Stephen Clark. 2021. Something old, something new: Grammar-based ccg parsing with transformer models.

Bob Coecke, Edward Grefenstette, and Mehrnoosh Sadrzadeh. 2013. Lambek vs Lambek: Functorial vector space semantics and string diagrams for Lambek calculus. *Annals of Pure and Applied Logic*, 164(11):1079–1100.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2015. Concrete models and empirical evaluations for the categorical compositional distributional model of meaning. *Computational Linguistics*, 41(1):71–118.

Marwan Ait Haddou and Mohamed Bennai. 2025. Sculpting quantum landscapes: Fubini-study metric conditioning for geometry aware learning in parameterized quantum circuits.

Vu Tuan Hai and Le Bin Ho. 2023. Universal compilation for quantum state tomography. *Scientific Reports*, 13(1):3750.

Lisa Anne Hendricks and Aida Nematzadeh. 2021. Probing image–language transformers for verb understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3635–3644, Online. Association for Computational Linguistics.

T. Hubert, R. Mehta, L. Sartran, et al. 2025. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 632:290–297.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip. If you use this software, please cite it as below.

Raphi Kang, Yue Song, Georgia Gkioxari, and Pietro Perona. 2025. Is clip ideal? no. can we fix it? yes! In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22436–22446.

Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. 2021. lambeq: An Efficient High-Level Python Library for Quantum NLP. *arXiv preprint arXiv:2110.04236*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Darina Koishigarina, Arnas Uselis, and Seong Joon Oh. 2025. Clip behaves like a bag-of-words model cross-modally but not uni-modally.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks.

J. Lambek. 1999. Type grammar revisited. In *Logical Aspects of Computational Linguistics*, pages 1–27, Berlin, Heidelberg. Springer Berlin Heidelberg.

Martha Lewis, Nihal Nayak, Peilin Yu, Jack Merullo, Qinan Yu, Stephen Bach, and Ellie Pavlick. 2024. Does CLIP bind concepts? probing compositionality in large image models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1487–1500.

Haobo Li, Zhaowei Wang, Jiachen Wang, Yueya Wang, Alexis Kai Hon Lau, and Huamin Qu. 2025a. Cllmate: A multimodal benchmark for weather and climate events forecasting. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Tianle Li, Jihai Zhang, Yongming Rao, and Yu Cheng. 2025b. Unveiling the compositional ability gap in vision-language reasoning model.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer.

Kin Ian Lo, Hala Hawashin, Mina Abbaszadeh, Tilen Gaetano Limbäck-Stokin, Hadi Wazni, and Mehrnoosh Sadrzadeh. 2025. DisCoCLIP: A distributional compositional tensor network encoder

for vision-language understanding. In *Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM 2025)*, pages 316–327, Suzhou, China. Association for Computational Linguistics.

Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. 2021. Qnlp in practice: Running compositional models of meaning on a quantum computer.

Jean Maillard, Stephen Clark, and Edward Grefenstette. 2014. A type-driven tensor-based semantics for CCG. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 46–54.

Glyn Morrill. 2010. *Categorial Grammar: Logical Syntax, Semantics, and Processing*. Oxford University Press.

Ruikang Ni, Da Xiao, Qingye Meng, Xiangyu Li, Shihui Zheng, and Hongliang Liang. 2024. Benchmarking and understanding compositional relational reasoning of llms.

Michael A. Nielsen and Isaac L. Chuang. 2000. *Quantum Computation and Quantum Information*. Cambridge University Press.

H. Pan, N. Mudur, W. Taranto, et al. 2025. Quantum many-body physics calculations with large language models. *Communications Physics*, 8:123–130.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2556–2565.

Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12).

K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, et al. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31:943–950.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

Mark Steedman and Jason Baldridge. 2011. *Combinatory Categorial Grammar*, chapter 5. John Wiley & Sons, Ltd.

James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. 2020. Quantum natural gradient. *Quantum*, 4:269.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Hadi Wazni, Kin Ian Lo, Lachlan McPheat, and Mehrnoosh Sadrzadeh. 2024. Large scale structure-aware pronoun resolution using quantum natural language processing. *Quantum Machine Intelligence*, 6(2):60.

Hadi Wazni and Mehrnoosh Sadrzadeh. 2023. Towards transparency in coreference resolution: A quantum-inspired approach. In *Proceedings of The Sixth Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2023)*, pages 15–27, Singapore. Association for Computational Linguistics.

Gijs Wijnholds, Mehrnoosh Sadrzadeh, and Stephen Clark. 2020. Representation learning for type-driven composition. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 313–324.

Richie Yeung and Dimitri Kartsaklis. 2021. A CCG-based version of the DisCoCat framework. In *Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science (SemSpace)*, pages 20–31, Groningen, The Netherlands. Association for Computational Linguistics.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2023. When and why Vision-Language Models behave like Bags-of-Words, and what to do about it? In *International Conference on Learning Representations (ICLR)*.

Superframes: A Schema for Lexicon-free Frame-semantic Annotation

Kilian Evang
Heinrich Heine University Düsseldorf, Germany
evang@hhu.de

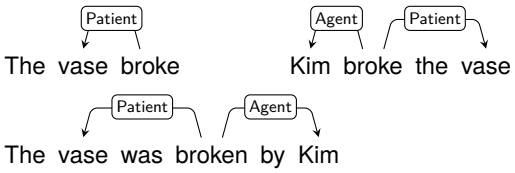
Abstract

Semantic frame, role, and relation labels are important parts of symbolic meaning representations. The mainstream approach is to define large language-specific lexicons that map predicate senses to their frame and argument labels, and to use separate label inventories for modifiers. Maintaining lexicons is very labor-intensive and scales poorly to the multilingual case. There are schemas that aim to simplify the annotation task using a more coarse-grained inventory of frames and roles, but suffer from a lack of systematicity and clear definitions. We present a schema that 1) uses a small inventory of frames and roles for lexicon-free annotation, 2) systematizes the frame inventory by factoring out aspect and mode, 3) has a unified vocabulary for arguments and modifiers, and 4) is designed to be annotated atop Universal Dependencies syntactic annotation. We argue for the adoption of such a schema for multilingual annotation and demonstrate promising results in annotation experiments on German.

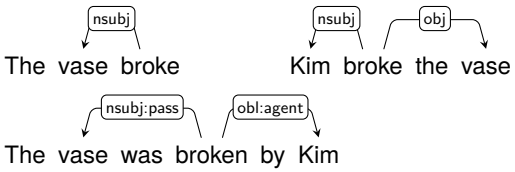
Keywords: frame semantics, semantic roles, annotation

1. Introduction

Semantic role labeling (SRL) is a classic task in natural language processing (NLP; Gildea and Jurafsky, 2002). Given a predicate and its arguments, SRL aims to answer the question "who does what to whom, where and when", i.e., which argument plays which role in the situation described. It abstracts away from syntactic alternations such as causative or passive, as exemplified in (1).

(1)


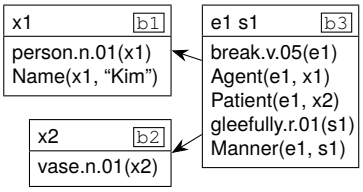
The situations described are similar, and consequently the semantic roles of all arguments (*Kim* and *the vase*) stay the same across alternations, while the syntactic roles differ, as seen in (2).

(2)


Because of this abstracting away from syntactic differences, SRL is an important upstream or auxiliary task for semantic applications such as relation extraction (e.g., Tsai et al., 2006; Christensen et al., 2010; Guo et al., 2016; Humphreys et al., 2020; Zhang et al., 2022), dialogue systems (e.g., Rachman et al., 2018; Xu et al., 2020; Wu et al., 2022), Natural Language Inference (Cengiz and Yuret, 2020), or Natural Language Understanding in general (Zhang et al., 2020). Finally, semantic roles are a central part of formal meaning representations in frameworks such as the Parallel Meaning Bank (PMB; Abzianidze et al., 2017) or Abstract Meaning Representations (AMR; Banarescu et al., 2013). Such meaning representations are useful for semantic parsing and providing structured input to downstream applications such as chatbots and information extraction systems, but can also be useful to corpus linguists who want to query for specific semantic structures. Different meaning representation frameworks use different semantic role inventories. For example, the PMB uses a small set of generic roles across all predicates and across arguments and modifiers. AMR, on the other hand, uses predicate-specific roles based on PropBank (Palmer et al., 2005) for arguments, and a separate set of relations for modifiers. Examples of both are shown in Figure 1.

Syntactic analyses of sentences are these days relatively easy to produce automatically. Software libraries like spaCy (Montani et al., 2023), Stanza (Qi et al., 2020), or UDPipe (Straka et al., 2016) come with readily trained syntactic parsing models for many languages. Part of the reason for this wide availability is the success of the Universal Dependencies project (UD; de Marneffe et al., 2021), from which the syntactic relations shown in (2) are drawn. UD emphasizes universality (language-independence) over fine-grained distinctions, using a relatively coarse inventory of syntactic relations, which enables easy annotation, a shared understanding between annotators of different languages, therefore a cost-effective way to produce large amounts of training data, and the ability for machine-learning models to generalize

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 124–135




```
(b / break-01
   :ARG0 (k / Kim)
   :ARG1 (v / vase)
   :manner (g / gleeful))
```

Figure 1: A DRS and an AMR for the sentence “Kim broke the vase gleefully”

multilingually and cross-lingually (Kondratyuk and Straka, 2019; Samardžić et al., 2022). UD corpora (manually or automatically annotated) have been used for psycholinguistic and cross-linguistic corpus research, e.g., on word order (Futrell et al., 2015; Guzmán Naranjo and Becker, 2018; Levshina, 2019). But since UD only provides morphosyntactic relations, it is less well suited for similar corpus linguistic research on semantic phenomena, or for NLP tasks such as relation extraction and dialogue systems.

Given the advantages of semantic roles, one would think that SRL tools for many languages should be as readily available as UD parsers. However, this is not currently the case. None of the software libraries mentioned above comes with the capability to predict semantic roles. Freely available SRL systems are usually research prototypes and come with pre-trained models only for a single or few languages and domains (e.g., He et al., 2017; Strubell et al., 2018; Larionov et al., 2019). To a large part this gap is due to the lack of a language-independent annotation standard for semantic roles, comparable to the UD annotation guidelines for syntax. In Section 2, we review existing schemas for semantic role annotation and point out limitations that hamper their widespread adoption. We argue that bottlenecks stem mainly from either lack of frames or too many and fine-grained frames, and from restrictions of schemas to specific morphosyntactic word classes. We present a scheme that addresses these issues in Section 3, report on annotation experiments in Section 4, and conclude in Section 5.

2. Limitations of Existing SRL Annotation Schemes

Large frame inventories Most schemas define semantic roles via *frames*: A predicate is labeled as evoking a frame such as `break-01`, which defines frame-specific roles such as `ARG0` (breaker) and `ARG1` (thing broken). Some schemas have rather large and fine-grained inventories of frames. For example, PropBank has 10 687, SynSemClass has 1 993 (Urešová et al., 2022; Urešová et al., 2025), FrameNet has 1 224 (Baker et al., 1998), and VerbAtlas, 466 (Di Fabio et al., 2019). This creates a practical problem in annotation: for each predicate instance, annotators have to find the most appropriate frame, which is non-trivial, and fine-grained distinctions mean that the probability of disagreement is high. It may also be that an appropriate frame does not exist yet and has to be created. Frame lexicons mitigate this problem by creating explicit mappings from lexemes to frames. However, such lexicons are still forever incomplete, and they are language-specific. For a new language, a new lexicon has to be created, or annotators have to translate predicates, which introduces additional ambiguity and disagreement. There are ways to annotate non-English corpora with PropBank frames and roles automatically (Akbik et al., 2015; Jindal et al., 2022), but this also introduces errors.

Focus on specific morphosyntactic word classes In annotation, it is important to determine whether and when an annotation is complete. In UD, a syntactic annotation is complete when the dependency edges form a tree over the words of the sentence. A comprehensive semantic role annotation should label the semantic function of every argument and modifier edge between content words. Existing schemas, however, have a tendency to focus on a specific subset of natural-language predicates and their arguments and to some extent modifiers, typically by morphosyntactic word class. VerbNet (Kipper Schuler, 2005) and VerbAtlas (Di Fabio et al., 2019) focus on English verbs; so did PropBank, initially, although support for other words such as nouns and adjectives was later added (Pradhan et al., 2022). FrameNet focused on “nouns, adjectives, and verbs” (Baker et al., 1998) from the start. Modifier relations can be considered to evoke a frame on their own that is not necessarily evoked by a content word (although it is often associated with an adposition or conjunction). PropBank and AMR define different inventories of modifier relations that are separate from the inventory of frames for verbs and other frame-evoking content words. There are also schemas that specialize in other relations that sometimes coincide with modifier relations, such as adposition senses (Schneider et al., 2018), discourse relations (Carlson et al., 2003; Prasad et al., 2008), or compound relations (Tratz and Hovy, 2010; Pepper, 2022). FrameNet and SynSemClass comprehensively address arguments and modifiers of all content word classes, but there is no single *compact* set of frames that does this. As a result, treatment of predicates of different word classes and

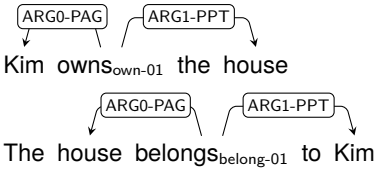
```
(o / own-01
   :ARG0 (k / Kim)
   :ARG1 (h / house))

(h / house
   :poss (k / Kim))
```

Figure 2: AMR annotations of the phrases *Kim owns the house* and *Kim's house*

especially between arguments and modifiers remains somewhat incoherent in many existing annotation frameworks. To illustrate this point, consider the phrases *Kim owns the house* and *Kim's house*. Both evoke a similar frame, namely a possessor-possessum relationship between Kim and a house. Now consider their AMR annotations in Figure 2. The similarity of the situations denoted is lost in this representation because the verb *own* and the possessive modifier draw from different inventories of labels.

Lack of precise definitions for roles Considering the problems with large numbers of frames discussed above, one might be tempted to do away with frames entirely and label each argument (and modifier) in isolation, according to a small set of role labels, each defined in some way. This approach is taken by schemas such as LIRICS (Petukhova and Bunt, 2008, used with some modifications in the PMB) or WISeR (Feng et al., 2023). In addition, several schemas do use some notion of frame, but attempt to give meaningful and frame-independent names to roles that are supposed to enable annotators to determine the correct role for each argument even without reference to the frames. Examples of this are VerbAtlas, the tectogrammatical level of the Prague Dependency Treebank (Hajič et al., 2024), and the "functions" decorating PropBank's frame-specific numbered argument roles. However, it is very hard to clearly define roles without reference to frames, because what makes different arguments of one predicate distinct from each other is their relation to each other. To illustrate this point, consider the sentences *Kim owns the house* and *The house belongs to Kim*. Both evoke a similar frame, namely a possessor-possessum relationship between Kim and a house. Now consider their PropBank annotations (showing both numbered argument roles and "functions") in (3).

(3)


In the first annotation, Kim is PAG (prototypical agent) and the house is PPT (prototypical patient) whereas in the second, it is the other way around. The similarity between the situations described is missed. Moreover, it is not clear how annotators could determine that these are the correct role labels without looking them up in the frame inventory, or in a language-independent way.

3. The Superframes Annotation Scheme

3.1. Design Principles

Superframes was developed to enable rapid, consistent annotation with semantic roles the way UD does for syntactic relations. To this end, Superframes is designed with the following key features:

(1) Small number of frames There are only a few dozen frames, shown in Table 1, all binary and indicating coarse semantic relations between two concrete or abstract entities. In annotation, predicates are sorted into these coarse semantic classes, similar to the sorting of word senses into coarse "supersenses" by Ciaramita and Altun (2006); Schneider et al. (2018). For annotation consistency, frames are kept as few and general as possible, but as many and specific as necessary to provide intuitive labels for each argument. For example, while the predicates *own* and *include* both have two arguments, the relation between both arguments in each case is different enough to warrant separate frames: POSSESSION with roles possessum and possessor, and PART-WHOLE with role part and whole, respectively.

(2) Frame composition Predicates with more than two arguments are annotated by composing two or more frames. For example, *include* with a causativ subject *include* evokes an additional CAUSATION frame where the core PART-WHOLE frame fills the result role.

(3) Aspect decomposition Frames denoting state changes are derived from frames denoting the corresponding states. For example, while *own* evokes POSSESSION, *buy* evokes POSSESSION-INIT.

(4) Unified frame inventory The same inventory of frames is used to annotate arguments of all classes of content words, and all modifiers.

(5) Constrained granularity Superframes is annotated atop Universal Dependencies. Content words receive frame labels, and the edges connecting them receive argument or modifier labels. In addition, secondary edges for nonlocal dependencies arising from coordination, control, raising,

SUPERFRAME	initial-arg2	arg1	arg2	transitory-arg2	target-arg2
SITUATION	initial-situator	theme	situator	transitory-situator	target-situator
└ ACCOMPANIMENT	initial-accompanier	accompanied	accompanier		target-accompanier
└ DEPICTIVE		has-depictive	depictive		
└ ASSET		has-asset	asset		
└ ATTRIBUTE		has-attribute	attribute		
└ COMPARISON		compared	reference		
└ CONCESSION		asserted	conceded		
└ EVENT		undergoer	event		
└ ACTIVITY		is-active	activity		
└ EXISTENCE	initial-exists	material	exists		target-exists
└ REPRODUCTION		original			copy
└ TRANSFORMATION-CREATION		material			created
└ EXPERIENCE		experiencer	experienced		
└ EXPLANATION		explained	explanation		
└ PURPOSE		has-purpoe	purpose		
└ IDENTIFICATION	initial-identifier	identified	identifier		target-identifier
└ LOCATION	initial-location	has-location	location	transitory-location	target-location
└ ADORNMENT-TARNISHMENT	initial-surface	ornament	surface		target-surface
└ EXCRETION	excreter	excreted		transitory-location	target-location
└ HITTING		hitting	hit		
└ INGESTION		ingested		transitory-location	ingester
└ UNANCHORED-MOTION		in-motion		transitory-location	
└ WRAPPING-WEARING	initial-wearer	worn	wearer		target-wearer
└ MEANS		has-means	means		
└ MESSAGE	initial-content	topic	content		target-content
└ MODE		has-mode	mode		
└ NONCOMP		has-noncomp	noncomp		
└ PART-WHOLE	initial-whole	part	whole		target-whole
└ POSSESSION	initial-possessor	possessed	possessor		target-possessor
└ QUANTITY	initial-quantity	has-quantity	quantity		target-quantity
└ RANK	initial-rank	has-rank	rank		target-rank
└ SCENE	initial-scene	participant	scene	transitory-scene	target-scene
└ STATE	initial-state	has-state	state		target-state
└ QUALITY	initial-quality	has-quality	quality		target-quality
└ CLASS	initial-class	has-class	class		target-class
└ DESTRUCTION		destroyed			
└ SENDING		sent	sender		
└ SEQUENCE		follows	followed		
└ CAUSATION		result	causer		
└ CONDITION		has-condition	condition		
└ EXCEPTION		has-exception	exception		
└ REACTION		reaction	trigger		
└ RESULTATIVE		has-resultative	resultative		
└ SOCIAL-RELATION	initial-social-relation	has-social-relation	social-relation		target-social-relation
└ TIME	initial-time	has-time	time		target-time

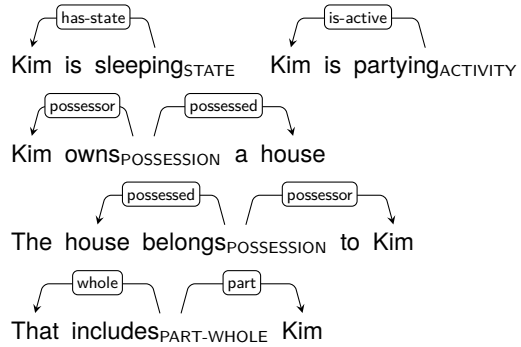
Table 1: Hierarchy of superframes and their roles

secondary predicates, etc., are added and labeled. No additional nodes are introduced.

3.2. Overview of the Annotation Guidelines

Superframes is annotated atop UD. Content words (tagged NOUN, VERB, ADJ, or ADJ in UD) receive a frame label, and dependency arcs between content words (labeled nsubj, dobj, nmod, etc.) receive a role label.

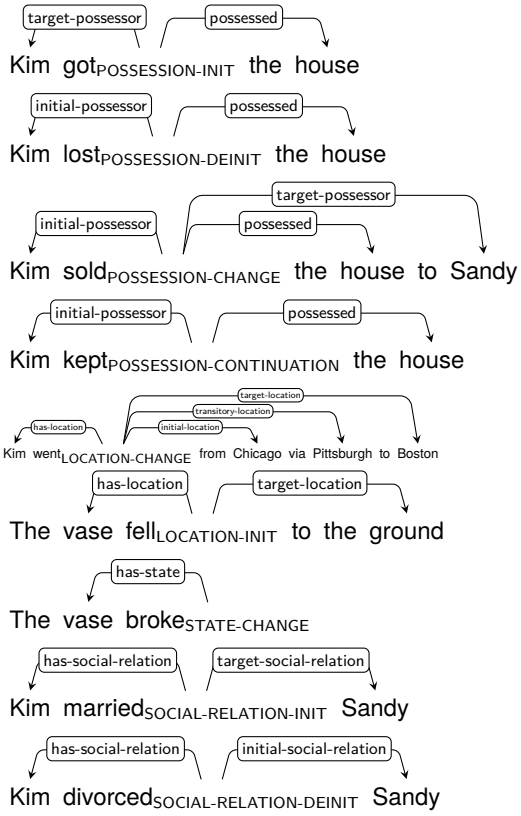
(4)



Stative verbal predicates An example of the annotation of some unary and binary verbal stative predicates are shown in (4). Role labels have to be drawn from the frame that the predicate is annotated with.

Aspect Predicates denoting events that can be framed in terms of an actual or hypothetical change of state are framed using one of the aspect operators -INIT, -DEINIT, -CHANGE, or -CONTINUATION. The second argument accord-

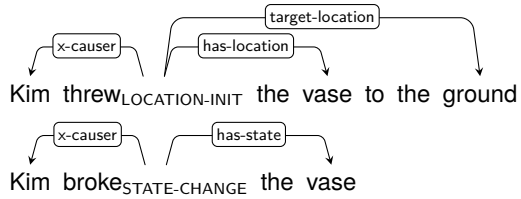
ingly receives a role label prefixed with initial-, target-, or transitory-. Examples are shown in (5).

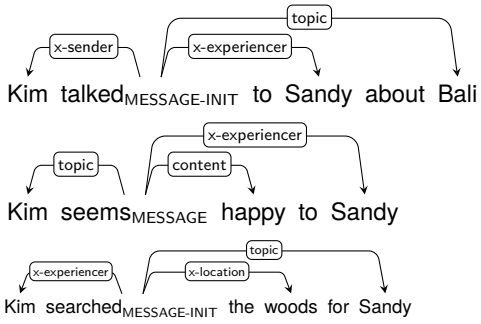
(5)


Mode In addition to aspect operators, we also use the modal operators -NECESSITY, -POSSIBILITY, and -NEG. An example are shown in (6).

(6)


Frame composition Predicates can be annotated as evoking more than one frame, with a limited form of embedding. For example, a causative subject is considered to evoke an additional CAUSATION frame and fill its causer role while the core frame fills the result role. To avoid introducing an additional CAUSATION node into the annotation format, a special notation indicates this configuration, annotating the causative subject as x-causer. Analogously, arguments such as senders and perceivers are annotated with x-sender and x-experiencer, respectively. This frame composition mechanism is also used for predicate-specific arguments that do not fit into onto one of the core argument slots. Examples are shown in (7).

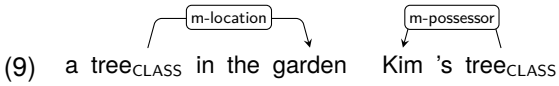
(7)




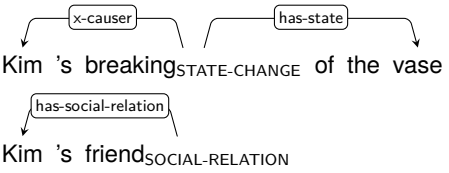
Modification Like non-core arguments, modifiers are assumed to evoke an additional frame and labeled with the role they fill in that frame, but with the prefix m- marking them as modifiers. Examples are shown in (8).

(8)

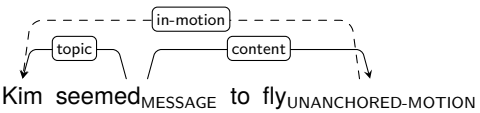

Nonverbal predicates We have so far shown only examples of *verbal* predicates, but everything applies to the annotation of nonverbal predicates as well. An ordinary noun like *tree* evokes the CLASS frame, marking the entity it refers to as a member of a class (in this case: the class of trees). There are no arguments here because the predicate itself doubles as a referent. However, it can be modified, as shown in (9).

(9)


Event nouns and relational nouns evoke frames and have arguments just like verbs, as shown in (10).

(10)


Nonlocal dependencies Many constructions introduce semantic predicate-dependent dependencies that do not correspond to (surface) syntactic dependencies. This includes constructions like control, raising, relative clauses, secondary predicates, coordination, etc. We add such links in the annotation beyond those provided by UD. Examples are shown in (11) with nonlocal dependencies dashed.

(11)


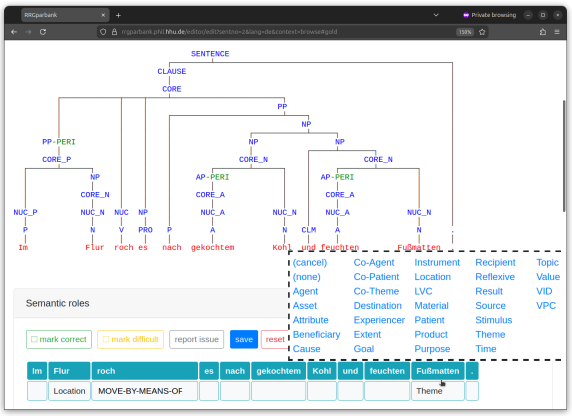


Figure 3: Browser-based VerbAtlas annotation interface showing the German translation of the sentence “The hallway smelt of boiled cabbage and old rag mats”

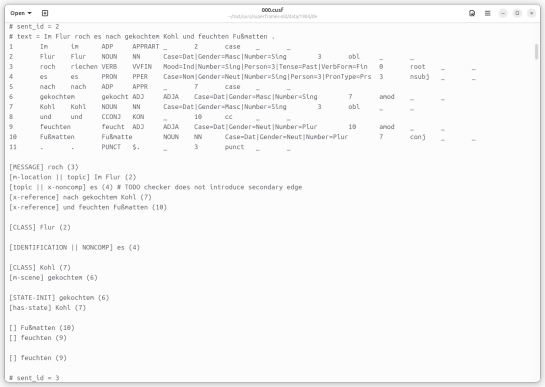
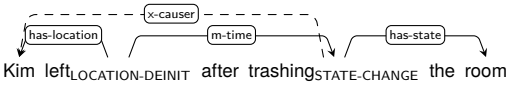
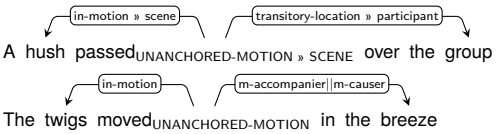


Figure 4: Text-based Superframes annotation interface. There is a UD syntax tree and 7 predicates, 5 of which have already been annotated with frames and roles.



Figurativity and uncertainty Figurative use of predicates is a pervasive phenomenon and often makes it hard to choose a single frame for a predicate because a more literal meaning and a more idiomatic reading are both salient. In such cases, annotators are instructed to provide annotations for both readings using labels of the form A » B where A is for the more literal reading and B is for the more figurative reading. When it is hard to choose a predicate for reasons other than figurativity, annotators can likewise provide two alternatives using the A || B notation. Examples are shown in (12).

(12)


4. Experiments

To demonstrate the viability of Superframes as an annotation scheme, we report the results of two pilot annotation projects, both carried out on a German translation of George Orwell’s novel *1984* (Orwell, 2003), one using the VerbAtlas annotation scheme and one using Superframes. Both annotation projects were carried out by paid student assistants who were either currently pursuing or had recently completed a bachelor’s degree in computational linguistics, and who were native or advanced speakers of German. Annotators worked independently but came together for weekly annotation meetings with the author, discussing difficult cases and producing consensus annotations for them.

VerbAtlas VerbAtlas (Di Fabio et al., 2019) is a frame inventory with full explicit coverage of the verb senses in WordNet 3.0 (Fellbaum, 1998). It contains 466 frames created by semiautomatically clustering WordNet synsets and 25 semantic role labels that are shared between frames, such as Agent, Patient, Source, Goal. Annotation was carried out between July 2021 and March 2024. Annotators were provided an annotation manual (Evang, 2024) and a graphical annotation interface that showed them syntactic trees for the sentences they had to annotate (taken from RRGparbank; Bladier et al., 2022) as well as preliminary annotations generated using InVeRo (Conia et al., 2020, 2021), which they had to correct (see Figure 3 for an example). Annotators were instructed to find the most appropriate VerbAtlas frame for each verbal predicate, considering mainly meaning correctness (the frame should map to at least one English WordNet synset whose gloss fits the use of the predicate in this context) and role coverage (the frame should have an appropriate role for every argument present in the instance). They were also instructed to make sure the role labels matched their use in the chosen frame. They were given a searchable document containing for each VerbAtlas frame 1) its VerbNet synsets with lemmas, glosses, and example sentence 2) its PropBank frames with role descriptions and annotated example sentences. No modifiers ore nonverbal predicaets were annotated. The same annotators annotated the corresponding parallel English sentences in parallel. Statistics of the resulting annotations are shown in Table 2.

Superframes The annotation was carried out between April 2024 and March 2025. Annotators were provided an annotation manual (reference anonymized) and text files that contained UD anno-

Annotator(s)		Sentences	Predicates	Arguments
jh		2 192	4 784	13 475
sg		2 405	5 122	14 799
sk		2 175	4 606	12 445
jh+sg	Both annotated	893	1 863	4 669
	Agreement		.74	.75
jh+sk	Both annotated	566	1 113	2 466
	Agreement		.76	.80
sg+sk	Both annotated	765	1 604	3 548
	Agreement		.71	.81

Table 2: VerbAtlas annotation results. The lower part excludes sentences with consensus annotations.

Annotator(s)		Sentences	Predicates	Dependents
ab		350	4 168	4 219
sg		296	2 955	2 431
xs		300	3 213	3 404
ab+sg	Both annotated	147	476	308
	Agreement (strict)		.71	.63
	Agreement (lax)		.77	.68
	Agreement (superlax)		.81	.73
ab+xs	Both annotated	150	655	609
	Agreement (strict)		.78	.73
	Agreement (lax)		.84	.81
	Agreement (superlax)		.89	.86
sg+xs	Both annotated	99	885	869
	Agreement (strict)		.85	.79
	Agreement (lax)		.92	.84
	Agreement (superlax)		.93	.86

Table 3: Superframes annotation results. The lower part excludes sentences with consensus annotations.

tations of the sentences to annotate as well as blank templates for filling in frame and role labels (see Figure 4 for an example). The templates were generated automatically from the UD annotation. Annotators were provided a Python script with which they checked their annotations for validity (all blanks filled out, all frame and role labels defined, all role labels appropriate for the respective frame, etc.) before committing them to a shared Git repository. Statistics of the resulting annotations are shown in Table 3. Strict agreement is defined as both annotators choosing the exact same label for a predicate or dependency edge. Lax agreement is also satisfied if they agree in at least one sub-label when using the uncertainty/figurativity mechanism. Superlax agreement is like lax agreement but ignores all prefixes or suffixes in labels such as -INIT, m-, or target-, focusing only on the main frame or role label.

Close comparison 69 sentences were doubly annotated with no consensus annotation under both schemas. We examine this intersection more closely and report overall agreement between the respective two annotators in Table 4.

Results Looking at the close comparison in Table 4, we see that with strict agreement, Superframes scores lower than VerbAtlas for frame labels and role labels. Agreement becomes higher than for VerbAtlas with lax agreement for frames and with superlax agreement for roles. When we look at the bigger but less comparable dataset in Tables 2 and 3, we see that agreement on frame labels from .71 to .76 for VerbAtlas (depending on the pair of annotators), but from .71 to .85 for Superframes even under strict agreement. On the other hand, agreement on role labels ranges from .75 to .81 for VerbAtlas but only from .63 to .79 for Superframes unders strict agreement, but from .73 to .86 under superlax agreement.

	Sentences	Predicates	Dependents
All annotated	69	98	190
VerbAtlas agreement		.67	.79
Superframes agreement (strict)		.62	.69
Superframes agreement (lax)		.71	.77
Superframes agreement (superlax)		.77	.81

Table 4: Average agreement between two annotators using VerbAtlas vs. Superframes on the same sentences. Sentences with consensus annotations are excluded.

We take these numbers to hint at general viability of Superframes, but of course cannot draw very strong conclusions from them. The Superframes dataset is much smaller and the intersection of both datasets is smaller still. Annotations were made by (partially) different people and under different conditions. The numer of Superframes used is much lower than that of VerbAtlas frames whereas the Superframes roles are more diverse due to VerbAtlas's reuse of role labels across different frames. VerbAtlas focuses on verbs and their arguments whereas all content words and their relations are annotated in Superframes. Perhaps most importantly, VerbAtlas agreement profits from the anchoring bias due to pre-annotations whereas Superframes was annotated from scratch.

We provide additional statistics in Tables 5, 6, 7, 8, 9, 10, 11, and 12.

Frame	Count
IDENTIFICATION	1946
CLASS	1060
MESSAGE	973
QUALITY	778
TIME	565
MODE	538
QUANTITY	407
PART-WHOLE	353
MESSAGE-INIT	322
LOCATION	258

Table 5: Most common frame labels in the Superframes annotation

Role	Count
m-scene	725
m-quality	642
m-time	523
m-mode	500
topic	476
x-experiencer	430
m-quantity	376
has-location	344
x-sender	310
participant	261

Table 6: Most common role labels in the Superframes annotation

Pair		Count
MESSAGE	MESSAGE-INIT	45
CLASS	IDENTIFICATION	40
CLASS	PART-WHOLE	34
IDENTIFICATION	NONCOMP	28
QUALITY	STATE	26

Table 7: Most common frame label disagreement pairs in the Superframes annotation

Pair		Count
m-quality	m-scene	35
m-scene	m-time	16
m-noncomp	x-noncomp	16
m-quality	m-state	14
target-content	topic	13

Table 8: Most common role label disagreement pairs in the Superframes annotation

Label	Count
QUALITY » NONCOMP	41
SOCIAL-RELATION » IDENTIFICATION	40
QUALITY-NEG » QUALITY	23
COMPARISON » QUALITY	15
RANK » TIME	15

Table 9: Most common figurative frame labels in the Superframes annotation

Label	Count
SOCIAL-RELATION || STATE	8
CLASS || LOCATION	7
STATE || QUALITY	5
STATE || SOCIAL-RELATION	5
MESSAGE || SEQUENCE	4

Table 10: Most common uncertain frame labels in the Superframes annotation

VerbAtlas frame	Superframe	Count
KNOW	MESSAGE	86
SPEAK	MESSAGE-INIT	77
SEEM	MESSAGE	75
SEE	MESSAGE	73
EXIST-WITH-FEATURE	SCENE	47
WRITE	MESSAGE-INIT	38
EXIST_LIVE	EXISTENCE	37
LIE	LOCATION	33
GO-FORWARD	LOCATION-INIT	32
EXIST_LIVE	SCENE	32

Table 11: Most common combinations of VerbAtlas and Superframes frames

VerbAtlas role	Superframes role	Count
Time	m-time	392
Theme	has-location	287
Agent	x-causer	257
Experiencer	x-experiencer	241
Agent	x-sender	227
Theme	topic	224
Agent	x-experiencer	145
Destination	target-location	139
Stimulus	topic	121
Theme	content	115

Table 12: Most common combinations of VerbAtlas and Superframes roles

5. Conclusions, Limitations, and Future Work

We have presented Superframes, a frame-semantic annotation scheme that aims to enable consistent annotation of predicates with frame labels and dependents with role labels across languages, achieved through a small number of frames, frame composition, aspect decomposition, and annotation atop Universal Dependencies. Non-local dependencies, figurativity, idiomaticity, and uncertainty are addressed through dedicated mechanisms. We have presented the results of a preliminary annotation study where Superframes is compared with a traditional annotation scheme with a larger number of frames. We were able to show higher agreement for Superframes under certain conditions, although it should be noted that the comparability is limited, especially due to the fact that annotators corrected pre-annotations for VerbAtlas but annotated from scratch for Superframes. Based on the insights from annotation discussion, we are currently in the process of revising the scheme and making the inventory of superframes more compact and more systematic, as well as preparing a larger-scale and more systematic annotation effort.

6. Acknowledgements

The author would like to thank the anonymous reviewer for important pointers. This work was funded by grant no. 560341082 (Superframes) of the German Research Foundation (DFG).

7. Bibliographical References

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating high quality proposition Banks for multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407, Beijing, China. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Tatiana Bladier, Kilian Evang, Valeria Generalova, Zahra Ghane, Laura Kallmeyer, Robin Möllemann, Natalia Moors, Rainer Osswald, and Simon Petitjean. 2022. RRGparbank: A parallel role and reference grammar treebank. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4833–4841, Marseille, France. European Language Resources Association.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. pages 85–112.

Cemil Cengiz and Deniz Yuret. 2020. Joint training with semantic role labeling for better generalization in natural language inference. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 78–88, Online. Association for Computational Linguistics.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2010. Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 52–60, Los Angeles, California. Association for Computational Linguistics.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 594–602, Sydney, Australia. Association for Computational Linguistics.

Simone Conia, Fabrizio Brignone, Davide Zanfardino, and Roberto Navigli. 2020. InVeRo: Making semantic role labeling accessible with intelligible verbs and roles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 77–84, Online. Association for Computational Linguistics.

Simone Conia, Riccardo Orlando, Fabrizio Brignone, Francesco Cecconi, and Roberto Navigli. 2021. InVeRo-XL: Making cross-lingual Semantic Role Labeling accessible with intelligible verbs and roles. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–328, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637, Hong Kong, China. Association for Computational Linguistics.

Kilian Evang. 2024. Superframes manual. Technical report, Heinrich Heine University Düsseldorf. https://github.com/texttheater/superframes/blob/main/doc/manual/manual.pdf, retrieved 2024-11-05.

Christiane Fellbaum, editor. 1998. *WordNet: An Eletronic Lexical Database*. MIT Press, Cambridge, MA.

Lydia Feng, Gregor Williamson, Han He, and Jinho D. Choi. 2023. Widely interpretable semantic representation: Frameless meaning representation for broader applicability. *ArXiv*, abs/2309.06460.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. 112:10336–10341.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Jiang Guo, Wanxiang Che, Haifeng Wang, Ting Liu, and Jun Xu. 2016. A unified architecture for semantic role labeling and relation classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1264–1274, Osaka, Japan. The COLING 2016 Organizing Committee.

Matías Guzmán Naranjo and Laura Becker. 2018. Quantitative word order typology with UD. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 91–104.

Jan Hajič, Eduard Bejček, Alevtina Bémová, Eva Buráňová, Eva Fučíková, Eva Hajičová, Jiří Havelka, Jaroslava Hlaváčová, Petr Homola, Pavel Ircing, Jiří Kárník, Václava Kettnerová, Natalia Klyueva, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, David Mareček, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Michal Novák, Petr Pajas, Jarmila Panevová, Nino Peterek, Lucie Poláková, Martin Popel, Jan Popelka, Jan Romportl, Magdaléna Rysová, Jiří Semecký, Petr Sgall, Johanka Spoustová, Milan Straka, Pavel Straňák, Pavlína Synková, Magda Ševčíková, Jana Šindlerová, Jan Štěpánek, Barbora Štěpánková, Josef Toman, Zdeňka Urešová, Barbora Vidová Hladká, Daniel Zeman, Šárka Zikánová, and Zdeněk Žabokrtský. 2024. Prague dependency treebank - consolidated 2.0 (PDT-c 2.0). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

Llio Humphreys, Guido Boella, Luigi Di Caro, Livio Robaldo, Leon van der Torre, Sepideh Ghanavati, and Robert Muthuri. 2020. Populating legal ontologies using semantic role labeling. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2157–2166, Marseille, France. European Language Resources Association.

Ishan Jindal, Alexandre Rademaker, Michał Ulewicz, Ha Linh, Huyen Nguyen, Khoi-Nguyen Tran, Huaiyu Zhu, and Yunyao Li. 2022. Universal Proposition Bank 2.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1700–1711, Marseille, France. European Language Resources Association.

Karin Kipper Schuler. 2005. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Daniil Larionov, Artem Shelmanov, Elena Chistova, and Ivan Smirnov. 2019. Semantic role labeling with pretrained language models for known and unknown predicates. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 619–628, Varna, Bulgaria. INCOMA Ltd.

Natalia Levshina. 2019. Token-based typology and word order entropy: A study based on universal dependencies. 23:533–572.

Ines Montani, Matthew Honnibal, Matthew Honnibal, Adriane Boyd, Sofie Van Landeghem, Henning Peters, Paul O'Leary McCann, Jim Geovedi, and others. 2023. explosion/spacy: v3.7.1: Bug fix for 'spacy.cli' module loading.

George Orwell. 2003. *1984*, 37th edition. Ullstein. German translation by Kurt Wagenseil (first published 1950 by Alfons Bürger Verlag).

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Steve Pepper. 2022. Hatcher-Bourque: Towards a reusable classification of semantic relations. In *Binominal Lexemes in Cross-Linguistic Perspective*, pages 303–354. De Gruyter Mouton.

Volha Petukhova and Harry Bunt. 2008. LIRICS semantic role annotation: Design and evaluation of a set of data categories. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O'gorman, James Gung, Kristin Wright-bettner, and Martha Palmer. 2022. PropBank comes of Age—Larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288, Seattle, Washington. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Valdi Rachman, Rahmad Mahendra, Alfan Farizki Wicaksono, Ahmad Rizqi Meydiarso, and Fariz Ikhwantri. 2018. Semantic role labeling in conversational chat using deep bi-directional long short-term memory networks with attention mechanism. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Tanja Samardžić, Ximena Gutierrez-Vasques, Rob van der Goot, Max Müller-Eberstein, Olga Pelloni, and Barbara Plank. 2022. On language spaces, scales and cross-lingual transfer of UD parsers. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 266–281, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. Comprehensive supersense disambiguation of English prepositions and possessives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 185–196, Melbourne, Australia. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Stephen Tratz and Eduard Hovy. 2010. A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 678–687, Uppsala, Sweden. Association for Computational Linguistics.

Richard Tzong-Han Tsai, Wen-Chi Chou, Yu-Chun Lin, Cheng-Lung Sung, Wei Ku, Ying-Shan Su, Ting-Yi Sung, and Wen-Lian Hsu. 2006. BIOSMILE: Adapting semantic role labeling for biomedical verbs:. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 57–64, New York, New York. Association for Computational Linguistics.

Zdeňka Urešová, Cristina Fernández Alcaina, Peter Bourgonje, Eva Fučíková, Jan Hajič, Eva Hajičová, Veronika Kolářová, Georg Rehm, Kateřina Rysová, and Karolina Zaczynska. 2025. SynSemClass 5.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Zdeňka Urešová, Karolina Zaczynska, Peter Bourgonje, Eva Fučíková, Georg Rehm, and Jan Hajic. 2022. Making a semantic event-type ontology multilingual. In *Proceedings of the Thir-*

teenth Language Resources and Evaluation Conference, pages 1332–1343, Marseille, France. European Language Resources Association.

Shijie Wu, Benjamin Van Durme, and Mark Dredze. 2022. Zero-shot cross-lingual transfer is underspecified optimization. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 236–248, Dublin, Ireland. Association for Computational Linguistics.

Kun Xu, Haochen Tan, Linfeng Song, Han Wu, Haisong Zhang, Linqi Song, and Dong Yu. 2020. Semantic Role Labeling Guided Multi-turn Dialogue ReWriter. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6632–6639, Online. Association for Computational Linguistics.

Bo Zhang, Yue Zhang, Rui Wang, Zhenghua Li, and Min Zhang. 2020. Syntax-aware opinion role labeling with dependency graph convolutional networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3249–3258, Online. Association for Computational Linguistics.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. Transfer learning from semantic role labeling to event argument extraction with template-based slot querying. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2627–2647, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

First Shared Task on UMR Parsing

Jan Štěpánek*, Daniel Zeman*, Markéta Lopatková*
Federica Gamba*, Hana Hledíková*, Nianwen Xue†
*Charles University, Faculty of Mathematics and Physics, ÚFAL
Prague, Czechia
{stepanek, zeman, lopatkova, gamba, hledikova}@ufal.mff.cuni.cz

†Brandeis University
Waltham, MA, USA
xuen@brandeis.edu

Abstract

The paper presents the first shared task on parsing Uniform Meaning Representation (UMR), a graph-based framework for cross-linguistic semantic annotation of typologically diverse languages. The task requires systems to enrich plain text with sentence-level structure, node–token alignment, and document-level relations. It involves processing data for seven languages from four language families (Indo-European, Sino-Tibetan, Na-Dene, and Algic). Six languages have at least some training data; for one language, data is not available, leading to a zero-shot scenario. The training dataset as well as the gold-standard test set for all seven languages is released and made available for follow-up research. We present the task setup and evaluation methodology, using two graph matching approaches – a traditional, and an alignment-sensitive one, tailored specifically for UMR. Two participating systems are compared, each representing different modeling approaches. Results highlight the challenges of UMR parsing, particularly for alignment prediction and document-level semantics, and reveal substantial variation across languages and annotation conditions.

Keywords: uniform meaning representation, parsing, evaluation, shared task

1. Uniform Meaning Representation

UMR is a graph-based framework that is semantically grounded and designed for cross-linguistic applicability (Van Gysel et al., 2021; Bonn et al., 2024). Based on Abstract Meaning Representation (AMR, Banarescu et al., 2013), it abstracts from surface syntax to represent concepts—such as entities and events—as graph nodes. The relationships between these concepts are captured as graph edges, and their attributes are included in a normalized, language-independent format. Notably, all syntactic variations of a statement are represented uniformly within this framework. These graphs constitute the *sentence-level annotation*.

In addition, UMR provides a comprehensive annotation of epistemic modality, and marks temporal and coreference relations (both intra- and inter-sentence); this forms the *document-level representation*.

Furthermore, to support automatic processing of the data, the graph nodes are aligned with surface tokens; this *node-to-token alignment* forms an additional annotation block for each sentence. Figure 1 shows an example of UMR annotation with alignment and document-level relations.

The goal of the shared task on UMR parsing is to attract researchers to focus on automatic prediction of UMR annotation across typologically diverse languages, which so far remains largely unexplored.

This paper presents the shared task data (Sect. 2) and briefly describes the task settings (Sect. 3). Section 4 first discusses the metrics used to evaluate the competing systems (Sect. 4.1) and then provides a summary of these systems (Sect. 4.2). Finally, the systems' performance is compared (Sect. 4.3). Section 5 summarizes the task findings, highlighting key challenges and outlining directions for future research.

1.1. Previous Work

While recently developed UMR datasets have become available for several languages, providing an essential foundation for cross-linguistic semantic analysis and modeling, the task of automatically parsing raw text into UMR structures remains relatively underexplored, with only limited efforts dedicated to building robust, generalizable UMR parsers.

The first published UMR parsing model for English was introduced by Chun and Xue (2024). This approach uses a pipeline that leverages existing AMR parsers to generate AMR structures, which are then converted into UMR sentence-level graphs using linguistically motivated heuristics. Document-level annotation is learned inde-

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 136–147
11 May 2026.

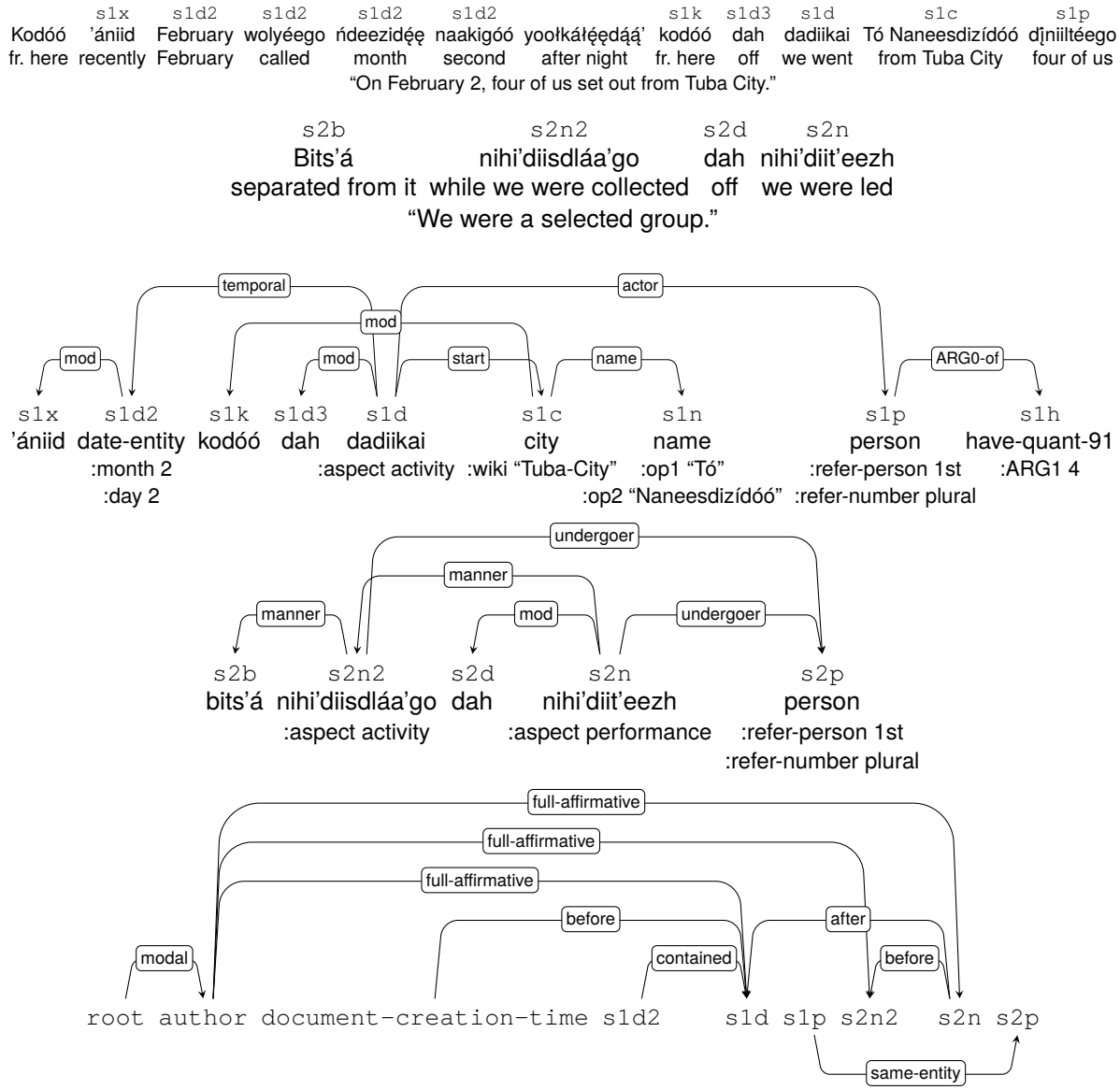


Figure 1: Example of two sentences from the Navajo test data with node–token alignments, sentence-level graphs, and document-level relations.

pendently from the sentence graphs. It is modeled as a set of triples that consist of relevant sentence tokens and their specific relations. The authors developed separately trained models for each of the three tasks: modal relations, temporal relations, and coreference. This approach helps to address the scarcity of available UMR data at that time. Finally, the tokens in the document-level triples are aligned with the corresponding nodes in the sentence graphs.

Inspired by this approach, Markle et al. (2026) present two methods for English text-to-UMR parsing. The first method also utilizes several existing text-to-AMR parsers. These parsers are directly fine-tuned on UMR data. The second method involves first creating UD trees and then converting these into partial UMR structures (Gamba et al., 2025). Subsequently, a T5 Transformer model is trained to expand these partial graphs into complete UMRs.

Finally, Sun et al. (2024) investigate the performance of a GPT-4 model in generating draft sentence-level UMR structures for Chinese. They use few-shot learning and Think-Aloud prompting to guide GPT-4 to generate UMR sentence-level graphs, then compare the results with manually annotated data. The findings are promising, indicating the accuracy of the GPT-generated UMRs is approximately 10-20 % lower than the inter-annotator agreement.

	Training data						Testing data		
	Clean			Dirty					
	Files	Sents.	Tokens	Files	Sents.	Tokens	Files	Sents.	Tokens
Arapaho	1	53	274	2	292	1,627	2	55	274
Chinese	20	557	15,375	1	1,435	18,656	15	236	6,467
Czech	6	103	1,587	6,516	159,906	2,537,317	5	220	4,048
English	4	180	2,023	570	29,872	292,023	5	195	4,092
Italian	—	—	—	—	—	—	1	100	2,212
Latin	—	—	—	3	1,049	19,139	1	50	889
Navajo	1	5	60	3	337	2,663	1	163	1,194

Table 1: Statistics of the data used in the shared task.

2. The Data

The shared task involves processing data for seven languages. For six of them, at least some training data is available; for one language (Italian), no training data is available, leading to a zero-shot scenario. The overall data statistics are presented in Table 1.

The shared task data is now available in the Lindat repository as the UMR release 2.2 (Bonn et al., 2026).[1]

2.1. Training Data

There are six languages for which training data is available: Arapaho, Chinese, Czech, English, Latin, and Navajo. For Arapaho, Chinese, English, and Navajo, the training data is primarily based on UMR 2.0 (Bonn et al., 2025), though not identical: it has been modified to align with the (newly published) UMR format specification (Section 2.2.4) when necessary (for example, typos end errors in bracketing were fixed, and some empty lines, trailing whitespace, or references to non-existent nodes were removed). The training data for Czech and Latin is based on UMR 2.1 (Štěpánek et al., 2025), however, it includes some improvements (the automatic conversion procedure was improved to replace most of the "substitute" lemmata from PDT by corresponding UMR constructs). For Italian, no training data is available at all.

Two types of data available for training are distinguished, "clean" and "dirty":

Clean data is reasonably similar to gold-standard test data, but they are typically very small (if they exits at all). In majority, they contain all annotation parts and should better conform to the annotation guidelines.

Dirty data is much larger, especially for Czech and English. However, it is imperfect or incomplete in various aspects, e.g., it lacks some or all document-level annotation, sometimes also the node–token alignment is missing. There may be additional relations or concepts not defined in UMR.[2]

All training data was freely available during the shared task, without the need to register or sign a contract.

2.2. Test Data

For Arapaho, English and Navajo, the cleaned part of UMR 2.0 data was split approximately to halves and designated as clean training data and test data, respectively.[3] For Latin, the single manually annotated file from UMR 2.0 was used as test set. For Chinese, 25 clean files were available in UMR 2.0. The first 20 were designated as clean training data, the remaining 5 files were combined with 10 previously unpublished files (see below) to become the test data. For Czech, no files released

[1] http://hdl.handle.net/11234/1-6132
(Besides training and test data from the shared task, the release also includes system outputs, as well as two languages that were not part of the shared task but were previously released in UMR: Sanapana and Kukama.)

[2] In English, manual AMR annotation has been partially converted to UMR. Word alignment and document-level relations are missing. In Chinese, one large file lacks document-level relations. Czech and Latin are conversions from the t-layer of Prague Dependency Treebank, resp., Latin Dependency Treebank. In Arapaho and Navajo, the "dirty" files are simply incomplete manual UMR annotations, missing document-level relations and sometimes also word alignment.

[3] Two files in the English test set, `en-0005.umr` and `en-0007.umr`, turned out to contain identical sentences, although their UMR annotation was not identical, with similarity score just below 82%. This is an undocumented feature of UMR 2.0. As we only became aware of it during the test phase of the shared task, we decided to leave the test data as it was.

in UMR 2.0 and 2.1 were used in the shared task test set.

2.2.1. Data Annotated for the Shared Task

New Chinese data. The Chinese UMR test set comprises five documents drawn from the UMR 2.0 release, supplemented by ten newly annotated documents created specifically for this shared task and not previously released. Consistent with the existing Chinese UMR corpus, all ten new documents are sourced from Wikinews, ensuring stylistic and domain continuity. The annotation process follows a semi-automatic pipeline: each document is first processed using an LLM-based UMR parser (Sun et al., 2024), after which trained annotators perform thorough manual correction and validation to ensure high-quality semantic representations. All documents in the test set include comprehensive sentence-level UMR annotations as well as document-level structures, capturing cross-sentence phenomena such as coreference, temporal relations, and modal dependencies. This design ensures that the dataset supports rigorous evaluation of both local semantic parsing and discourse-level understanding.

New Czech data. Four files with manual UMR annotation have been prepared specifically for the shared task:

- general journalistic genre texts, comprising two original Czech documents (newspaper texts from 1992-1994) and one translated document (Czech translations of one Penn Treebank-WSJ text);
- spoken data (a part of a testimony, originally recorded for the Shoah Memory project).

The annotations include sentence-level graphs, node–token alignment, and a partial document-level representation (limited to coreferential relations, both within and across sentences).

2.2.2. Data from PUD

Finally, the test set was extended with 100 parallel sentences in Czech, English, and Italian, originating in the PUD treebank (Zeman et al., 2017) (genre-wise, they are split 50:50 between online news and Wikipedia). These sentences were manually annotated with UMR in order to evaluate UD-to-UMR conversion (Gamba et al., 2025) and they were not included in previous UMR releases.

2.2.3. Differences in UMR Annotation

Due to the complexity of UMR annotation, not everything is annotated in all the test files to the same extent. Some specifics were already mentioned above; here we summarize them:

- All test sets include node–token alignment.
- The PUD datasets do not have document-level relations. All non-PUD test sets have document-level annotation, but in Czech it is limited to coreference.
- The `:modal-strength` sentence-level relation is annotated in datasets that do not have document-level modal relations, that is in all Czech files and in the PUD files of English and Italian.
- The English PUD file lacks the `:wiki` attribute of named entities. Other English files and other PUD files have this attribute, but it is also missing in Arapaho, Chinese, and non-PUD Czech. English and Navajo data use article titles from English Wikipedia as the value of `:wiki`; Czech, Italian, and Latin use the more portable WikiData identifiers.
- The relations `:actor`, `:undergoer`, `:recipient`, `:experiencer`, `:stimulus`, and `:theme` are only used in so-called Stage 0 annotation of argument roles, i.e., only in Arapaho and Navajo and in the three PUD files. The other test sets use numbered `:ARG`N roles instead. (Note that numbered arguments occur even in Stage 0 annotation because they are used with predefined abstract predicates.)

The shared task evaluation is configured so that systems are not penalized for predicting an attribute or relation that is omitted in the gold data.

2.2.4. UMR File Format

The training data in all languages use the same `.umr` file format, which is a text-based format where each sentence is organized into four annotation blocks, viz. metadata, sentence graph, alignment, and document relations. A similar format was used in previous UMR releases, but without any public formal specification (and more importantly, without validating basic requirements, such as matching brackets in graph encoding). Therefore, we published the format specification,[4] as well as a Python validation script[5] that the participants could use to validate their system output before submitting it.

3. The Task

The participants were given blind test data as the input for their systems. The input text contained no UMR annotation but it was tokenized and segmented to sentences; systems were required to preserve tokenization and segmentation, as this

[4] https://ufal.mff.cuni.cz/umr-parsing/umr-file-format

[5] https://github.com/ufal/umrtools

was necessary for evaluation of the output. Systems were expected to generate sentence-level UMR graphs with nodes aligned to input tokens where appropriate, as well as document-level relations. Omitting some parts of the annotation (e.g., some document-level relations) would be possible but it would be penalized by lower scores. Participants were specifically instructed that node–token alignment is integral part of the annotation and that it plays a crucial role in the evaluation procedure.

Training data was made available at the beginning of the shared task, clearly distinguishing the "clean" and "dirty" subsets (see Section 2.1). Participants had to figure themselves the difference between clean an dirty annotation in each language, and they were given no guidance about what to do with Latin (no clean training data) and Italian (no training data at all). They were not informed which relations and attributes are omitted in individual gold files, only our evaluation procedure was informed not to penalize them for predicting such relations.[6] No restrictions were placed on using external language resources, only the previous UMR releases had to be excluded, as they contain a subset of the data that we now use for testing. We are aware that this subset may have been seen by large language models; nevertheless, given the complexity of the task and scarcity of clean annotated data, we decided to take the risk and allow using pretrained models.

We launched a dedicated virtual machine where system outputs were submitted and immediately evaluated, with no limits on the number of submissions per team. The submission site remains open and can be used to evaluate other systems on the same data, using the same evaluation metric.[7]

The time available for the task was very short, with slightly less than four weeks between releasing the training data and collecting the system outputs; blind test data were made available approximately in the middle of this period.

4. Evaluation

4.1. Evaluation Metrics

Standard approaches to evaluation of semantic graphs consist of two phases:

1. Find mapping between nodes of the corresponding graphs;
2. Using the mapping, compute F_1-score of triples of the following types:
   - parent node – relation – child node[8]
   - node – attribute – value[9]

We use two metrics with different node mapping algorithms. Our main metric is *ju:mætʃ* (Zeman and Gamba, 2026), which takes node–token alignment as the main factor influencing the node-node mapping. For comparison purposes, we also compute *smatch* (Cai and Knight, 2013), although its public implementation[10] can only compare sentence-level graphs. The *ju:mætʃ* evaluation script was publicly available during the shared task.[11]

Smatch will map as many nodes as possible. If one of the graphs has more nodes than the other, remaining nodes will stay unmapped. If the graphs have the same number of nodes, every node will be mapped to a node in the other graph, even if they are clearly unrelated. This may occasionally improve the score when a random attribute occurs in both nodes, but it blurs the interpretation of the score, and any (dis)agreement in attribute values of such nodes is meaningless. Consider the two graphs in Figure 2. Identical concepts and attributes will lead to mapping `x3a-y3a`, `x3i-y3i`, `x3c2-y3c`, `x3s-y3s`. Likewise, `x3p-y3p` will be mapped, despite the mismatch in `:refer-number`. All these mappings are intuitively correct; but we cannot say the same about the remaining nodes. None of `x3u` and `x3u2` will be mapped to `y3u`, which was successfully lemmatized to *utor*, yet the expected concept is utor-03. The values of `:aspect` do not match either, so the only positive point would be the `:ARG0` relation to the person node. This is not enough to enforce the mapping `x3u2-y3u`, as *smatch* also considers whether a node is the 'top node' (root) of the graph, and the 'top' attribute gives a point to `x3c-y3u`. Moreover, the latter mapping allows to earn another point by following the `:ARG1` relation from the top node and mapping `x3u-y3a`, although these nodes are also semantically unrelated.

In contrast, *ju:mætʃ* primarily maps nodes aligned to the same word(s), and for nodes without word alignment (assumed to be a minority in UMR graphs) it requires concept identity. As word alignment is part of the annotation, the metric evaluates

[6] Nevertheless, we should have told the participants that the Czech and English PUD files differed from other Czech resp. English files in the relations they use for predicate-argument structure, and we failed to do so. This is definitely a lesson for future instances of this task.

[7] https://ufal.mff.cuni.cz/umr-parsing/submission

[8] This includes document-level relations in UMR, although they may connect nodes from different sentences, or even special fixed nodes such as `author` or `document-creation-time`.

[9] In the context of UMR, the link between a node and its concept string is a special kind of attribute-value pair.

[10] https://github.com/snowblink14/smatch

[11] Now it is available together with the validator in https://github.com/ufal/umrtools.

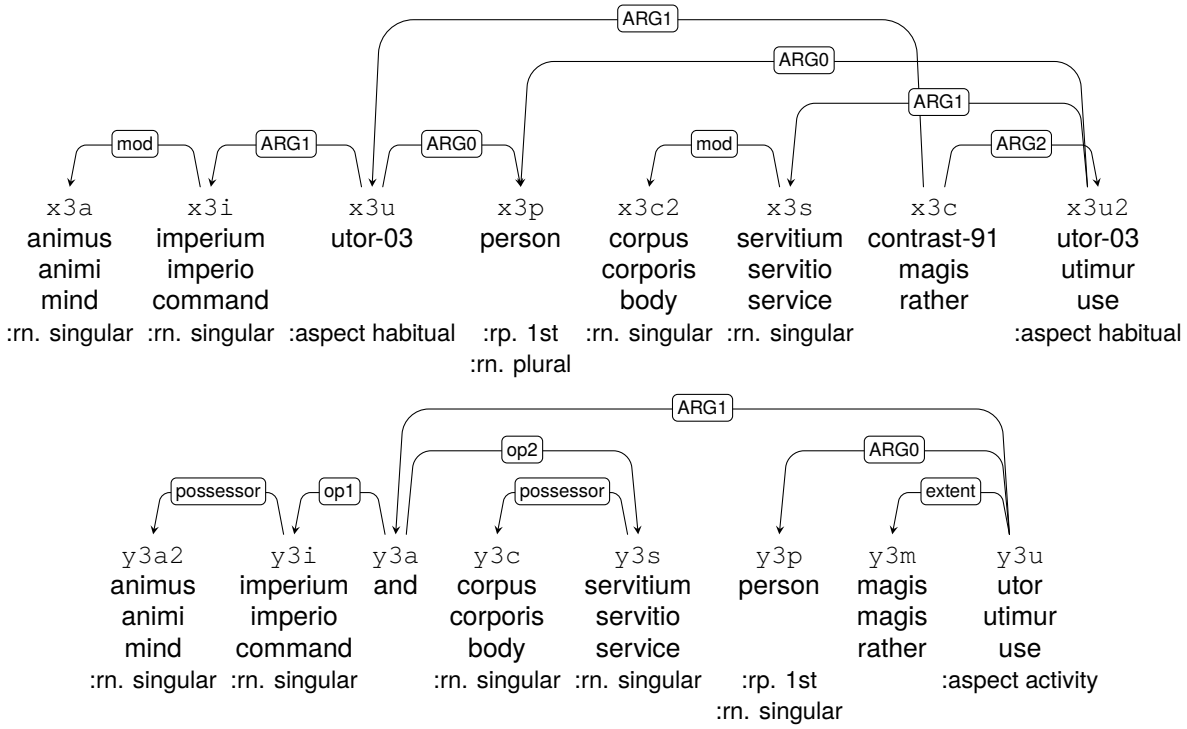


Figure 2: Competing annotations of Latin *animi imperio, corporis servitio magis utimur* "We use the command of the mind rather than the service of the body." In this case, each node in the two graphs is aligned to at most one token and the aligned tokens are shown below node ids and concept strings (lemmas). Abbreviated attributes ":rn." = ":refer-number", ":rp." = ":refer-person".

it, too, though only indirectly. Missing or nonsensical alignments will lead to suboptimal node mapping and thus lower scores. On the other hand, using imperfect alignment for node matching is less straightforward than it may seem. A node may be aligned to multiple words;[12] alignments of nodes from two annotations of the same sentence may overlap instead of being identical. Overlapping alignments have to be resolved so that at most one node from either side is retained. We need symmetric one-to-one node mappings—not only because it simplifies subsequent comparison of node properties, but also because it follows the intuition that both nodes were intended to represent the same concept in the meaning structure of the sentence.

The symmetrization works as follows. Whenever a node on either side is mapped to multiple nodes on the other side, mappings are gradually removed until just one target node remains. When removing target nodes, we try to remove those that are least similar to the source node, according to several criteria. The most important criterion is concept string identity. If it does not lead to unique node mapping, we also consider all attributes of the node and their values (outgoing relations are treated analogously). Next, we also do a "weak" comparison of attributes, where we only consider the presence of an attribute, without requiring identical value of the attribute on both sides. This can help distinguish e.g. eventive concepts (having attributes such as `:aspect`) from entities (having e.g. `:refer-number` or `:name`). Finally, we also prioritize alignment to longer words (trying to avoid relying too much on function words, which tend to be shorter).

Nodes without word alignment are paired when they have the same concept; if there are multiple options, we use the same symmetrization approach as described above. The assumption is that many of them are abstract UMR concepts such as `identity-91`, `have-mod-91` or `name`, hence comparing their concept strings will often point at the correct mapping.[13]

When applied to the example in Figure 2, *ju:mætʃ* will trivially discover the mappings that *smatch* got right: `x3a-y3a`, `x3i-y3i`, `x3c2-y3c`, `x3s-y3s`, `x3p-y3p`. It will also map `x3u2-y3u`

[12]For example, in English *He wants to go*, the node representing the going event may be aligned just to the verb *go*, or to the two-word segment *to go*.

[13]Specifically for the `name` concept, we enhance it with the real name from its `:opN` attributes before comparing it to other nodes, e.g., `name["United" "States"]`. This helps in sentences with multiple named entities.

because both are aligned to the verb form *utimur* "we use"; and `x3c-y3m` because both correspond to *magis* "rather". The nodes `x3u` and `y3a` will stay unaligned.

4.2. Summary of Competing Systems

Twelve people expressed interest through our registration form. In the end, the shared task received two submissions, labeled as "orange" (Heinecke and Asadullah, 2026) and "sema" (de Vergnette and Amblard, 2026).

The team "orange" used a two step approach for intra-sentence and inter-sentence information, but merging the document-level annotation with the UMR graph where the relations didn't reach to a different sentence. For the first step, they fine-tuned 3 different models (monolingual Flan-T5 and mT5, and multilingual mT5) and selected the best for each language based on the development data. For the second step, the team used Qwen3 8B with sentence pairs not farther apart than 6, which seemed to cover most of the inter-sentence relations.

The team "sema" used parameter-efficient fine-tuning of a small LLM (Qwen 4B) in three stages:

1. Training on the sentence-level graphs only on dirty data (with a limit of sample per language not to invisibilize less endowed languages);
2. Training on the sentence-level graphs on clean data;
3. Further training the system to output alignments, of the form

```
Word1: node1 (or nothing)
Word2: ...
```

   This way, doing word-to-token and not token-to-word alignment, the format was less ambiguous (avoiding node ordering issues) and more easily predictable.

4.3. Results

Table 2 presents the main evaluation with *ju:mætʃ* scores for each language and system, as well as macro-averages over languages.[14] Since the Stage 0 annotation of the PUD data differs from the rest, we also present separate scores for PUD files (Table 3) and non-PUD files (Table 4). Czech and English are the two languages which occur in both tables; for the Orange system, the PUD files were clearly more difficult, but for Sema the results are mixed: PUD is better than non-PUD in Czech, but the opposite holds in English.

Language	Orange	Sema
Arapaho	0.0815	**0.1115**
Chinese	**0.3651**	0.2585
Czech	0.1587	**0.2652**
English	**0.2219**	0.1919
Italian	0.1394	**0.2137**
Latin	0.1724	**0.1918**
Navajo	**0.2155**	0.1273
Average	0.1935	**0.1943**

Table 2: Main *ju:mætʃ* scores per language and system.

Language	Orange	Sema
Czech	0.1445	**0.2764**
English	0.1590	**0.1641**
Italian	0.1394	**0.2137**
Average	0.1476	**0.2181**

Table 3: Separate *ju:mætʃ* for PUD data.

Language	Orange	Sema
Arapaho	0.0815	**0.1115**
Chinese	**0.3651**	0.2585
Czech	0.1685	**0.2573**
English	**0.2734**	0.2146
Latin	0.1724	**0.1918**
Navajo	**0.2155**	0.1273
Average	**0.2127**	0.1935

Table 4: Separate *ju:mætʃ* for non-PUD data.

Table 5 shows evaluation using the *smatch* score. Here the overall numbers are higher for two main reasons: 1. unlike *ju:mætʃ*, *smatch* does not use word alignment to restrict node mapping between the system output and gold standard; 2. *smatch* does not evaluate document-level relations, which proved difficult and were only partially predicted by the participating systems (see also Table 13). At the same time, *smatch* considers a special relation marking the "top node" (root) of the graph, which is typically easy to predict and which was not included in *ju:mætʃ* by default. To compensate for the second point, Table 6 shows modified *ju:mætʃ* for sentence graphs with top nodes.

Since *ju:mætʃ* heavily depends on the system's ability to predict node-word alignment, we also tried to shed some light on that factor. In Table 7, we evaluate word alignment independently of node

[14] The outputs of the two systems are released together with the shared task training and gold standard test data in UMR 2.2 (Bonn et al., 2026).

Language	Orange	Sema
Arapaho	**0.4308**	0.3447
Chinese	**0.5194**	0.4363
Czech	**0.4921**	0.4456
English	**0.4625**	0.4169
Italian	0.3567	**0.3796**
Latin	**0.4446**	0.3879
Navajo	**0.3754**	0.2568
Average	**0.4402**	0.3811

Table 5: *Smatch* scores per language and system.

Language	Orange	Sema
Arapaho	0.1008	**0.1272**
Chinese	**0.3777**	0.2471
Czech	0.1582	**0.2724**
English	**0.2509**	0.2057
Italian	0.1368	**0.2155**
Latin	**0.1950**	0.1846
Navajo	**0.2551**	0.1267
Average	**0.2107**	0.1970

Table 6: *Ju:mætʃ* scores modified to be more comparable with *smatch*, i.e., disregarding document-level relations but counting a special relation for top nodes.

Language	Orange	Sema
Arapaho	0.4078	**0.7368**
Chinese	**0.7074**	0.6122
Czech	0.4308	**0.4957**
English	**0.7148**	0.6457
Italian	0.1873	**0.2539**
Latin	0.5114	**0.5493**
Navajo	**0.4818**	0.2611
Average	0.4916	**0.5078**

Table 7: F_1 score of tokens aligned to a node.

mapping. We consider sets of tokens corresponding to a node in each file, regardless of whether the node mapping algorithm actually mapped such nodes to each other (although it is likely that it did). For example, if there is the phrase *to the city* in the English data, and all three words are included in the gold standard alignment of the same node, we expect the system output to also align all three words to one node. If the system predicts a node representing the `city` concept and decides to align it only to *city* (leaving the preposition and the article unaligned), Table 7 will not count it as matching alignment, although it is still

Language	Orange	Sema
Arapaho	0.2517	**0.5372**
Chinese	**0.6046**	0.4521
Czech	0.3933	**0.5488**
English	**0.5643**	0.4856
Italian	0.3766	**0.5194**
Latin	0.4289	**0.4700**
Navajo	**0.5309**	0.3514
Average	0.4500	**0.4806**

Table 8: Proportion of mapped nodes computed as $F_1 = 2PR/(P + R)$, where R is the number of gold nodes mapped to system nodes, divided by the total number of gold nodes, and P is the number of system nodes mapped to gold nodes, divided by the total number of system nodes.

Language	Orange	Sema
Arapaho	**0.5570**	0.1742
Chinese	**0.5965**	0.5208
Czech	**0.5210**	0.4872
English	**0.4198**	0.4071
Italian	0.4044	**0.4116**
Latin	**0.4545**	0.4289
Navajo	**0.4641**	0.3568
Average	**0.4882**	0.3981

Table 9: *Ju:mætʃ* scores ignoring triples governed by unmapped nodes.

possible that the node mapping algorithm will map the concept nodes correctly. On the other hand, Table 8 gives the proportion of nodes for which a corresponding node was found in the other file. Note that these scores indicate success in finding *some* mapping; they do not say anything about quality of the mapping. Finally, Table 9 shows the *ju:mætʃ* scores computed solely over triples where the governing nodes were successfully mapped. When contrasted with Table 2, Orange now looks much better than Sema; however, one has to remember that Orange was less successful in reproducing word alignment, thus excluding more potentially difficult nodes from the comparison.

We also offer separate evaluation of a few selected attributes or relations. These are partial *ju:mætʃ* F_1 scores; for nodes that do not have a counterpart in the cross-file node mapping, values of all attributes are automatically wrong. Table 10 shows evaluation of concept prediction, Table 11 evaluates all numbered argument relations (`:ARG`N and `:ARG`N`-of`). It should be noted again that some of the test documents contain Stage 0 UMR annotation, hence they do not use

Language	Orange	Sema
Arapaho	**0.2119**	0.1354
Chinese	**0.5287**	0.3503
Czech	0.2724	**0.3091**
English	**0.3866**	0.2957
Italian	0.2110	**0.2653**
Latin	**0.2541**	0.2337
Navajo	**0.4287**	0.2099
Average	**0.3276**	0.2571

Table 10: Concept *ju:mæʧ*.

Language	Orange	Sema
Arapaho	0.0000	0.0000
Chinese	**0.2270**	0.1369
Czech	0.0231	**0.1010**
English	**0.1490**	0.0777
Italian	0.0089	**0.0113**
Latin	0.0538	**0.0581**
Navajo	0.0000	0.0000
Average	**0.0660**	0.0550

Table 11: *Ju:mæʧ* of `:ARG`N and `:ARG`N`-of` relations.

Language	Orange	Sema
Arapaho	0.0764	**0.2192**
Chinese	0.2828	**0.3153**
Czech	0.0047	**0.3554**
English	0.0031	**0.3072**
Italian	0.0645	**0.2741**
Latin	0.1576	**0.2042**
Navajo	**0.1002**	0.0470
Average	0.0985	**0.2460**

Table 12: *Ju:mæʧ* of `:aspect`.

Language	Orange	Sema
Arapaho	0.0000	**0.3596**
Chinese	0.4261	**0.4634**
English	0.0000	**0.1386**
Latin	0.0000	**0.3236**
Navajo	0.0000	**0.2062**
Average	0.0852	**0.2983**

Table 13: *Ju:mæʧ* of document-level modal relations (not available in Czech and Italian).

such relations with normal verbs; however, all languages may have such relations under abstract

Language	Orange	Sema
Czech	0.0000	**0.3864**
English	0.0000	**0.0195**
Italian	0.0000	**0.0653**
Navajo	**0.0316**	0.0000
Average	0.0079	**0.1178**

Table 14: *Ju:mæʧ* of `:modal-strength` (sentence-level modal annotation, not available in Arapaho, Chinese, and Latin).

predicates such as `have-quant-91`. Table 12 evaluates the `:aspect` attribute of events.

Document-level relations are split to three categories: modal, temporal, and coreferential. None of the two systems scored in coreference, and the only non-zero score for temporal relations was achieved by Orange on the Chinese data ($F_1 = 0.0911$). For modal relations, the scores are presented in Table 13. They do not include Czech and Italian because the gold data in these languages use Stage 0 approach to modal annotation (although Czech is Stage 1 in other aspects, such as argument roles). Instead, these two languages employ the `:modal-strength` attribute in sentence-level graphs (Table 14). The same approach is also taken in one English file (the PUD data, parallel in Czech, English, and Italian). The Navajo dataset sticks out, as it contains both document-level and sentence-level modal relations (the latter partially recovered by the Orange system).

5. Conclusion

We have introduced the first shared task on UMR parsing, establishing a benchmark for evaluating multilingual semantic graph generation. The goal of the shared task is to assess how effectively current systems can construct structured meaning representations that capture the underlying semantics of sentences across diverse languages. The evaluation results demonstrate that UMR parsing remains a challenging problem, particularly with respect to node–token alignment and document-level relations such as modality and temporal structure. While participating systems achieved partial success with sentence-level representations, the overall performance scores indicate significant room for improvement, especially in low-resource and zero-shot settings.

The findings further emphasize the importance of data quality and the completeness of annotations. Performance varied notably across different languages and datasets, suggesting that inconsistencies, incomplete annotations, or language-

specific phenomena can dramatically affect parser effectiveness. These observations highlight the need for more robust modeling approaches and richer annotated resources.

Overall, these findings reveal that significant obstacles remain in UMR parsing, both in terms of model architecture and data availability. Future research should prioritize improving alignment prediction mechanisms, advancing systems' treatment of document-level structures, and leveraging multilingual transfer learning to enhance performance in low-resource contexts.

By establishing this shared task, we hope to stimulate further research in UMR parsing and support the broader goal of building scalable, cross-linguistic meaning representation systems that can facilitate a deeper understanding of natural language semantics across diverse languages and domains.

6. Acknowledgments

The work described herein has been supported by the grants *LINDAT/CLARIAH-CZ* (Project No. LM2023062) of the Ministry of Education, Youth, and Sports of the Czech Republic, and GAUK No. 104924 of the Charles University.

The project has been using data and tools provided by the *LINDAT/CLARIAH-CZ Research Infrastructure* (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

This work is also supported in part by grants from the CNS Division of National Science Foundation (Awards no: NSF_2213804) entitled "Building a Broad Infrastructure for Uniform Meaning Representations". Any opinions, findings, conclusions or recommendations expressed in this material do not necessarily reflect the views of NSF.

7. Bibliographical References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E. L. Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italia. ELRA and ICCL.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Jayeol Chun and Nianwen Xue. 2024. Uniform Meaning Representation Parsing as a Pipelined Approach. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 40–52, Bangkok, Thailand. Association for Computational Linguistics.

Rémi de Vergnette and Maxime Amblard. 2026. Sema system for the DMR 2026 shared task: Multistage UMR parsing with Qwen3-4B. In *Proceedings of the Seventh International Workshop on Designing Meaning Representations*, Palma, Spain. ELRA.

Federica Gamba, Alexis Palmer, and Daniel Zeman. 2025. Bootstrapping UMRs from Universal Dependencies for Scalable Multilingual Annotation. In *Proceedings of the 19th Linguistic Annotation Workshop ((LAW-XIX-2025)*, pages 126–136, Wien, Austria. Association for Computational Linguistics.

Johannes Heinecke and Munshi Asadullah. 2026. Orange @ UMR parsing shared task. In *Proceedings of the Seventh International Workshop on Designing Meaning Representations*, Palma, Spain. ELRA.

Emma Markle, Javier Gutierrez Bach, and Shira Wein. 2026. SETUP: Sentence-level English-To-Uniform Meaning Representation Parser. In *Proceedings of the 2026 International Conference on Language Resources and Evaluation (LREC 2026)*, Palma, Spain. ELRA.

Haibo Sun, Nianwen Xue, Jin Zhao, Liulu Yue, Yao Sun, Keer Xu, and Jiawei Wu. 2024. Chinese UMR annotation: Can LLMs help? In *Proceedings of the Fifth International Workshop on Designing Meaning Representations @ LREC-COLING 2024*, pages 131–139, Torino, Italia. ELRA and ICCL.

Jens E. L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a Uniform Meaning Representation for Natural Language Processing. *Künstliche Intelligenz*, 35(3):343–360.

Daniel Zeman and Federica Gamba. 2026. Word alignment-based evaluation of Uniform Meaning Representations. In *arXiv:2603.26401 [cs.CL]*. arXiv.org.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

8. Language Resource References

Julia Bonn and Claire Bonial and Matt Buchholz and Hsiao-Jung Cheng and Alvin Chen and Ching-wen Chen and Andrew Cowell and William Croft and Lukas Denk and Ahmed Elsayed and Eva Fučíková and Federica Gamba and Carlos Gomez and Jan Hajič and Eva Hajičová and Jiří Havelka and Loden Havenmeier and Ath Kilgore and Veronika Kolářová and Lucie Kučová and Kenneth Lai and Bin Li and Jingyi Li and Markéta Lopatková and Marie MacGregor and Marie Mikulová and Jiří Mírovský and Anna Nedoluzhko and Skatje Myers and Michal Novák and Tim O'Gorman and Petr Pajas and Alexis Palmer and Nartha Palmer and Jarmila Panevová and Benét Post and James Pustejovsky and Petr Sgall and Jialin Song and Li Song and Magda Ševčíková and Jan Štěpánek and Zdeňka Urešová and Haibo Sun and Yao Sun and Rosa Vallejos Yopán and Jens Van Gysel and Meagan Vigus and Kristin Wright-Bettner and Jiawei Wu and Nianwen Xue and Dan Xing and Keer Xu and Zhixing Xu and Liulu Yue and Daniel Zeman and Jin Zhao and Šárka Zikánová and Zdeněk Žabokrtský. 2025. *Uniform Meaning Representation 2.0*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Julia Bonn and Claire Bonial and Matt Buchholz and Hsiao-Jung Cheng and Alvin Chen and Ching-wen Chen and Andrew Cowell and William Croft and Lukas Denk and Ahmed Elsayed and Eva Fučíková and Federica Gamba and Carlos Gomez and Jan Hajič and Eva Hajičová and Jiří Havelka and Loden Havenmeier and Hana Hledíková and Ath Kilgore and Veronika Kolářová and Lucie Kučová and Kenneth Lai and Bin Li and Jingyi Li and Markéta Lopatková and Marie MacGregor and Marie Mikulová and Jiří Mírovský and Anna Nedoluzhko and Skatje Myers and Michal Novák and Tim O'Gorman and Petr Pajas and Alexis Palmer and Martha Palmer and Jarmila Panevová and Benét Post and James Pustejovsky and Petr Sgall and Jialin Song and Li Song and Magda Ševčíková and Jan Štěpánek and Zdeňka Urešová and Haibo Sun and Yao Sun and Rosa Vallejos Yopán and Jens Van Gysel and Meagan Vigus and Kristin Wright-Bettner and Jiawei Wu and Nianwen Xue and Dan Xing and Keer Xu and Zhixing Xu and Liulu Yue and Daniel Zeman and Jin Zhao and Šárka Zikánová and Zdeněk Žabokrtský. 2026. *Uniform Meaning Representation 2.2*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jan Štěpánek and Markéta Lopatková and Daniel Zeman and Federica Gamba and Hana Hledíková and Eva Fučíková and Michal Novák and Šárka Zikánová and Eva Hajičová and Jiří Havelka and Veronika Kolářová and Lucie Kučová and Marie Mikulová and Jiří Mírovský and Anna Nedoluzhko and Petr Pajas and

Jarmila Panevová and Petr Sgall and Magda Ševčíková and Zdeňka Urešová and Zdeněk Žabokrtský and Jan Hajič. 2025. *Uniform Meaning Representation 2.1 (Czech and Latin)*. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Orange @ UMR Parsing Shared Task

Johannes Heinecke, Munshi Asadullah
Orange Research, 22300 Lannion
johannes.heinecke@orange.com, munshi.asadullah@orange.com

Abstract

Uniform Meaning Representation (UMR) is a novel meaning representation formalism emanating from Abstract Meaning Representation (AMR). Since it is more complex than AMR, including document level annotation it is more difficult to create a parsing pipeline which can predict an UMR document from a set of consecutive sentences. The UMR Parsing Shared Task was created to compare different approaches. We decided to use a 2-step approach to predict sentence level and document level annotation. Since the available data was limited, we opted for a multilingual model, even though unlike AMR, in UMR the concepts of the meaning graph are not drawn from a single source, but from language dependend resources. Our final score was 19.35%, 0.08 points behind the best participant (19.43%).

1. Introduction

Uniform Meaning Representation (UMR, Van Gysel et al., 2021) is an evolution and extension of Abstract Meaning Representation (AMR, Banarescu et al., 2013). In addition to semantic graphs present in AMR, UMR not only adds information not represented in AMR graphs like number and person, and alignments between tokens of the sentence and instances of the semantic graph, but also includes a document level annotation to represent the temporal relations of states and events (before, after, overlap, …) mentioned in a set of sentences. Finally UMR annotates coreferential entities and events crossing the sentence boundary. UMR is conceived to be multilingual. This means that the semantic graph contains concepts not only drawn from the English PropBank (Kingsbury and Palmer, 2002; Palmer et al., 2005) as in AMR, but from the language of the sentence.

UMR data comes in documents which contain a set of consecutive sentences, their semantic graphs, alignments and temporal and coreferential links between sentences. Optionally UMR files may also contain information about POS and even morphemes of the words of the sentences and translations.

The UMR Parsing Shared Task (Štěpánek et al., 2026) proposes to create UMR data from a (segmented and tokenised) text documents. Test data was in the six languages for which training data was available and a "surprise" language, Italian.

2. Data

The UMR data provided by the organisers contained UMR documents in 6 languages: Arapaho[1] (arp), Chinese (zh), Czech (cs), English (en), Latin (la) and Navajo[2] (nv), cf. Tables 1 and 2. Apart from Czech the data does not contain many sentences and a large part of it is considered "dirty", i.e. synthetic data without final human validation. For Latin no "clean" data was available at all.

	docs.	sent.	words	chars.	sents/ doc
arp	1	53	274	2 029	53.0
cs	6	103	1 587*	8 846	17.2
en	4	180	2 023	9 488	45.0
nv	1	5	60	494	5.0
zh	20	557	15 375	41 292	27.9

Table 1: Available "clean" training. *For Chinese the character count means number of ideograms.

	docs.	sent.	words	chars.	sents/ doc
arp	2	292	1 627	12 915	146.0
zh	1	1 435	18 656	42 129	1 435.0
cs	6 516	159 906	2 537 317	14 100 350	24.5
en	571	29 872	292 023	1 493 470	52.3
la	3	1 049	19 139	117 420	349.7
nv	3	337	2 663	19 709	112.3

Table 2: Available "dirty" training

3. Our Approach

In the past we got state-of-the-art results in multilingual AMR parsing (Heinecke and Shimorina, 2022) by finetuning Flan-T5-base (Chung et al., 2022) or mT5-base (Xue et al., 2021), the latter for languages other than English. Since we did not succeed in improving our scores by using larger

[1] Algonquian language spoken in the US states of Wyoming and Oklahoma

[2] Athabascan language spoken in the South West of the USA

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 148–154
11 May 2026.

origin	dirty	clean	dirty	clean
destination	dev		train	
arp	1	0	1	1
cs	651	1	5 865	5
en	57	1	514	3
la	1	n/a	2	n/a
na	1	0	0	3
zh	0	1	1	19

Table 3: Split of "clean" and "dirty" documents into train and dev

LLM (like various version of Qwen2.5 (0.5B, 1.5B, 3B and 7B) and Qwen3 8B), unlike Chun and Xue (2024) we decided to try a two-staged approach for this task. A first step to get all intra-sentence information (the semantic graph, alignments, temporal relations and modal attribute roles), and a second step which tries to predict all inter-sentence related information (mainly temporal relations and coreferences).

Since UMR data differs considerably from AMR data, we did not use any of the AMR 3.0 data (Knight et al., 2020) and tried to build the entire pipeline using only the provided training data, including the "dirty" data. This means that we also refrained from using the corefences annotated partially in AMR 3.0. Another reason for not using AMR 3.0 data is the fact that it is only available for English, whereas in UMR the concepts of the graphs are drawn from the language of the sentence.

To train the 2 models for our two steps, we split the data in dev and train as shown in Table 4. In general all the clean dataset is used to finetune our models. If there is enough clean data, we put at least a clean document in the dev dataset. No document is in both, dev and train at the same time.

After playing with different finetunings we added some preprocessing

- replace relations `:refer-number` and `:refer-person` by `:person` and `:person` respectively,
- put all literals between quotes including special tokens (`3rd`, `full-affirmative`) (excepting numerical values, - or +),
- deleting `:wiki` relations (present 7 times in in the clean train set of Czech, twice Arapaho and Chinese, once in Navajo and 46 occurences in English). In the "dirty" training set only english and Navajo contain this relation. In addition, the `:wiki` relation points to wikidata entities in Czech and to Wikipedia pages in the other languages, for our multilingual this proved to be too complicated taking into account the available time.

dev

lang.	docs.	sents.	words	chars.
ar	1	59	329	3 040
cs	652	14 228	219 116	1 254 202
en	58	833	14 252	78 234
la	1	327	6 229	37 155
nv	1	50	344	2 474
zh	2	63	1 860	4 978

train

lang.	docs.	sents.	words	chars.
arp	2	286	1 572	11 904
cs	5 870	145 781	2 319 788	12 854 994
en	517	29 219	279 794	1 424 724
la	2	722	12 910	80 265
nv	3	292	2 379	17 729
zh	19	1 929	32 171	78 443

Table 4: number of documents, sentences etc. after our split in dev (top) and train (bottom)

We also detected some inconsistencies which we did not correct since the time frame of the shared task was too dense:

- named entities like `:op1 "Obama"` are included in the alignments in English (clean) but not in Chinese (clean)
- concepts like `s1x / %Rcp` found in Czech (dirty) (Left-overs from the translation of data taken from the Prague Dependency Treebank (PDT, Hajič et al. (2020))[3] UMR)

The exact size of the files per language after split is shown in Table 4.

Even though the size of the datasets is very small, we did not try to create synthetic data as proposed by Gamba et al. (2025) (starting from Universal Dependencies treebanks).

3.1. Sentence graph

The first step in our pipeline tries to predict all intra-sentence information: the graph, the alignments and the document level annotations unless a variable from another sentence is involved. To do so, we merge the graph from the training data and add all intra-sentence document level annotation by creating instances for tokens like "root" or "document-creation-time". Since the relations used in the document level annotation are different than the relations in the graph, they can be identified and extracted during the inference (cf. example in Fig. 1). We then finetuned 3 models

[3] https://ufal.mff.cuni.cz/prague-dependency-treebank

```
(v / publication-91
  :ARG1 (v2 / landslide-01
    :ARG3 (v3 / and
      :op1 (v4 / die-01
        :ARG1 (v5 / person
          :quant "200")
        :aspect "state")
      :op2 (v6 / fear-01
        :ARG1 (v7 / miss-01
          :ARG1 (v8 / person
            :quant "1500")
          :aspect "state")
        :aspect "state")
      :aspect "process")
    :place (v9 / country
       :name (v10 / name
         :op1 "Philippines"))
    :before-of
        (v11 / document-creation-time
           :overlap v6)
    :overlap v4
    :overlap v7)
  :root (v12 / root
    :modal (v13 / author
      :full-affirmative v2
      :full-affirmative v4
      :full-affirmative v6
      :partial-affirmative v7))
```

Figure 1: Sentence graph enriched by intra-sentence relations taken from document level annotation in bold. Temporal relations are bold and underlined, modal attribute roles are bold only (taken from clean training data `english_umr-0001.umr`)

to predict an enriched UMR graph from a simple sentence. Two models were only trained in monolingual data, even if it was very little: 1) Flan-T5-base 2) mt5-Base. The third model was trained on training data of all languages combined. To avoid a dominance of the Czech data, we took only 10% of the "dirty" Czech data. Fig. 2 details our training pipeline for the first step

Since Flan-T5 is mainly pretrained on English data, for all other languages the (monolingual) fine-tuning of mT5 resulted in much better results. In lack of the test-data we used our dev split to identify the best model. However, for the languages for which only very little data is available (Arapaho, Latin and Navajo) the multilingual model performed far better. Only for Czech with a big train set the monolingual model was better (Table 5). Note that we used the dev dataset to determine the best option. This is not optimal, especially since our dev set contains documents from the "dirty" dataset, but the official test set only contains clean data.

	monolingual		multilingual
	mT5	Flan-T5	mT5
Arapaho	36.12	34.85	**40.97**
Chinese	41.71	11.67	**49.70**
Czech	**86.30**	68.14	77.14
English	58.73	59.73	**60.86**
Latin	33.48	31.81	**48.95**
Navajo	31.76	27.77	**41.15**

Table 5: Training results to identify which model performs best for which language. Except for Czech, the multilingual model based on mT5 outperforms the other versions

A different problem is the alignment of instances of the sentence graph with the words (tokens) of the sentence. We have regarded this problem with the least priority due to time reasons. There exist aligners for AMR (such as AMRlib[4]), but we did not use this approach for two reasons: The obvious absence of training data and the fact that alignment in AMR is different from the one in UMR. In AMR tokens are not only aligned to instances, but also to literals (names, quantities) or to relations (e.g. the English preposition "by" can be aligned to an `:ARG0` relations in a Passive Voice construction).

3.2. Alignments

In order to have at least some – basic – alignments we opted for a guessing method: We use a Lemmatizer (UDParse[5]), trained on data from the Universal Dependencies project (UD)[6] and then map lemmas to instances of concepts. This fails of course when two different instances of the same concept appear in the sentence. In case of named entities, where the name itself is an attribute in the sentence graph, we add a postprocessing to find the instance of the concept having the name. For Navajo and Arapaho, no UD data is available. In this case our fallback is mapping forms to concepts. Due to the complex morphology of these two languages, the recall is very low.

3.3. Document level annotations

We employed a different approach to infer inter-sentence document level annotations. The training data suggested that most inter-sentence relations between instances are rarely more than six sentences apart. To finetune our model we used Qwen3 8B. We ran 3 experimental setups to resolve inter-sentence annotations, i.e. co-reference and temporal relations. For all 3 setups the data

[4] https://github.com/bjascob/amrlib
[5] https://github.com/Orange-OpenSource/UDParse
[6] https://universaldependencies.org

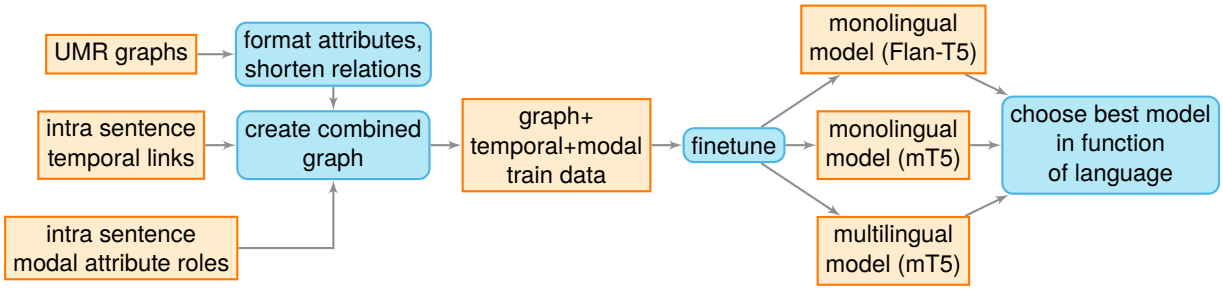


Figure 2: Schema of the training of the first step. For the multilingual model we combined all data of our train split (but only 10% of the Czech data)

was generated by creating sentence pairs for a maximum span of 6 on a sliding window, within a document as these relations do not exist on inter-document level. Each data point thus consists of the first sentence and its corresponding UMR graph, the second sentence and and its corresponding UMR graph. For setup 1 and 2 inter-sentence co-references were not exploited and for setup 3 we also provided the distance between the sentences to the training and prediction. We quickly realized the overwhelming number of data points without any co-references to exploit thus it produced a model that did not produce any positive outputs. So we introduced filtering and rejected all documents without any inter-sentence relations for the second setup. Although, improved, the best F value achieved on the validation data was 0.06. For the third setup, we kept the filter and included inter-sentence relations and balanced the number of positive and negative samples. On validation data we achieved 0.48 F value. A schema of the training flow is shown in Fig. 3.

4. Inference of the official test data

The test data provided by the organisers consists of 30 documents with in total 1019 sentences (cf. Table 6). In addition to the languages already available for training, one document was in Italian (it).

lang.	docs.	sents.	words	chars.	sents./ doc
arp	2	55	274	2 076	27.5
cs	5	220	4 048	26 042	44.0
en	5	195	4 092	22 084	39.0
it	1	100	2 212	12 135	100.0
la	1	50	889	5 554	50.0
nv	1	163	1 194	12 911	163.0
zh	15	236	6 467	37 998	15.7
total	30	1 019	19 176	23 668	

Table 6: test data size

The data flow is as shown in Fig. 4. As said above, the models of the first step predict the sentence graph with the temporal and modal document level annotations (unless the relation contains an instance of a preceding sentence). The graph is then split into the UMR graph and the document level annotations. The sentence and the UMR graph is the input for the second step (dashed box in Fig. 4). Alignments are added at the same moment (cf. section 3.2). Unfortunately, we could not generate any inter-sentence document level annotations. We suppose that our approach was not adapted to the scarcity of the training data.

The final step is the generation of the UMR output file. Here we also delete quotes around special tokens like `3rd` or `full-affirmative` and rename the relations `:number` and `:person` back to their official form `:refer-number` and `:refer-person` respectively. Due the the little training data, our mT5 finetuned models tend to output duplicate relations, which we also remove. Since formal errors in the UMR file will mean a score of 0, we used the provided validation script to spot formal error (missing instances in alignments, cycles in graphs) and repair them automatically.

5. Results on official test data

Our results are shown in Tables 7 and 8. Unsurprisingly both Arapaho documents score very badly, most likely due to the little data available for training and absence of any Arapaho data in the pretrained mT5 mode which we finetuned. Interestingly the Navajo document scores much better, even though the clean training data was even smaller than for Arapaho (but dirty training data was slightly bigger). Interestingly our simple aligner worked rather well for Chinese (70%) and English (71.4%) but failed completely for Italian (18.7%). Even the morphologically complex languages of Arapaho (40/7%) and Navajo (48.2%) scored better. The modal and temporal document level annotation were 0 for all languages but Chinese. And we failed to predict a single coreference.

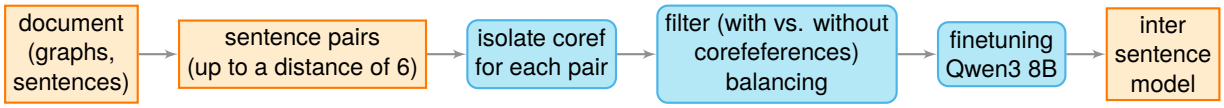


Figure 3: Finetuning of the inter-sentence document level annotations

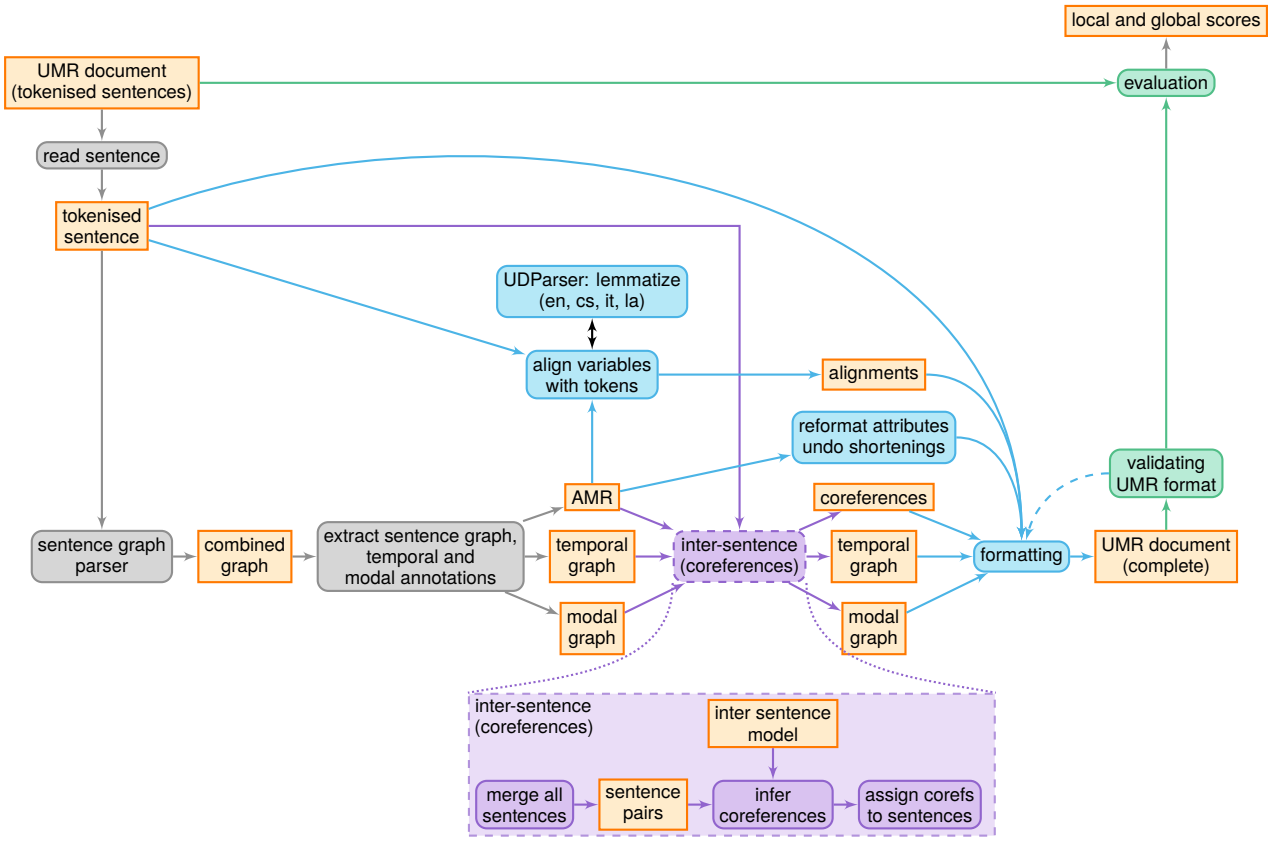


Figure 4: Flow diagramme for inference (the large dashed box at the bottom is a zoom on the inter-sentence inference)

In a post-Shared-Task experiment we corrected a bug in the prediction of coreference so that we finally got coreferences, but this did not change the global score.

English and Chinese score best in our case. Both languages are well represented in mT5. Additionally in the case of Chinese only one document of the training data was “dirty” which obviously improved the parsing result.

6. Conclusion and perspective

We presented our approach to the UMR shared task where we were able to finish in with a global score of 19.35%. Due to this, in absolute terms low score, the UMR parsing is currently not exploitable since there are too many errors, even in major languages like English, Chinese or Czech. Apart from needing more high-quality data, inconsistencies between languages and annotations in the existing data must be resolved.

7. Bibliographical References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Jayeol Chun and Nianwen Xue. 2024. Uniform Meaning Representation Parsing as a Pipelined Approach. In *Proceedings of TextGraphs-17:*

testfile	F1 score	testfile	F1 score
arp-0005	5.64%	zh-0032	29.44%
arp-0003	8.76%	zh-0036	29.68%
cs-0004	13.75%	en-0006	29.97%
it-0000	13.94%	zh-0026	33.32%
cs-0000	14.45%	zh-0028	34.79%
cs-0003	15.50%	zh-0024	35.36%
en-0000	15.90%	zh-0027	35.72%
cs-0001	16.29%	zh-0030	35.99%
la-0001	17.24%	zh-0034	39.39%
cs-0002	20.65%	zh-0023	40.29%
nv-0004	21.55%	en-0008	41.09%
en-0007	24.18%	zh-0029	41.47%
en-0005	24.34%	zh-0031	41.67%
zh-0021	25.31%	zh-0035	45.54%
zh-0022	28.61%	zh-0033	48.21%

Table 7: Our results per test file, sorted by score (average: 27.49%)

language	F1 score
Arapaho	8.15%
Italian	13.94%
Czech	15.87%
Latin	17.24%
Average	19.00%
Navajo	21.55%
English	22.19%
Chinese	36.51%

Table 8: Our results per language, sorted by score (average: 19.35%)

Graph-based Methods for Natural Language Processing, pages 40–52, Bangkok, Thailand. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Federica Gamba, Alexis Palmer, and Daniel Zeman. 2025. Bootstrapping UMRs from Universal Dependencies for Scalable Multilingual Annotation. In *Proceedings of the 19th Linguistic Annotation Workshop (LAW-XIX-2025)*, pages 126–136, Vienna, Austria. Association for Computational Linguistics.

Jan Hajič, Eduard Bejček, Jaroslava Hlavacova, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague dependency treebank - consolidated 1.0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5208–5218, Marseille, France. European Language Resources Association.

Johannes Heinecke and Anastasia Shimorina. 2022. Multilingual Abstract Meaning Representation for Celtic Languages. In *Proceedings of the 4th Celtic Language Technology Workshop within LREC2022*, pages 1–6, Marseille. ELRA.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 1989–1993, Las Palmas, Canary Islands - Spain. European Language Resources Association.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Jens E. L. Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a Uniform Meaning Representation for Natural Language Processing. *Künstliche Intelligenz*, 35:343–360.

Linting Xue, Noa Constant, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–498. Association for Computational Linguistics.

Jan Štěpánek, Daniel Zeman, Markéta Lopatková, Federica Gamba, Hana Hledíková, and Nianwen Xue. 2026. First shared task on UMR parsing. In *Proceedings of the Seventh International Workshop on Designing Meaning Representations*, Palma, Spain. ELRA.

8. Language Resource References

Kevin Knight and Bianca Badarau and Laura Baranescu and Claire Bonial and Madalina Bardocz and Kira Griffitt and Ulf Hermjakob and Daniel Marcu and Martha Palmer and Tim

O'Gorman and Nathan Schneider. 2020. *Abstract Meaning Representation (AMR) Annotation Release 3.0*. Linguistic Data Consortium. distributed via LDC: LDC2020T02, 3.0, ISLRN 676-697-177-821-8.

Sema System for the DMR 2026 Shared Task: Multistage UMR Parsing with Qwen3-4B

Rémi de Vergnette, Maxime Amblard
LORIA, UMR 7503, Université de Lorraine, CNRS, Inria
54000 Nancy, France
remi.de-vergnette@loria.fr, maxime.amblard@univ-lorraine.fr

Abstract

We present the Sema system for the DMR 2026 shared task on parsing from natural language to UMR. Our approach relies on parameter-efficient fine-tuning of Qwen3-4B with a multistage training procedure. We first train on a capped subset of the noisy training data, then continue training on the clean split, and finally fine-tune a dedicated stage for word-to-node alignment prediction. The system generates sentence-level graphs, selected document-level information, and alignments in separate steps, followed by rule-based post-processing to satisfy the official evaluation format. Results show that the approach is viable across several languages and exhibits promising transfer to Italian despite the absence of Italian data for fine-tuning , while very low-resource languages remain challenging.

Introduction

Parsing to AMR (Banarescu et al., 2013) has been studied extensively. In contrast, parsing from natural language to UMR (Bonn et al., 2024) remains much less explored. UMR extends AMR with the goal of providing a more expressive representation of meaning and capturing phenomena that are not well handled in English-centered sentence-level graph formalisms. Compared with AMR, UMR differs in three main ways: (i) it annotates phenomena that are largely ignored in AMR, (ii) it explicitly includes node-to-word alignment, and (iii) it introduces an additional document-level layer to capture inter-sentential phenomena.

The DMR 2026 shared task (Štěpánek et al., 2026) is the first shared task dedicated to this problem. In this paper, we present the system we developed for this shared task. Our system is based on parameter-efficient fine-tuning (PEFT) of a large language model (LLM) in several stages. More precisely, it combines four main design choices: staged fine-tuning on noisy and clean data, a dedicated alignment generation stage, a custom linearization for alignments, and rule-based post-processing to satisfy the official evaluation format. Contrary to previous work on UMR parsing (Chun and Xue, 2024; Markle et al., 2026), we do not explicitly rely on AMR as an intermediate step by either training on AMR data or using an AMR parser. Instead, we directly train on UMR data.

The remainder of the paper is organized as follows. Section 1 describes the data. Section 2 presents the system architecture and training procedure. Section 3 describes generation. Section 4 details the post-processing pipeline. Section 5 gives official results and some elements of analysis.

	clean	dirty
English	180	29872
Chinese	557	1435
Navajo	5	337
Arapaho	53	292
Czech	103	159906
Latin	0	1049

Table 1: Number of training samples per split and language.

1. Data

The DMR 2026 shared task provides a training set composed of two splits: a *dirty* split and a *clean* split. The number of samples of each split, for each language, is given in Table 1. The training data is highly unbalanced, especially in the dirty split, where Czech represents more than 80% of the samples. Annotation quality also varies across the data. In particular, the dirty split contains automatically generated annotations that may be incomplete, especially for the document-level layer and word-to-node alignment.

Annotations are organized by document and by sentence. For each sentence, the annotation contains three parts: the sentence-level graph, the document-level annotation, and the word-to-node alignment. Figure 1 gives an example. Note that alignment is represented as a relation between nodes and a possibly empty set of token-index ranges.

We do not train on the full dirty split, as explained below.

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 155–159

```
# meta-info :: sent_id = u_tree-cs-s2-
root
# :: snt2
Index: 1  2  3     4 5     6     7
 8   9      10
Words: If it rains , Alana won't water
 the plants .

# sentence level graph:
(s2w / water-01
    :ARG0 (s2p / person
        :name (s2n / name :op1 "Alana"))
    :ARG1 (s2p2 / plant
        :refer-number plural)
    :condition (s2r / rain-01
        :aspect process)
    :aspect performance)

# alignment:
s2w: 7-7
s2p: 5-5
s2n: 0-0
s2p2: 9-9
s2r: 3-3

# document level annotation:
(s2s0 / sentence
    :temporal ((document-creation-
time :after s2w)
               (document-creation-
time :after s2r))
    :modal ((root :modal author)
            (author :full-
affirmative s2r)
            (author :full-
negative s2w)))
```

Figure 1: Example of annotation for a sentence.

2. System architecture

2.1. Overview

We fine-tune Qwen3 4B (Yang et al., 2025) with LoRA (Hu et al., 2022). The model predicts the intra-sentential fragment of UMR, namely the sentence-level graph and the word-to-node alignment, as well as part of the document-level annotation, mainly modal relations.

Our training pipeline consists of three stages:

1. training sentence-graph generation on a capped subset of the dirty split;
2. continuing training on the clean split while adding document-level annotations;
3. further fine-tuning on the clean split for alignment generation.

For graph prediction, the model is trained to produce a PENMAN-style representation of the graph without tab characters. Appendix A gives input and output templates for each of the different stages. We include both the language of the input sentence and its id, as node identifiers are always of the form `s{sentence id}`.

For alignment prediction, we do not exactly use the same format as the original data. Instead of generating node-to-word alignment, we generate word-to-node alignment. This yields a more natural generation order and makes it possible to leave words unaligned explicitly when needed.

All experiments are run on a single A100 GPU with 40GB of memory. Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER, several universities, and other organizations.

2.2. Data preparation

As mentioned above, the training data is highly unbalanced. In particular, the dirty split contains a very large number of Czech examples and much fewer samples for other languages. We found that training on the full dirty split leads to prohibitive training times for limited gains in performance. We therefore cap the number of dirty examples per language at 2000 and sample accordingly.

We also reverse the alignment direction and generate alignments in a word-to-node format. Concretely, alignments are represented line by line as `word:node` or `word:` when no node is aligned.

We normalize sentences by trimming each tab character with a space and collapsing repeated spaces into a single space. We also normalize graphs by removing all tab characters. The data includes morphological and syntactic annotations for Navajo and Arapaho, but we do not include them in the model input, in order to keep a consistent format across languages. Note that we don't use any special linearization technic for the graphs that was done for AMR in variants of SPRING (Bevilacqua et al., 2021). We simply use a PENMAN-style linearization without tab characters, which is sufficient to obtain well-formed outputs.

2.3. Training

We use LoRA (Hu et al., 2022) to fine-tune Qwen3 4B (Yang et al., 2025). We apply LoRA to both attention and feed-forward projections. We use different hyperparameters for the dirty and clean stages; the alignment stage uses the same hyperparameters as the clean stage. Table 2 summarizes the training setup.

We train the model with the HuggingFace implementation of LoRA and the `transformers` li-

brary. We use the AdamW optimizer and a constant learning-rate schedule with warmup. In preliminary experiments, a small number of warmup steps gave slightly better performance. Because the per-device batch size is small, we use gradient accumulation.

We reset the optimizer and scheduler states between stages, while initializing each stage from the model obtained at the previous one. For this reason, we also keep warmup in the clean stage, even though it starts from a previously trained checkpoint.

We experimented extending the dirty split with alignment annotations using on a simple Levenshtein-based heuristic, but this did not yield improvements and was not included in the final training pipeline.

3. Generation

We generate annotations in two steps. First, we generate the sentence-level graph together with the document-level annotation using the model saved at the end of the clean stage. Second, we generate word-to-node alignments using the model saved at the end of the alignment stage.

We decode the test set with beam search using a beam size of 4. The generation parameters are given in Table 3. For very long sentences, generation can become unstable: the model sometimes falls into loops and keeps introducing new nodes. When this happens, we restart the generation with the same configuration and keep the first valid output obtained after post-processing.

4. Post-processing

Because the official evaluator requires a specific output format, we apply several post-processing steps to the generated annotations. First, we parse the generated PENMAN-style graph and check whether it is well formed. When parsing fails, we apply simple repair heuristics : inserting missing parentheses or removing unmatched ones if they can be deteced. We then convert the graph to the format expected by the official evaluator.

We also convert alignments from our word-to-node format back to the official format and add empty alignments for nodes that are not aligned to any word. Alignment entries referring to non-existing nodes are removed. Finally, when the document-level annotation is missing, we insert an empty annotation block.

If the output still cannot be parsed after these steps, we fall back to an empty sentence-level graph and remove the corresponding alignment annotation.

5. Results

We observe a general agreement with the expected format, the main errors being related to the introduction of cycles in the generated graphs, visible by the fact that some nodes dominate themselves in the generated PENMAN.

Apart from this issue, the model generally produces well-formed annotations strictly following the expected format. As argued in (Xin et al., 2024), the ability to produce well-formed outputs is a key advantage of LoRA fine-tuning, and must be distinguished from the ability to produce semantically correct outputs. In our case, the model is able to learn to produce well-formed annotations, but performance remains limited in terms of semantic correctness, and especially for navajo and arapaho. Table 4 shows the official $\mathrm{ju{:}mætʃ}$ scores. The results are not uniform across languages, with very low scores for Navajo and Arapaho, and better results for Chinese, Czech, and Italian. The average score across languages is 0.1943, with lowest scores 0.1115 for Arapaho and highest score 0.2652 for Czech.

A qualitative analysis of the outputs shows that, for Navajo and Arapaho, the model often produces very generic and sparse annotations, as illustrated in Figure 2. This is likely due to the very small number of training examples for these languages and to their limited representation in the model pretraining data.

By contrast, Italian, which is absent from the fine-tuning data but likely present in the model pretraining data, obtains better results, as shown in Table 4. This is consistent with the hypothesis that multilingual pretraining enables some transfer to languages that are not present in task-specific fine-tuning, provided that they are sufficiently represented during pretraining. At the same time, the low scores for very low-resource languages indicate that pretraining alone is not enough in the most data-scarce settings.

Conclusion

We presented the Sema system for the DMR 2026 shared task on multilingual parsing from natural language to UMR. Our approach relies on multi-stage LoRA fine-tuning of Qwen3-4B, using capped dirty-data training, continuation on the clean split, a dedicated alignment generation stage, and rule-based post-processing. The results show that this approach is viable across several languages and that it exhibits encouraging cross-lingual transfer to Italian despite its absence in the fine-tuning data. However, performance remains limited for all languages, especially for very low-resource languages such as Navajo and Arapaho.

	dirty	clean / alignment
per-device train batch size	1	1
gradient accumulation steps	32	32
warmup steps	4	3
num train epochs	5	3
learning rate	2e-4	1e-4
weight decay	0.001	0.001
lr scheduler type	constant with warmup	constant with warmup
lora r	16	16
lora alpha	32	32
lora dropout	0	0

Table 2: Training hyperparameters for each stage.

	sentence graph & document-level annotation	alignment
max new tokens	1024	1024
num beams	4	4
temperature	0.7	0.7
top p	0.9	0.9

Table 3: Generation parameters for each step.

```
# :: snt43
Words: Noh ne'nih'iisnoo3woohok
 nouun nehe' hisi' .

# sentence level graph:
(s43h / hii-00
    :actor (s43p / person
        :refer-person 3rd
        :refer-number singular)
    :undergoer (s43p2 / person
        :refer-person 2nd
        :refer-number singular)
    :aspect performance)
```

Figure 2: Example of a generated annotation for a Navajo sentence.

Language	F1
Arapaho	0.1115
Chinese	0.2585
Czech	0.2652
English	0.1919
Italian	0.2137
Latin	0.1918
Navajo	0.1273
Average	0.1943

Table 4: Score for each language and the macro-average reported by the shared task.

6. Bibliographical References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer, Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E. L. Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italia. ELRA and ICCL.

Jayeol Chun and Nianwen Xue. 2024. Uniform meaning representation parsing as a pipelined

approach. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 40–52, Bangkok, Thailand. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Emma Markle, Javier Gutierrez Bach, and Shira Wein. 2026. Setup: Sentence-level english-to-uniform meaning representation parser.

Chunlei Xin, Yaojie Lu, Hongyu Lin, Shuheng Zhou, Huijia Zhu, Weiqiang Wang, Zhongyi Liu, Xianpei Han, and Le Sun. 2024. Beyond full fine-tuning: Harnessing the power of LoRA for multi-task instruction tuning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2307–2317, Torino, Italia. ELRA and ICCL.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Jan Štěpánek, Daniel Zeman, Markéta Lopatková, Federica Gamba, Hana Hledíková, and Nianwen Xue. 2026. First shared task on UMR parsing. In *Proceedings of the Seventh International Workshop on Designing Meaning Representations*, Palma, Spain. ELRA.

A. Prompts and templates

This appendix gives simplified input and output templates for the main stages of our pipeline.

Stage 1: sentence-level graph generation
Input:

```
Parse into the Uniform Mean-
ing Representation the fol-
lowing <language> sen-
tence <num> : " <text> ". You only need to in-
clude the sentence level graph.
```

Output:

```
# sentence level graph:
<sentence_graph>
```

Stage 2: sentence-level graph generation and document-level annotation
Input:

```
Parse into the Uniform Mean-
ing Representation the fol-
lowing <language> sen-
tence <num> : " <text> ". You need to in-
clude sentence level graph and docu-
ment level graph.
```

Output:

```
# sentence level graph:
<sentence_graph>
# document level annotation:
<document_level_annotation>
```

Stage 3: alignment generation
Input:

```
Complete with word to to-
ken alignment the follow-
ing Uniform Meaning Represen-
tation annotation for the sen-
tence " <text> " :  <sentence_graph>
```

Output:

```
# alignment:
<word1> : <node1>
...
<wordN> : <nodeM>
```

Meaning Annotation Experience. A Tribute to Petr Sgall

Marie Mikulová, Jan Štěpánek,
Barbora Štěpánková, Jarmila Panevová, Eva Hajičová
Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague 1, Czech Republic
mikulova@ufal.mff.cuni.cz

Abstract

We present ongoing work on annotating fine-grained semantic distinctions for circumstantial meanings, focusing on spatial expressions. We describe our theoretical background, and annotation process, as well as how we evaluate the results obtained. Using multiple independent annotations across the 3-million-token, genre-diverse Prague Dependency Treebank - Consolidated corpus of Czech data, we analyse inter-annotator agreement, recurrent disagreement patterns, and the limits of semantic categorization. Our results highlight the inherent vagueness of linguistic meaning. We also propose strategies for handling disagreement, such as weighted annotations, intermediate labels, and fuzzy labels that preserve annotation nuance. This work builds on the legacy of Petr Sgall and the Functional Generative Description theory that underpins the multi-layer form–meaning framework.

Keywords: meaning, semantics, vagueness, annotation, human label disagreement

1. Motivation

The world we live in—and the world we talk about—is immensely complex. Natural language is inherently ambiguous and vague (Sgall, 2002; Piantadosi et al., 2012); for example, no language can provide the means to distinguish between all possible locations of objects in such a world. Linguistic means (e.g., prepositions) typically serve to describe a wide range of extra-linguistic facts. We illustrate both the diversity of the world and the poly-functionality of linguistic expressions with examples from the Internet (1)–(3); see also Fig. 1.[1]

(1) *How the pumpkin got **on the tower**?*

(2) *The babies were first placed **on the tower** in 2000. Ten sculptures of toddlers climbing up and down...*

(3) *Some students standing behind a pillar of the Academic Center started shouting something about a guy **on the Tower** shooting people.*

In the examples (1)–(3), there is always the same expression from a formal perspective: the preposition *on* with the noun *tower*. However, the actual locations referred to this expression differ. Let's assume localizations (a)-(d):

(a) **at the top** of the specified place
(b) **on the outer surface** of the specified place
(c) **in the upper part** of the specified place
(d) **inside** the specified place

Localization (a) is referred to (1), as evidenced by the accompanying picture of the Cornell University tower in the respective text (cf. first picture in Fig. 1). Localization (b) is intended in (2), as illustrated by the picture, this time of the Prague tower (cf. the right picture in Fig. 1). In (3), the prepositional phrase *on the tower* expresses a rather complex localization (which includes (b), (c), (d)): the shooter was located in the upper part of the specified place, but not entirely at its top like the pumpkin in (1); he was on the tower's observation deck, but unlike the babies in (2), he was not only on the outer surface but also inside the tower (compare also the middle illustration in Fig. 1). The complexity of this localization is evidenced by the fact that while Texas Monthly magazine calls the article about the University of Texas tower shooting *The Madman **on** the Tower*, Time magazine reports on the same event under the title *The Madman **in** the Tower*.[2] This variability highlights how challenging it is for any semantic representation to capture such fine-grained distinctions.

2. Introduction

This paper contributes to the ongoing effort to build semantic representations,[3] emphasizing that the inherent vagueness of linguistic meaning is crucial for capturing the boundless diversity of the world.

[1] Examples (1)–(3) and the pictures in Fig. 1 are taken from the following websites:
https://cornelldailysun.github.io/pumpkin-feature/;
https://www.ourbeautifulprague.com/babies-on-the-tower-and-on-kampa/;
https://www.texasmonthly.com/true-crime/the-madman-on-the-tower.

[2] Prepositions highlighted by the authors of the paper.

[3] Enhanced Rhetorical Structure Theory (Zeldes et al., 2025), Uniform Meaning Representation (Van Gysel et al., 2021), Deep Universal Dependencies (Droganova and Zeman, 2019), Xposition project (Gessler et al., 2022); cf. also the survey papers: Ma et al. (2025); Sadeddine et al. (2024); Dobnik et al. (2022).

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 160–171
11 May 2026.

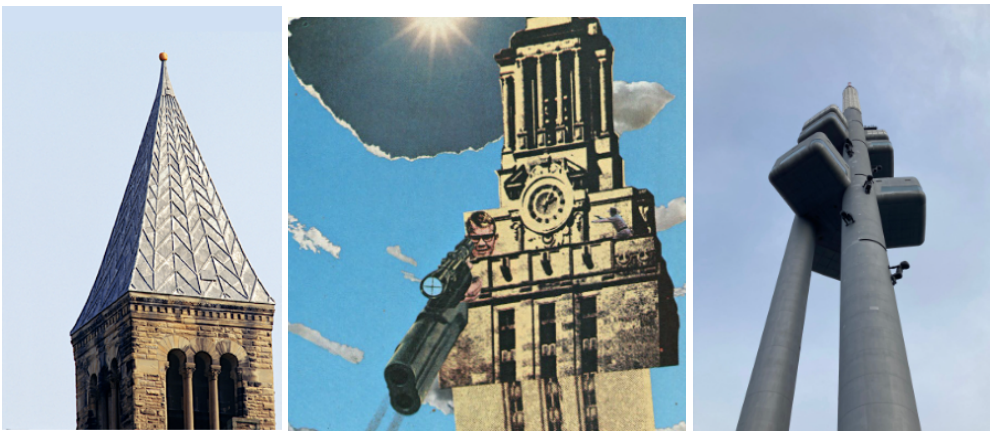


Figure 1: 'On the tower' examples illustrating different placements of pumpkin, shooter, babies on a tower, all described using the same prepositional phrase.

While underspecification, vagueness, and ambiguity rarely hinder everyday communication, they pose challenges when designing a semantic classification that is fine-grained, cognitively plausible, distinguishable, and human-understandable.[4] In this paper, we present an update on our work on designing and annotating fine-grained semantic distinctions in the expression of spatial, temporal, manner, and other circumstances in Czech (cf. previous contributions to this topic: Mikulová, 2024; Mikulová et al., 2025a) within the framework of the Prague Dependency Treebank (Mikulová et al., 2026). We address the following issues:

(i) granularity of meaning categories to ensure its credibility, broadness in coverage, and suitability for consistent manual annotation of real texts;

(ii) the relation between language and the world it describes.

We are now completing the annotation of fine-grained semantic distinctions in the spatial domain. We have produced multiple independent annotations of over 126,500 circumstants in the Prague Dependency Treebank - Consolidated 2.0, a 3-million-token corpus of genre-diverse Czech texts (Hajič et al., 2024). In the paper, we summarize the key results of this large-scale annotation, assess annotator disagreement, and discuss how the annotations will be represented in the final dataset.

The concept of extensively annotated corpora within the Prague Dependency Treebank framework is rooted in the Functional Generative Description (FGD), one of the most influential modern Czech linguistic theories (Sgall et al., 1986). The theory provides a multi-layer framework based on form-meaning relation, capturing semantic and syntactic phenomena such as valency and information structure.

2026 marks the centenary of **Petr Sgall**, the founder of FGD. Petr Sgall greatly influenced natural language processing and helped develop computational linguistics in the Czech Republic. This paper pays tribute to his enduring impact on linguistics and language technologies at the occasion of what would have been his hundredth birthday.

The paper consists of two parts. The first chapters outline the theoretical background established in FGD: multi-layer annotation scheme (Sect. 3); the notion of linguistic meaning (Sect. 4); the distinction between meaning and content (Sect. 5); and vagueness in language (Sect. 6). The second part describes the annotation of real texts: task definition (7.1), guideline development (7.2), the annotation process (7.3), and result evaluation (7.4), followed by a discussion of disagreement treatment (Sect. 8). The paper concludes in Sect. 9.

3. Multi-layer Language Description

It is indisputable that language has a complex, multi-level structure, and therefore the process of describing a language must also be structured in a certain way.[5] (Sgall, 2006)

Our classification of circumstants is developed within the **Prague Dependency Treebank** (PDT) framework.[6] The long-term process of building the PDT corpora, as well as the current research on the circumstantial meanings, has repeatedly convinced us that a complex multi-layer annotation scheme is well founded both theoretically and computationally. Multi-layer language description views the form–meaning relation as composed of several layers, each with distinct functions contributing to overall meaning. The PDT multi-layer

[4] A considerable amount of work has emerged in this area; cf. specialized workshops: Pyatkin et al. (2024); Roth and Schlechtweg (2025); Lai and Wein (2025).

[5] The original quote is in Czech: *Je nesporné, že jazyk má složitou, mnohavrstevnou strukturu, takže i postup popisu jazyka musí být určitým způsobem strukturován.*

[6] https://ufal.mff.cuni.cz/pdt-c

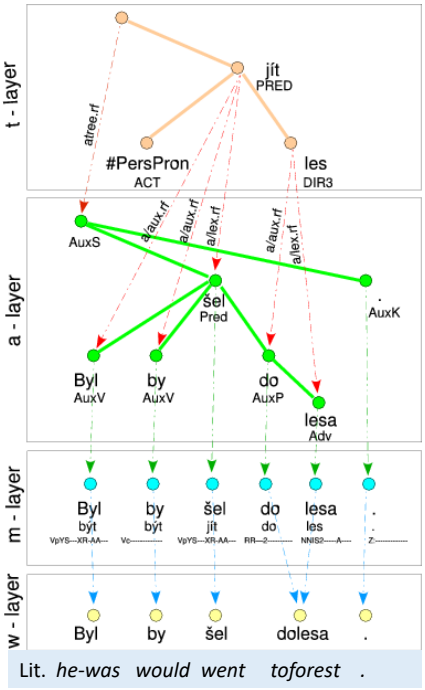


Figure 2: PDT multi-layer annotation scheme illustrated on the example of the Czech sentence: *Byl by šel do lesa.*'lit.: He-was would went to forest.'

architecture (originally proposed by Petr Sgall, e.g., in Sgall, 1965), schematically illustrated in Fig. 2, and described in numerous studies (most recently in Mikulová et al., 2026) is based on form–meaning relation and enables a comprehensive description of the relations between morphological properties, syntactic functions, and expressed meanings. It contributes to higher accuracy in language description and to the overall data consistency (Hajičová et al., 2022; Mikulová et al., 2025b).

The highest layer in PDT scheme is the layer of meaning (*t-layer* in Fig. 2),[7] while the lower layers capture surface structure (*a-layer*) and morphological properties (*m-layer*).[8] For example, the spatial circumstant *do lesa* 'to forest' (cf. Fig. 2) is represented at the t-layer by a single node with the so-called *functor* `DIR3` (meaning "where to"), which corresponds at the a-layer to a two-node prepositional structure, and at the m-layer, the morphological properties of these words are captured by a 15-character tag, indicating, among other things, their POS, case, number, and gender. In the original input text (*w-layer*), there is a typo: the prepositional phrase is incorrectly written without a space.

[7] It captures complex semantic annotations of a sentence: predicate-argument structure, semantic roles, semantic counterparts of morphological categories, topic-focus articulation, coreference, ellipsis.

[8] The other lower layers contain the input text, or, where applicable, transcription and audio.

The PDT multi-layer annotation makes it easy to examine the relations between form and meaning. It is obvious that there is no one-to-one correspondence between meaning categories and forms. For example, the form *do*+2[9] expresses not only the meaning "where to" (`DIR3`, as in Fig. 2), but also the meaning "until when" (e.g., *do pondělí* 'until Monday', `TTILL`) or quantificational modification (*do dvaceti let* 'under the age of twenty', `EXT`). And vice versa, the meaning "where to" is expressed not only by the form *do*+2, but also by a wide range of other forms, e.g., *nad*+4 (*nad les* 'above forest'), *poblíž*+2 (*poblíž lesa* 'near forest'). From these examples, we can see that functors capture circumstantial meanings only as generalized categories and from the perspective of semantic description, reflect only a coarse classification. These instances of the general meaning "where to" differ in a more specific locations ("into the specified place", "above the specified place", "behind the specified place", "near the specified place", etc.). The introduction of a set of "narrower" meanings, the so-called *subfunctors*, makes it possible to capture these semantic distinctions.[10]

The annotation of functors has already been completed in the PDT-C corpus. For the upcoming release in 2028, we further enrich the semantic annotation in the corpus by subfunctor annotation. In accordance with the principles on which the corpus is built, we establish a repertoire of subtle meaning categories and specify the formal means by which partial meanings are expressed. Methodologically, however, we proceed in the opposite direction: for particular forms, we determine their semantic functions. Morphosyntactic description and linguistic meaning representation are related but pursue opposite goals: the former uses semantic equivalence to establish formal patterning, while the latter uses formal distinctions to uncover the structure of meaning categories and compare semantic concepts (cf. Haspelmath, 2010).

4. Linguistic Meaning Layer

Without distinguishing the level of meaning it is difficult to imagine an integrated description of language, since the linguistic structuring of semantic and pragmatic issues has to be described independently on what we assume to be the "real" or "actual" structure of the world. (Hajičová and Sgall, 1980)

The top layer in the PDT multi-layer scheme (at which we annotate subfunctors) is conceived as

[9] In the paper, the nouns are indicated by number of morphological case, i.e. 2 for noun in *Genitive*, 3 for *Dative*, 4 for *Accusative*, 6 for *Locative*, 7 for *Instrumental*.

[10] The need for a finer classification of functors was first described in Panevová (1980).

Subfunctor	Forms	Example
`above`	*nad* 'above/over'	*nad lesem* 'above the forest'
`adjacency`	*u*, *při* 'by'	*u lesa* 'by the forest'
`alongside`	*podle*, *podél* 'along'	*podél lesa* 'along the forest'
`among`	*mezi* 'among'	*chodit mezi stromy* 'to walk among trees'
`area`	*po* 'on/around'	*chodit po domě* 'walk around the house'
`around`	*okolo*, *kolem* 'around'	*kolem lesa* 'around the forest'
`behind`	*za* 'behind/beyond'	*za lesem* 'behind the forest'
`below`	*pod* 'below/under'	*pod lesem* 'under the forest'
`beside`	*vedle* 'beside/next to'	*vedle lesa* 'next to the forest'
`between`	*mezi* 'between'	*cesta mezi dvěma lesy* 'path between two forests'
`direction`	*na+4*, *směrem na+4* 'towards'	*jet směrem na Prahu* 'to go towards Prague '
`facing`	*čelem k* 'facing'	*čelem k lesu* 'facing the forest'
`foreground`	*v čele* 'at the head of'	*v čele kolony* 'at the head of the column'
`front`	*před* 'in front of'	*před lesem* 'in front of the forest'
`inside`	*v* 'in', *uvnitř* 'inside'	*v lese* 'in the forest'
`middle`	*uprostřed* 'in middle of'	*uprostřed lesa* 'in the middle of the forest'
`near`	*blízko*, *poblíž* 'near'	*blízko lesa* 'near the forest'
`opposite`	*naproti* 'opposite'	*naproti lesu* 'opposite the forest'
`otherside`	*přes*, 'across'	*hodit kámen přes řeku* 'to throw a stone across the river'
`outside`	*stranou*, *mimo* 'outside'	*stranou lesa* 'outside the forest'
`side`	*po boku* 'alongside'	*po boku manželky* 'alongside the wife'
`surface`	*na* 'on'	*nová barva na domě* 'new paint on the house'
`through`	*přes*, *skrz* 'through'	*strčit ruku skrz mříž* 'to put a hand through the bars'
`within`	*na*, *u* 'at/on/in'	*svatba na věži* 'wedding on the tower'

Table 1: Core subfunctors and selected forms for spatial circumstants

a layer of linguistic meaning. It captures the way semantic distinctions are structured within a given language. It differs from other domains that reflect non-linguistic structuring of (cognitive or ontological) content, primarily in two aspects (taken from the timeless article by Hajičová and Sgall (1980)):

(i) *while there is a clear support in the form of analysed language for the representation of linguistic meaning, no clear criteria have been found for the classification of units in the content/knowledge domain*,

(ii) *while a representation of meaning is one of the levels of the language system, a representation of the content is beyond language*.

We began addressing circumstantial meaning categories within the domain of linguistic meaning. This domain focuses on how a language reflects reality through its form and structure; consequently, our spatial subfunctors do not describe the exact placement of the pumpkin, the shooter, or the babies in (1)–(3) in Sect. 1, because the language itself (in our case, Czech) does not distinguish them—the same formal means are used for all three placements.[11] This strategy appears to work well for core spatial, temporal, and other circumstantial meanings; cf. Tab. 1, which summarizes our core subfunctors for the spatial circumstants.

[11] To address this, we introduced `surface` subfunctor for "on the surface" meaning and `within` for underspecified location "somewhere in there"; cf. Tab. 1.

5. And What is Beyond?

The level of meaning may be considered to constitute a suitable starting point for semantic-pragmatic interpretation of the sentence, i.e. of an analysis of its cognitive (ontological) content. (Sgall, 1995)

The interplay of semantics and pragmatics in the structure of natural language is far too complex a matter to be dealt with by simply including some pragmatic features in the structuring of meaning.(Sgall et al., 1986)

However, within spatial circumstants not only basic, literal spatial distinctions are expressed, but also a wide range of abstract, metaphorical, or otherwise extended meanings, which may differ from the basic one to varying degrees. Compare examples (4)–(6) with the prepositional phrase *v novinách* 'in newspaper', where the core spatial meaning `inside` is present only in (4). The other examples express transferred meanings: in (5), the newspaper is understood as an institution, and in (6) it refers to the content of the newspaper as a (literary) work. These readings appear to be desirable to distinguish in addition to the core meanings.

There are two main ways to handle these transferred meanings: assign a general "non-core" meaning label or try to divide them into finer categories. It is clear that language understanding is not based solely on linguistic meaning, but also on further semantic-pragmatic interpretation, during which the interpreter draws on contextual informa-

Subfunctor	Examples
event	*potkat se na návštěvě* 'to meet on a visit', *odejít do války* 'to go to war'
state	*být v domácnosti* 'to be a stay-at-home mother', *dostat se do bezpečí* 'to get to safety'
aim	*hnát se do útoku* 'to rush into attack', *přijmout někoho do služby* 'to take him into service'
institute	*pracovat ve škole* 'to work at school', *odejít od Siemensu* 'to leave (from) Siemens'
institute-p	*jít k holiči* 'to go to the barber', *přijít od doktora* 'to come from the doctor'
ingroup	*nejmenší ve třídě* 'the smallest in the class', *vmísit se do davu* 'to blend into the crowd'
function	*odejít z funkce vedoucího* 'to resign from a position as manager'
work	*hledat humor v knize* 'to search for humour in a book'
media	*písnička v rozhlase* 'a song on the radio', *dostat se na obrazovku* 'to get on the screen'
actinfo	*uvést něco v prohlášení, ve zprávě* 'to state something in a statement, in a message'
domain	*působit v zemědělství* 'to work in agriculture'
placings	*doběhnout na třetím místě* 'to finish in third place'
level	*pozdvihnout zábavu na vyšší rovinu* 'to took entertainment to a higher level'
value	*uzavřít obchodování na 48 centech* 'to close trading at 48 cents'

Table 2: Other subfunctors for spatial domain

tion and general world knowledge. The proposed "non-core" subfunctors are semantic-pragmatic, grounded in context and human knowledge.

(4) ***V novinách*** *najdete i přílohu.*
'You will find an addendum **in the newspaper**.'

(5) ***V*** *našich* ***novinách*** *pracovat nemůžete.*
'You cannot work **at** (lit. in) **our newspaper**.'

(6) *Ten inzerát jsem našel* ***v novinách****.*
'I found the advertisement **in the newspaper**.'

In delimiting these subfunctors, we rely on only a few linguistic "crutches". The most important is the principle of substitutability of forms (cf. Mikulová, 2024). For instance, in the "institution" meaning (in contrast to the core meaning "inside"; cf. (5)), Czech allows in some cases the use of the preposition *u*+2 'by/at' (e.g., *pracuje u novin* 'to work at a newspaper'). Another useful clue is a cross-linguistic perspective: cases in which a given meaning is formally distinguished in another language. For example, in English, the meaning 'in the newspaper' as an institution is formally distinguished from the simple spatial meaning (*in* vs. *at*). Tab. 2 summarizes the proposed "non-core" subfunctors.

Where no such "linguistic crutch" exists, the above-mentioned method of trial and error becomes necessary, and we are aware that in this area, in particular, careful evaluation is required: to what extent annotators will agree on these categories, and how well the proposed classification covers the data (see Sect. 7.4).

6. How Do We Understand?

Without a certain degree of indistinctness of language meaning (i.e., of the units of the layer of functions of expressions in the language system) it would not be possible to capture with limited means the unlimited range of the world we perceive and speak of. (Sgall, 2002)

And what lies at the end of this process? Is it even possible to sort something as inherently vague as language into meaningful categories? It turns out that after we classify the basic and clearly identifiable meanings (Tab. 1) and single out the predominantly abstract ones (with respect to the domains of the annotated texts; Tab. 2), we are still left with a relatively large number of cases whose meaning is difficult to describe, or where it is unclear how fine-grained the analysis should be: should we distinguish in the semantic representation between (4) and (7), or between (6) and (8)? It becomes evident that in cases where the annotation is not supported by the language itself—namely when non-core meanings are involved—and is based instead on knowledge and understanding, factors that rely heavily on human judgment and are therefore subjective, the annotations often diverge (cf. Mikulová et al., 2025a).

(7) ***V novinách*** *byla díra.*
'There was a hole **in this newspaper**.'

(8) *Ty lži se objevily jen* ***v*** *těchto* ***novinách****.*
'Those lies appeared only **in this newspaper**.'

This raises a broader epistemological issue: to what extent can semantic annotation strive for objectivity when the boundaries between meaning categories are fluid and context- and knowledge-dependent? Even with carefully designed guidelines, annotators may interpret borderline cases differently, not because of a lack of training, but because language itself does not provide stable cues. As a result, annotation schemes inevitably reflect theoretical assumptions about meaning segmentation, highlighting the limits of any attempt to fully systematize meaning. Semantic categorization is inherently approximate. However, examining the sources of disagreement and the limits of meaning annotation (see Sect. 8) provides a valuable insight into how speakers conceptualise space (and content more broadly).

-	a12	a09	a07	a01	a10	a06	a03	a15	a14	a11	a04	a05	a02	a00	a13
a08	86	93	87	86	94	88	95	89	87	88	85	88	87	81	84
a12	-	89	85	84	90	87	90	87	85	87	86	86	78	78	81
a09		-	90	88	92	92	92	89	86	90	81	94	96	86	86
a07			-	88	90	89	90	88	87	89	90	90	89	79	85
a01				-	91	90	91	86	93	89	90	90	90	79	83
a10					-	94	95	94	92	93	94	93	91	91	89
a06						-	94	92	89	93	92	92	91	86	86
a03							-	95	89	94	91	96	87	88	87
a15								-	90	92	95	95	92	85	87
a14									-	91	93	87	91	86	84
a11										-	92	93	92	86	87
a04											-	93	96	83	91
a05												-	**98**	87	86
a02													-	80	88
a00														-	79

Table 3: Inter-annotator agreement over all the tasks. The percentage of non-empty intersections of annotated values in all the data. Special labels are ignored.

7. Annotation Process

In this section, we describe practical experience with what we have theoretically discussed above, including the specific course and outcomes of our annotation. More details are in the Appendix A.

7.1. Defining the Task

The semantic annotation of circumstantial meanings throughout the entire corpus involves assigning subfunctors to nodes expressing spatial, temporal, manner, causal, and other types of circumstances. We began with the spatial domain. Within the PDT framework, we distinguish four functors for spatial meanings: `LOC` ("where"), `DIR1` ("from where"), `DIR2` ("which way"), and `DIR3` ("where to"). In annotating subfunctors, we take advantage of the fact that the functors are already annotated in the corpus. To ensure greater consistency, we do not annotate all spatial circumstants at once (there are approximately 126,500 occurrences, expressed by roughly 120 different forms).[12] Instead, we proceed in smaller tasks defined by functor and form—e.g., the functor `DIR3` expressed by *do*+2.

7.2. Developing Guidelines

The set of subfunctors for a functor–form combination is proposed on the basis of an analysis of at least 200 randomly selected corpus examples for each such combination. We use the ForFun database[13] (Mikulová and Bejček, 2018), which is extracted from the corpus and organizes its formal and semantic annotations in a user-friendly tool. When designing subfunctors, we make use of available Czech dictionaries[14] in which the preposition meanings are described to a certain extent. Most of the proposed subfunctors are shared across all spatial functors, while several more specific subfunctors are developed individually for each one. An overview of the proposed subfunctors is provided in Tab. 1 and 2.

7.3. Performing Annotation

Based on the proposed sets of subfunctors, we carry out multiple annotations of all occurrences of each functor–form combination throughout the entire corpus. In addition to selecting a mandatory subfunctor (or marking the case as problematic, e.g., an error in the functor assignment; such cases are not included in the calculations presented here in Sect. 7.4), annotators can also use a special label indicating an abstract or idiomatic meaning. If annotators are uncertain about the appropriate subfunctor, they are allowed to provide one more option and add an explanatory comment. Each occurrence is annotated by at least three different annotators.

7.4. Evaluating Results

One of the key stages in meaning annotation is a careful evaluation of the results. We assess both the annotators and the annotations themselves. We measure the extent to which annotators agree (Sec. 7.4.2) and identify which annotators disagree the most (Sec. 7.4.1). Furthermore, we examine the extent of annotator agreement for each proposed subfunctor (Sect. 7.4.3) and how often individual subfunctors co-occur (Sect. 7.4.4).

[12] A form means a preposition (including a wide range of complex ones) plus a case combination here, i.e. we do not count adverbs, subordinate clauses.

[13] http://hdl.handle.net/11234/1-2542

[14] https://prirucka.ujc.cas.cz/

Considered	L+O+	L+O-	L-O+	L-O-
α	0.706	0.710	0.727	0.731

Table 4: Krippendorff's α. We use four different ways to calculate it: with or without the special label (L+/L-) and with or without considering option order (O+/O-).

7.4.1. Inter-Annotator Agreement

As the annotators were given the possibility to use two labels, the simple classic methods of measuring their agreement are not directly applicable. To get a general overview of the consistency of the annotation, we calculate a simplified agreement as shown in Tab. 3. The highest agreement is 98%, the lowest 78%. By summing the table (without omitting the mirrored values) by rows (or columns), we can get the "most different" annotators and further inspect how their annotation differs to others.

7.4.2. Krippendorff's Alpha

We also calculate Krippendorff's coefficient α to measure the overall inter-annotator agreement (Tab. 4). The coefficient stays above 0.667 recommended by Krippendorff for at least tentative conclusions, and when broken by task (i.e. functor and form), two tasks achieve over 0.8, the satisfactory threshold for firm conclusions (while some others drop below 0.667). Ignoring the order of options does not change the value much (mostly because a single option is prevalent), ignoring the special label has a larger impact (+0.02 overall, but up to +0.178 for one of the two worst performing tasks).[15]

7.4.3. Category-wise Kappa

To calculate the agreement for a category, we calculate Cohen's κ for each pair of annotators for the category and take the average value as the result. Tab. 5 shows the highest scoring values for spatial meanings. By category we here understand potentially both the annotated labels including their preference order but ignoring their special labels. Including two-value annotations, the entire table has 439 rows. The highest-scoring two-value category (`domain,work`) has a score of $\kappa = 0.111$. To calculate the expected agreement for each annotator pair, there are two possibilities: to base the distribution of each annotator on the overlapping data only, or to use all the annotated data by each annotator. The difference in the final κ is always less than 0.001.

[15] Although annotation is performed by tasks and comparing them helps us refine the categories and clarify guidelines, the coefficient α per task does not have a scientific value because, for each task, the number of possible forms, the number of its instances, and the number of possible categories varies widely.

Category	κ
`value`	0.878
`event`	0.808
`inside`	0.783
`below`	0.770
`outside`	0.766
`function`	0.765
`behind`	0.757
`front`	0.757
`placings`	0.753
`level`	0.739
`opposite`	0.721
`selection`	0.705
`work`	0.697
`domain`	0.696
`within-person`	0.666
`foreground`	0.635
`above`	0.593

Table 5: Category-wise Kappa (the highest scoring values for spatial meanings).

7.4.4. Confusion Matrices

For each task (i.e. functor and form(s)), group of tasks, or the whole spatial domain we plot a table similar to a confusion matrix (see Fig. 3 in Appendix A). This gives us clues for the distinction which subfunctors are well defined and understood and what are the common sources of disagreement among annotators.

8. Handling Label Disagreement

The in-depth analysis of the results reveals three major groups of cases arising from the multiple annotations:

(i) complete agreement (100% consensus),

(ii) recurrent disagreement patterns, and

(iii) indeterminate annotations—odd or seemingly random label combinations.

We now develop strategies for how to present the differing outcomes (i)–(iii) in the final dataset. We avoid simple aggregation or majority voting, as such approaches would not accurately reflect the reality of meaning annotation and leads to significant information loss and uncertain ground truth labels in applications with high label variance (cf. Uma et al., 2021; Plank, 2022). Instead, we aim to preserve the distinction between cases where annotators showed clear agreement on a subfunctor and cases where the choice of a semantic category was inherently ambiguous, with no clear consensus among annotators.

We conducted a small **experiment** to test whether additional rounds of simultaneous multiple labelling would lead to stronger agreement. We selected 50 random cases that showed zero agreement in the original annotation annotated by small number of annotators (see Sect. 7.3) and

-	Majority: yes	Majority: no
Twice: yes	25	8
Twice: no	2	15

Table 6: Experiment. "Twice" means the winning label was at least two times more frequent than the second most frequent one; "Majority" means the winning label was selected by more than a half of the annotators.

had them annotated by 14 annotators. The results are in some respects quite interesting, although not particularly convincing for drawing broader generalizations (see Tab. 6). For each case, we compared the frequencies of the two most frequently assigned subfunctors. They were never the same, i.e. there was always a "winning" label. Of the 50 sentences, in 33 cases the winning label occurred at least two times more often than the second one. 27 winning labels were higher than 7 (i.e. selected by the majority), but only a half of the cases shared the two characteristics. The intrinsic difficulty of these cases is well illustrated by one annotator's remark: *I must admit that this task made me doubt almost every case, and I often could not determine what seemed most appropriate.* Examples from the experiment are shown in (9)–(13), with the original ambiguous annotation and the annotation obtained in the experiment.

We examine which groups or combinations of subfunctors are most frequently involved in disagreements. Since some subfunctor were introduced intuitively and experimentally, without strong grounding in linguistic form (cf. Sect. 5), it is essential to verify whether annotators agree on them to a reasonable degree; those with consistently low agreement should be reconsidered. We apply two strategies: removing a label (or redistributing its instances across other labels) and introducing an intermediate label. Candidates for **removal** include `actinfo` and `media` (cf. ex. (9)), which are often confused with each other as well as with several other subfunctors, including `work` and `inside`—labels that have some of the highest category-wise Kappa scores (cf. Tab. 5).

The confusion matrices (7.4.4) reveal the recurrent disagreement patterns. Among the combinations with the highest number of occurrences are: `inside` / `institute`, `ingroup` / `institute`, `event` / `institute`. However, these subfunctors exhibit high agreement (cf. Tab. 5). For such borderline cases, it may be more appropriate to introduce an **intermediate label** rather than forcing annotators to choose a single label or removing certain labels altogether (cf. (10) and (11)). See more details in Appendix A.

The analysis also showed that there is a relatively large number of cases with indeterminate or noisy label combinations, where none of the well-defined labels fit neatly, or where labels overlap. Such cases may need to be treated as fuzzy. For instances where agreement remains inconclusive, we consider introducing a **fuzzy label** to better capture the uncertainty in meaning. Cf. (12).

Disagreements can be resolved by assigning different weights to labels obtained from multiple annotations. As well as reducing the weight of alternative labels added by annotators, lower weights can be assigned to annotations from annotators with low IAA (Sect. 7.4.1) and to early annotations.

(9) ***V** informačním **bulletinu** se na str. 3 píše: ...*
'**In** the information **bulletin**, on p. 3 it says: ...'
Orig.: 1 work / 1 media / 1 inside / 1 actinfo
Exp.: 10 work / 3 media / 1 inside

(10) *Odehrál jsem **v NHL** čtyři dobré sezony.*
'I played four good seasons **in the NHL**.'
Orig.: 2 event / 2 institute
Exp.: 7 event / 4 institute / 3 ingroup

(11) *Neudělal nic, aby své návrhy prosadil, a tak byly **v parlamentu** poraženy.*
'He did nothing to push his proposals through, and so they were defeated **in the parliament**.'
Orig.: 1 institute 1 ingroup
Exp.: 9 institute / 5 ingroup

(12) *Lidé klečeli **v písku** a hledali kamínky.*
'People knelt **in the sand** and looked for stones.'
Orig.: 1 inside / 1 among
Exp.: 8 inside / 4 among / 1 coulisse / 1 skip

(13) *V prvních letech byl **v** projekční **kanceláři**.*
'In the early years he was **in** a design **office**.'
Orig.: 1 institute / 1 inside/abstract / 1 domain
Exp.: 14 institute

Low IAA does not necessarily indicate errors but may reflect divergence from the majority, introducing inconsistency. Annotators' judgments improve with experience, becoming more confident and consistent; later annotations are therefore more reliable, as in (13), where all annotators eventually reached agreement.

The adjustments discussed here—removing labels with consistently low agreement, introducing intermediate and fuzzy labels, and assigning lower weights to certain annotations—could help bridge the gap between rigid taxonomies and the gradient nature of linguistic meaning.

9. Conclusion

We report ongoing work on annotating fine-grained circumstantial meanings in the Prague Dependency Treebank framework, focusing on spatial expressions. Our large-scale, multi-annotator annotation reveals patterns of agreement and disagreement, highlighting the inherent vague-

ness of linguistic meaning and the influence of context and world knowledge. Proposed strategies—weighted annotations, intermediate, and fuzzy labels—preserve this nuance, reflecting the true complexity of meaning annotation. This work also honors Petr Sgall, whose Functional Generative Description laid the foundation for multilayered, form–meaning-oriented approaches that continue to guide Czech linguistic annotation.

10. Limitations

This paper describes an ongoing research project. So far, only a subgroup of spatial circumstants has been annotated, with annotation of time circumstants being next. So far, the project has only targeted Czech, but we are aware other languages might structure the spatial details differently. We also believe that the experience gained here is applicable across languages.

11. Ethics Statement

Sixteen student annotators participated in the project. They were compensated fairly for their contributions in the form of monetary payment. To protect their confidentiality, all personal identifiers were removed from the data, ensuring anonymity throughout the research process.

12. Acknowledgements

The research reported in the paper has been supported by the Czech Science Foundation under the projects GA23-05238S and by the Ministry of Education, Youth and Sports of the Czech Republic, Project No. LM2023062 LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz).

We would like to thank all our outstanding annotators for not working like machines, but for thinking critically during annotation and pointing out the shortcomings of the annotation guidelines. Without their efforts, this contribution would not have been possible.

13. Bibliographical References

Simon Dobnik, Robin Cooper, Adam Ek, Bill Noble, Staffan Larsson, Nikolai Ilinykh, Vladislav Maraev, and Vidya Somashekarappa. 2022. In Search of Meaning and Its Representations for Computational Linguistics. In *Proceedings of the 2022 CLASP Conference on (Dis)embodiment*, pages 30–44, Gothenburg, Sweden. Association for Computational Linguistics.

Kira Droganova and Daniel Zeman. 2019. Towards Deep Universal Dependencies. In *Proceedings of the Fifth International Conference on Dependency Linguistics*, pages 144–152, Paris, France. Association for Computational Linguistics.

Luke Gessler, Austin Blodgett, Joseph C. Ledford, and Nathan Schneider. 2022. Xposition: An Online Multilingual Database of Adpositional Semantics. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1824–1830, Marseille, France. European Language Resources Association.

Jan Hajič, Eduard Bejček, Alevtina Bémová, Eva Buráňová, Eva Fučíková, Eva Hajičová, Jiří Havelka, Jaroslava Hlaváčová, Petr Homola, Pavel Ircing, Jiří Kárník, Václava Kettnerová, Natalia Klyueva, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, David Mareček, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Michal Novák, Petr Pajas, Jarmila Panevová, Nino Peterek, Lucie Poláková, Martin Popel, Jan Popelka, Jan Romportl, Magdaléna Rysová, Jiří Semecký, Petr Sgall, Johanka Spoustová, Milan Straka, Pavel Straňák, Pavlína Synková, Magda Ševčíková, Jana Šindlerová, Jan Štěpánek, Barbora Štěpánková, Josef Toman, Zdeňka Urešová, Barbora Vidová Hladká, Daniel Zeman, Šárka Zikánová, and Zdeněk Žabokrtský. 2024. *Prague Dependency Treebank - Consolidated 2.0* . LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Charles University, Prague, Czech republic, http://hdl.handle.net/11234/1-5813.

Eva Hajičová, Marie Mikulová, Barbora Štěpánková, and Jiří Mírovský. 2022. Advantages of a Complex Multilayer Annotation Scheme: The Case of the Prague Dependency Treebank. In *Proceedings of the 16th Linguistic Annotation Workshop*, pages 70–78, Marseille, France. ELRA.

Eva Hajičová and Petr Sgall. 1980. Linguistic Meaning and Knowledge Representation in Automatic Understanding of Natural Language. In *COLING 1980: The 8th International Conference on Computational Linguistics*, pages 67–75.

Martin Haspelmath. 2010. Comparative Concepts and Descriptive Categories in Cross-Linguistic Studies. *Language*, 86(3):663–687.

Kenneth Lai and Shira Wein, editors. 2025. *Proceedings of the 6th Workshop on Designing Meaning Representations*. Association for Computational Lingustics, Prague, Czechia.

Bolei Ma, Yuting Li, Wei Zhou, Ziwei Gong, Yang Janet Liu, Katja Jasinskaja, Annemarie

Friedrich, Julia Hirschberg, Frauke Kreuter, and Barbara Plank. 2025. Pragmatics in the Era of Large Language Models: A Survey on Datasets, Evaluation, Opportunities and Challenges. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 8679–8696, Vienna, Austria.

Marie Mikulová. 2024. Fine-grained Classification of Circumstantial Meanings within the Prague Dependency Treebank Annotation Scheme. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 7314–7323, Torino, Italia. ELRA and ICCL.

Marie Mikulová and Eduard Bejček. 2018. ForFun 1.0: Prague database of forms and functions – an invaluable resource for linguistic research. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, Miyazaki, Japan. ELRA.

Marie Mikulová, Jiří Mírovský, Milan Straka, Pavlína Synková, Barbora Štěpánková, Jan Štěpánek, and Jan Hajič. 2026. Prague Dependency Treebank - Consolidated 2.0: Enriching a Complex Annotation Scheme. In *Proceedings of the 15th Language Resources and Evaluation Conference*, Palma de Mallorca, Spain.

Marie Mikulová, Jan Štěpánek, and Jan Hajič. 2025a. Label Bias in Symbolic Representation of Meaning. In *Proceedings of the 19th Linguistic Annotation Workshop*, pages 142–159, Vienna, Austria. ACL.

Marie Mikulová, Barbora Štěpánková, and Jan Štěpánek. 2025b. From Form to Meaning: The Case of Particles within the Prague Dependency Treebank Annotation Scheme. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2163–2175, Abu Dhabi, UAE. Association for Computational Linguistics.

Jarmila Panevová. 1980. *Formy a funkce ve stavbě české věty*. Academia, Prague, Czechia.

Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2012. The Communicative Function of Ambiguity in Language. *Cognition*, 122(3):280–291.

Barbara Plank. 2022. The "Problem" of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Valentina Pyatkin, Daniel Fried, Elias Stengel-Eskin, Alisa Liu, and Sandro Pezzelle, editors. 2024. *Proceedings of the 3rd Workshop on Understanding Implicit and Underspecified Language*. ACL, Malta.

Michael Roth and Dominik Schlechtweg, editors. 2025. *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*. International Committee on Computational Linguistics, Abu Dhabi, UAE.

Zacchary Sadeddine, Juri Opitz, and Fabian Suchanek. 2024. A Survey of Meaning Representations – From Theory to Practical Utility. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2877–2892, Mexico City, Mexico. ACL.

Petr Sgall. 1965. Generation, Production, and Translation. In *COLING 1965*.

Petr Sgall. 1995. From Meaning via Reference to Content. In *Karlovy Vary studies in reference and meaning*, pages 172–183. Filosofia Publications, Prague, Czech Republic.

Petr Sgall. 2002. Freedom of Language: Its Nature, Its Sources, and Its Consequences. In *Prague Linguistic Circle Papers: Travaux du cercle linguistique de Prague nouvelle série. Volume 4*, pages 309–329. John Benjamins Publishing Company.

Petr Sgall. 2006. Valence jako jádro jazykového systému. *Slovo a slovesnost*, 67(3):163–178.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. Academia/Reidel, Prague/Dordrecht.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, Jan Hajič, James H. Martin, Stephan Oepen, Martha Palmer, James Pustejovsky, Rosa Vallejos, and Nianwen Xue. 2021. Designing a Uniform Meaning Representation for Natural Language Processing. *KI-Künstliche Intelligenz*, 35(3-4):343–360.

Amir Zeldes, Tatsuya Aoyama, Yang Janet Liu, Siyao Peng, Debopam Das, and Luke Gessler. 2025. eRST: A Signaled Graph Theory of Discourse Relations and Organization. *Computational Linguistics*, 51(1):23–72.

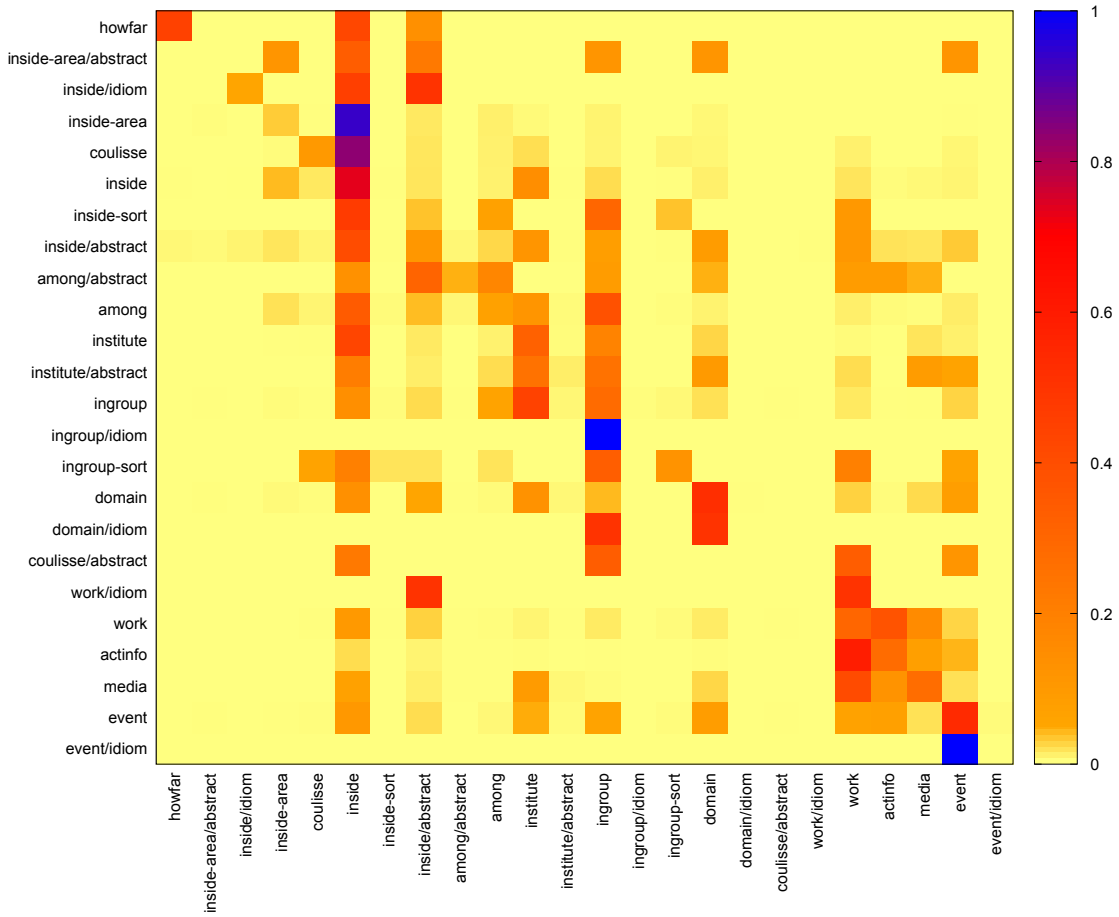


Figure 3: Confusion matrix for the spatial meaning "where" (functor `LOC`) expressed by *v*+6 'in'. There is no golden data, so we instead show how often each subfunctor rivals other subfunctors. The table is normalized by rows.

A. Appendix

Here, we provided more information on the course and evaluation of the annotation process. As described in Sect. 7.1, the annotation is carried out in smaller batches—tasks. An example of such a task is spatial meanings of "where" (assigned in the data with the functor `LOC`) expressed by *v*+6 'in'. In this task, annotators selected from the group of 14 subfunctors and could optionally assign a special label, `abstract` or `idiom`. In total, 23,522 instances were annotated in this task, with each instance annotated at least three times. The distribution of individual meanings/subfunctors is not proportional: the most frequent subfunctor assigned was `inside` (50,385 occurrences[16]), while the least frequent was `domain/idiom` (1 occurrence).

[16]Each instance was annotated at least three times, each annotator could specify two different values, so the number of subfunctor occurrences is higher than the number of instances.

The confusion matrix in Fig. 3 shows how often, in this task, a given subfunctor (optionally combined with an associated special label, indicated after a slash) was confused with another subfunctor during annotation. The darker a cell, the more frequently the row subfunctor co-occurs with the column subfunctor. Ideally, the darkest cells should lie on the diagonal, indicating agreement in subfunctor(/special label) selection. Dark cells off the diagonal may point to problematic phenomena, but they may also reflect marginal cases. Each such case therefore needs to be evaluated carefully.

The matrix shows, for example, that the `ingroup` subfunctor occurs much more frequently (in absolute terms across all pairs of subfunctors) in combination with `institute` (2,702 occurrences) than with `ingroup` itself (1,694 occurrences). Examples (see (14) and (15)) indicate that in some cases it is indeed difficult to clearly distinguish between the meaning "within a group based on shared interests" (`ingroup`) and "within an institution" (`institute`). For instance, a *committee* can

be understood both as an institution and as a group of people of the same interest. Given that both of these subfunctors also have a substantial number of clear cases where they do not compete with any other subfunctor (cases with 100% IAA), we do not consider it appropriate to merge them. Instead, we find it more suitable to label cases of ambiguity as an intermediate `ingroup-institute` label.

The matrix further indicates that `work`, `media`, and `actinfo` are very frequent competing subfunctors; moreover, `actinfo` co-occurs more often with `work` than with `actinfo` itself. The definitions of these subfunctors have been rather vague from the beginning of the annotation process, and this caused the confusion. Subfunctor `work` refers to a content of a work (e.g., content of a book: *violence in a story*). Subfunctor `media` also involves content (e.g., a program or information) conveyed through a particular medium (*a show on television*). The subfunctor `actinfo` emphasizes the form of communication—the act of conveying information in some way (*in his statement, he said that...*), which, however, may take the form of a written text (which may be mistaken for the `work` meaning) or a television program (which may be confuses with the `media` meaning). Cf. (16), in which all three values were assigned. The `actinfo` and `work` labels compete in (17) and (18). The blurred boundaries suggest that these values should be merged into a single category.

(14) *služba **v** cizím **vojsku***
'service **in** a foreign **army**'

(15) *Zasedal **v** nejrůznějších **komisích**.*
'He served **on** various **committees**.'

(16) *Řekl to **v pořadu** Proč na stanici ABC.*
'He said this **on** ABC's **program** Why.'

(17) ***Ve** svém **článku** citoval odůvodnění trestního stíhání.*
'**In** his **article**, he cited the justification for the criminal prosecution.'

(18) ***Ve** zvláštní **zprávě** ministerstvo oznámilo, že stavební náklady byly....*
'**In** a special **report**, the ministry announced that construction costs were...'

(19) *To bylo jenom **v** nejužším rodinném **kruhu**.*
'That was only **in** the closest family **circle**.'

The `ingroup/idiom` column is empty because, apart from three occurrences with `ingroup`, `ingroup/idiom` does not appear at all. The dark cell for `ingroup/idiom` vs. `ingroup` therefore represents the most frequent (indeed the only) combination for `ingroup/idiom`, but its shading does not necessarily indicate that confusion between them is common. These marginal cases mainly concern idioms (instances with the special label `idiom`) and reveal borderline cases of idiomaticity. (19).

Extending Uniform Meaning Representation to Persian: The First Corpus Resource

Minoo Nassajian, Daniel Zeman
Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics (ÚFAL)
Prague, Czechia
{nassajian, zeman}@ufal.mff.cuni.cz

Abstract

Uniform Meaning Representation (UMR) has cross-linguistic design principles that make it particularly well-suited as a semantic representation framework for capturing all language-specific phenomena. Despite its growing adoption, no UMR corpus currently exists for Persian. In this paper, we present the first version of a Persian UMR dataset created through a rule-based conversion of existing Persian AMR annotations from The Little Prince corpus, followed by manual mapping of split semantic roles from AMR to their finer-grained UMR counterparts. We report detailed statistics on the conversion, analyze the challenges of mapping Persian AMR structures into UMR, and provide illustrative examples. The resource is freely available and it lays the groundwork for subsequent enrichment of Persian UMR with additional semantic layers, including co-reference, named entities, and discourse relations.

Keywords: Uniform Meaning Representation, UMR, Persian, Abstract Meaning Representation, AMR

1. Introduction

Meaning representation is a fundamental concept in computational linguistics and natural language processing (NLP) that uses formal systems to capture the semantic content of language in structured, machine-processable format, and can improve the performance of different applications such as machine translation (Gao and Vogel, 2011; Song et al., 2019), summarization (Liu et al., 2015), semantic search (Ribeiro et al., 2022), data augmentation (Shou et al., 2022), or dialogue systems (Bonial et al., 2020; Kapanipathi et al., 2021; Pan et al., 2015). Meaning representation frameworks can be broadly categorized into graph-based and non-graph-based structures (Sadeddine et al., 2024). Among the graph-based ones, AMR (Banarescu et al., 2013) has been widely used to represent sentence-level meaning as directed, rooted, acyclic graph in which each node is associated with a specific concept and edges connect concept nodes while showing different semantic relation types. However, despite the widespread usage of AMR, the difficulties of extending it to diverse low-resource languages, coupled with its lack of annotations crucial for logical inference, such as modality, aspect, and scope, necessitated the development of UMR (Van Gysel et al., 2021). The present work is the first step in developing a Persian UMR dataset, with the goal of significantly enhancing cross-lingual semantic understanding, enabling more effective comparison of meaning between Persian and other languages already covered by UMR. In particular, we focus on split roles, following the methodology outlined by (Post et al., 2024).

This paper is structured as follows: Section 2 describes the related work within the context of both AMR and the emerging UMR framework, highlighting the gap in resources for Persian. Section 3 details the origin and characteristics of the initial corpus that serves as the foundation for our conversion effort. Subsequently, Section 4 describes the conversion methodology and the set of rules developed for mapping AMR semantic roles to their UMR counterparts. Section 5 provides a quantitative overview of the resulting Persian UMR corpus, analyzing the frequency and distribution of role mappings to validate our methodology and reveal insightful linguistic patterns. Finally, the conclusion and future work section summarizes our findings and outlines the critical next steps for expanding this corpus into a comprehensive UMR resource.

2. Related Work

UMR research has progressively expanded across diverse languages since its introduction. The foundations of the framework were laid down by Van Gysel et al. (2021), followed by the first data release in 2023 (Bonn et al., 2023b), which included datasets of varying sizes for six languages: English, Chinese (358 sentences from Wikinews), Navajo (Athabascan, USA; 522 sentences from historical narratives), Arapaho (Algonquian, USA; 408 sentences from narrative texts), Kukama (Tupian, Amazon; 105 sentences from traditional sto-

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 172–182

ries), and Sanapaná (Mascoian, Paraguay; 602 sentences). In the second release (Bonn et al., 2025), the English dataset grew substantially, with 87,038 sentences being converted from existing AMR resources through semi-automated processes. This release also added a dataset for Latin (50 manually annotated sentences from the Sallust treebank) and Czech (175,500 sentences automatically converted from the Prague Dependency Treebank, out of them 91 sentences also annotated manually) (Štěpánek et al., 2025).

There has been relatively limited research on meaning representation for the Persian language. One notable exception is the work by Mirzaei and Moloodi (2016), who introduced the Persian Proposition Bank (PerPB). This resource extends the Persian Dependency Treebank (PerDT) (Rasooli et al., 2013) with a semantic layer of predicate-argument annotations, inspired by PropBank (Kingsbury and Palmer, 2002) and VerbNet (Kipper et al., 2006). Their approach treats not only verbs but also propositional nouns and adjectives as semantic predicates, annotating over 29,000 sentences with detailed semantic roles.

The most recent research on Persian meaning representation focused on creating a Persian AMR dataset (Takhshid et al., 2022; Tohidi et al., 2024). They developed this corpus by annotating the Persian translation of "The Little Prince," containing 1,562 sentences, and addressed the annotation guidelines[1] for Persian-specific constructions such as light verb constructions, impersonal constructions, and clitics.

Building on this line of research, the present study introduces the first step towards a Persian UMR corpus by converting and extending the existing Persian AMR dataset. Specifically, we apply the AMR-to-UMR conversion framework proposed by Post et al. (2024), which provides guidelines for refining split semantic roles. This effort not only highlights the applicability of cross-linguistic UMR guidelines to Persian but also uncovers language-specific challenges—such as complex predicates, clitic behavior, and pro-drop—that shape the design and annotation of meaning representations. The next section outlines the characteristics of the Persian AMR dataset.

3. Source Data: Persian AMR Corpus

Due to the limited availability of UMR resources for Persian, the present corpus aims to provide an initial annotated dataset that supports linguistic analysis and future development of Persian UMR annotation. In this regard, we use the only available Persian AMR (PAMR) dataset (Takhshid et al., 2021) introduced by Takhshid et al. (2022). This corpus represents the first attempt to adapt AMR to Persian and provides a valuable foundation for cross-linguistic meaning representation. During the construction of PAMR, certain AMR features were adjusted for the Persian language as follows:

- Light verb constructions: Persian uses extensive light verb constructions (LVCs) where a non-verbal element combines with a semantically lighter verb to form a predicate and due to structural integrity they are considered as single lexical verbs (Karimi-Doostan, 1997). So, Persian AMR explicitly preserves them as unified semantic units. For example, in "talâŝ kardan" (to do an effort), the non-verbal element "talâŝ" (effort) cannot be separated from the light verb "kardan" (to do) in semantic representation without losing essential information. In AMR, "talâŝ kardan" is kept as one concept node.

- Pro-drop characteristics: As a pro-drop language, Persian allows null subjects realized only through verb morphology. This creates challenges for AMR's more explicit argument structure. Additionally, inanimate subjects might not follow subject-verb agreement in number, further complicating semantic representation (Karimi, 2008).

- Clitics and possessive constructions: Persian employs various clitics that can serve as subjects, objects, or possessors. Some constructions have no overt subject but use pronominal enclitics to indicate possession or experiencer roles.

As UMR is designed in a way that accommodates languages across diverse typological spectrums, it effectively addresses these linguistic features specific to Persian that were previously challenging for AMR. In particular, UMR allows for the compositional semantic representation of Persian LVCs by aligning the concept, representing the LVC, back to the surface tokens (the light verb and its object). It also systematically represents implicit subjects common in Persian pro-drop contexts by explicitly encoding inferred arguments. Furthermore, UMR is capable of clearly representing semantic roles of Persian clitics, thus resolving the ambiguity that AMR previously faced in encoding possessive and experiencer relationships. It should be also noted that since the corpus is derived from Persian AMR annotations, some upstream annotation errors may propagate. Manual inspection during split-role conversion mitigated this risk by correcting inconsistent semantic structures when detected.

[1] https://github.com/Persian-AMR/Annotation-Guidelines

In the next section, we describe the rule-based mapping approach for converting split roles from AMR to UMR, following the guidelines proposed for English, and discuss how we adapt and apply these rules to Persian.

4. Conversion Methodology

UMR inherits the overall graph-based architecture of AMR[2] but introduces finer-grained semantic roles that better capture cross-linguistic distinctions. Bonn et al. (2023a) provide the most comprehensive cross-mapping to date and show that AMR roles must undergo four distinct types of change when converted to UMR:

1. New roles: UMR introduces roles that have no direct AMR equivalent — such as `:actor`, `:experiencer`, and `:force` — to capture semantic distinctions (e.g., initiator vs. causer) that are crucial across languages.

2. Renamed roles: Certain AMR roles remain conceptually similar but are given more typologically neutral labels to improve clarity, consistency, and cross-linguistic applicability. For example, the `:location` tag becomes `:place`, and `:beneficiary` is changed to `:affectee`.

3. Split roles: Some AMR roles are underspecified and may correspond to multiple UMR roles depending on context. For example, AMR's `:cause` must be resolved to either `:cause` (physical causation) or `:reason` (motivational explanation); `:source` may map to `:source` (animate giver), `:start` (origin of motion), or `:material` (substance).

4. Unchanged roles: A number of roles, such as `:purpose`, `:instrument`, and `:manner`, transfer directly with no modification because they already align well with cross-linguistic semantic needs.

Among these categories, split roles are particularly critical because they require detailed linguistic analysis and often language-specific cues to resolve. Accordingly, this paper focuses on identifying and converting split roles as the essential first step toward a full Persian UMR corpus.

[2]Here we refer specifically to what is known as the sentence-level graph in UMR. In addition, UMR defines document-level relations, which are beyond the scope of the present paper.

4.1. AMR to UMR Conversion Principles

A key challenge in AMR to UMR conversion is the presence of split role cases where a single AMR relation can map to multiple possible UMR roles depending on context such as animacy, event type, or discourse function. Post et al. (2024) employ the animacy feature together with a probability distribution derived from gold-standard frequencies to define conversion rules (a - f) as follows:

(a) `:cause` →

$$\begin{cases} \texttt{:cause} & \text{if animate} \\ \texttt{:cause}, \ \texttt{:reason} & \text{if inanimate} \end{cases} \quad (1)$$

(b) `:destination` →

$$\begin{cases} \texttt{:goal} & \text{if animate} \\ \texttt{:recipient}, \ \texttt{:goal} & \text{inanimate} \end{cases} \quad (2)$$

(c) `:source` →

$$\begin{cases} \texttt{:source} & \text{if animate} \\ \texttt{:source}, \ \texttt{:material}, \ \texttt{:start} & \text{inanimate} \end{cases} \quad (3)$$

(d) `:consist-of` →

$$\begin{cases} \texttt{:group} & \text{if animate} \\ \texttt{:group}, \ \texttt{:part}, \ \texttt{:material} & \text{inanimate} \end{cases} \quad (4)$$

There are two other tags, called `:mod` and `:part`, that are considered "split roles" within the context of this conversion task for two key reasons. First, `:part` is a potential outcome of the genuinely non-deterministic AMR role `:consist-of` (which can also map to `:group` or `:material`). For the model to be complete, it must be able to predict `:part` as a target label, hence its inclusion in the set of "split" outcomes.

Second, for `:mod`, the split is not based on animacy but on semantic vagueness. In AMR, `:mod` denotes a dependent whose function does not fit into more specific relations. In UMR, there are two such vague roles: `:mod` for adjunct-like optional modifiers, and `:other-role` for entities

that seem to be core parts (participants) of the event but do not cleanly fit into any of the specific participant roles. Hence the converting rules for these tags are as follows:

(e) :mod → { :mod ; :other-role, if vague-modifier

(f) :consist-of, :part → {:part

As no robust animacy parser or large animacy-annotated corpus currently exists for Persian, and our objective was to achieve high-precision UMR annotations, we adopted the same role-mapping logic described above and adhered to the official UMR guideline[3] and all split-role decisions in our Persian corpus were performed manually. Moreover, linguistic characteristics such as LVCs, clitics, and pro-drop features create ambiguity when mapping AMR relations to UMR roles, since multiple UMR interpretations may be possible depending on contextual and semantic factors. Consequently, manual analysis was required to ensure consistent role assignment. The next section describes our manual adaptation of these rules for the Persian AMR (PAMR) corpus.

4.2. Manual Split-Role Conversion for Persian

A single expert annotator with a background in computational linguistics and Persian semantics systematically re-labeled every AMR edge that corresponds to a split role. For each occurrence, the annotator examined the full Persian sentence and animacy feature to select the correct UMR label. To assess the reliability of manual split-role annotation, an additional linguist independently annotated a subset of 100 instances containing split-role phenomena. Inter-annotator agreement was measured using Cohen's κ, yielding a score of $\kappa = 0.845$. Disagreements were subsequently resolved through expert adjudication by a third linguist following the official UMR guidelines. The adjudicated labels constitute the final gold annotations. Most disagreements occurred between roles such as :goal vs. :recipient and :source vs. :material, reflecting subtle distinctions encoded in UMR role definitions. The next sections outline the conversion of AMR structures to UMR role representations and we demonstrate the mapping process through concrete examples drawn from the Persian data.

[3] https://github.com/ufal/umr-guidelines/blob/master/guidelines.md

4.2.1. Mapping Modifier Role

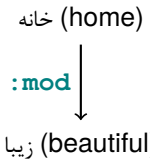
Based on the UMR guidelines, the :mod role (modifier) is a general-purpose semantic label used to indicate a property or attribute of an entity or event. It is applied as a default when the modification does not fit a more specific semantic role, such as :place or :time. The following list shows the categorization of words that can be annotated with this label:

- Adjectives (e.g., *large* in "large house")
- Adverbs denoting manner (e.g., *quickly* in "run quickly")
- Demonstratives (e.g., *this* in "this book")
- Participial modifiers (e.g., *broken* in "broken window")
- Attributive nouns (e.g., *steel* in "steel bridge")

In (1), there is an example of an adjective that has the same :mod label in both AMR and UMR.[4]

(1) زیبا خانه
zibâ xâne-ye
beautiful home
'beautiful home'

The AMR and UMR graphs of the above example are identical:



4.2.2. Mapping Destination Role

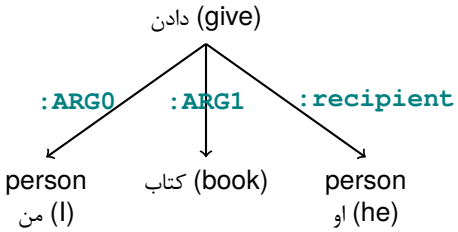
A significant refinement in UMR concerns the mapping of the AMR role :destination. In AMR, :destination is a general role denoting any endpoint of motion or transfer. UMR, informed by cross-linguistic typology, splits this role into two more semantically precise roles based on animacy and the nature of the transfer: :goal for inanimate endpoints and :recipient for animate endpoints that gain possession. This distinction allows UMR to more accurately capture the semantic roles of participants across diverse linguistic constructions. Examples (2) and (3) illustrate how to map AMR :destination role to UMR roles:

[4] Examples are presented with Persian text in the first line, transcription in the second line, English gloss in the third line, and English translation in the fourth line. Note that the words are ordered right-to-left, following the Persian writing system.

(2)

دادم	او	به	کتاب را
dâdam	u	be	ketâb râ
gave	him	to	the book

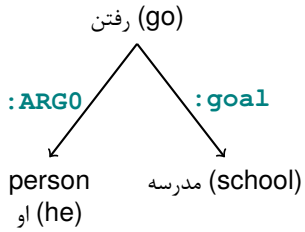
'I gave the book to him'



(3)

رفت	مدرسه	به	او
raft	madrese	be	u
went	school	to	he

'He went to school'



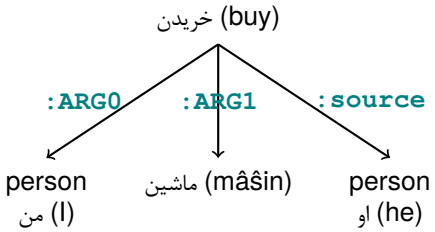
4.2.3. Mapping Source Role

In UMR, the :source role specifically marks an animate entity from which a theme separates or originates. It is distinct from the inanimate :start (location) and :material (composition) roles, reflecting a typologically-motivated split of AMR's general :source based on animacy. This role is typically applied to nouns and pronouns denoting people, animals, or organizations. The following examples show this role mapped to UMR roles.

(4)

خریدم	او	از	ماشین را
xaridam	u	az	mâšin râ
bought	him	from	the car

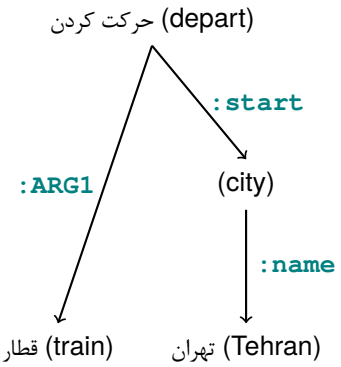
'I bought the car from him'



(5)

حرکت کرد	تهران	از	قطار
harekat kard	Tehrân	az	qatâr
departed	Tehran	from	the train

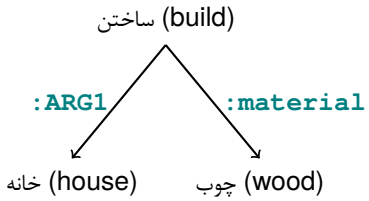
'The train departed from Tehran'



(6)

ساخته شد	چوب	از	خانه
sâxte ŝod	ĉub	az	xâne
was built	wood	from	the house

'The house was built of wood'



4.2.4. Mapping Consist-of Role

In the UMR schema, the AMR role :consist-of is decomposed into three semantically more precise roles based on the nature of the part-whole relationship:

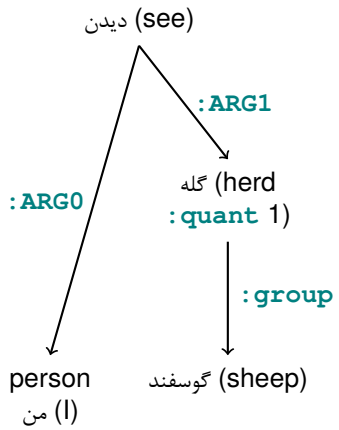
- :group for collections of animate entities (e.g., team, committee, herd; the relation is directed from the group concept to the member concept)
- :part for integral components of an inanimate whole (e.g., chapter, branch, piece; the relation is directed from the whole to the component)
- :material for the constituent substance an entity is composed of (e.g., wood, steel, water; the relation is directed from the entity to the material)

The examples of (7) and (8) show how to map the :consist-of tag to the UMR :group and :part labels (in this case, :part is inverted to :part-of because the whole is presented as a modifier). Moreover, some instances of :consist-of denote raw material, and like with :source, we map them to the UMR :material role (see the example 6).

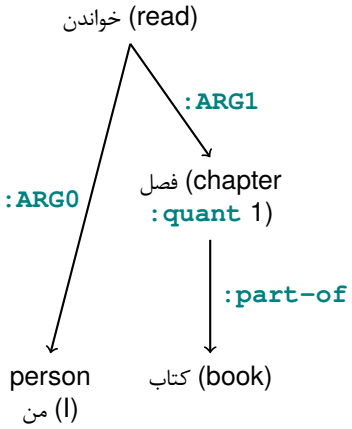
(7)

دیدم	گوسفند	گله	یک
didam	gusfand	galle	yek
saw	sheep	herd	a

'I saw a herd of sheep'



(8) خواندم کتاب را فصل یک
xândam ketâb râ fasl-e yek
read the book chapter a
'I read a chapter of the book'

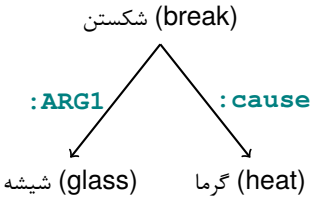


4.2.5. Mapping Cause Role

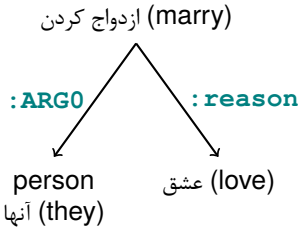
In the UMR schema, the representation of causality is refined through distinct strategies depending on the syntactic and discourse context. The AMR :cause role is split into :cause for inanimate entities that directly bring about an event and :reason for entities that motivate an action. This distinction enhances semantic precision. Examples (9) and (10) show causative constructions. Furthermore, UMR handles inter-sentential causality differently from intra-sentential causality. When a causal relationship links two separate sentences within a corpus, the abstract concept cause-01 is invoked to reify the relation, with :ARG1 pointing to the causing event and :ARG2 to the resulting event (see example 11).

It should be noted that the UMR guidelines also define a :causer role, to be used in causative constructions such as "Grandmother made the kid drink the water." However, that role is defined as Stage 0 role, to be used in languages that have no lexical resources with rolesets for predicates. Persian UMR qualifies as Stage 1 project, as the source AMR annotation already contains numbered PropBank-style arguments. Therefore, causative agents should be annotated as :ARG0 and would not result from conversion of :cause.[5]

(9) شکست شیشه گرما به‌خاطر
šekast šiše garmâ be xâter-e
broke glass heat because of
'The glass broke because of the heat'



(10) ازدواج کردند عشق به خاطر آنها
ezdevâj kardand ?eŝq be xâter-e ânhâ
married love because of they
'They got married because of love'



(11) a. طغیان کرد رودخانه
toqiyân kard rudxâne
flooded river
'The river flooded.'

b. نابود شدند محصولات و
nâbud ŝodand mahsulât va
was destroyed crops and
'And, the crops were destroyed'

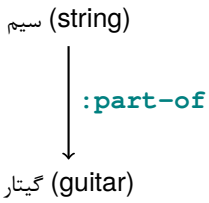
4.2.6. Mapping Part Role

Within the UMR schema, the AMR role :part represents a specific case within the broader reclassi-

[5]We did not encounter such sentences in the data, though.

fication of part-whole relationships. While its surface mapping is deterministic—AMR :part maps directly to UMR :part—it is conceptually grouped under the umbrella of "split roles." This classification arises because :part is one of three possible outcomes (alongside :group and :material) resulting from the decomposition of the more general AMR role :consist-of. Thus, it is considered a "split role" not due to a change in its label, but because it is a key member of the new, more precise set of roles that collectively replace the underspecified AMR :consist-of. Example (12) illustrates the usage of :part annotation in UMR.

(12) گیتار سیم
gitâr sim-e
guitar string
'Guitar string.'



The next section will discuss a quantitative profile of the final Persian-UMR dataset and highlight the key role-mapping results.

5. Corpus Statistics and Analysis

PAMR contains 1,562 manually annotated sentences of the Persian translation of "The Little Prince" story. Table 1 summarizes the main statistics of the dataset.

Feature	Value
Number of sentences	1,562
Number of unique words	3,520
Number of tokens	14,427
The shortest sentence length	1
The longest sentence length	65

Table 1: Corpus statistics for Persian AMR.

The quantitative analysis of the annotated corpus reveals critical insights into the complexity of converting AMR roles to UMR. Table 2 highlights the varying degrees of determinism and ambiguity inherent in the mapping process.

According to Table 2, the AMR role :mod maps deterministically to UMR :mod in all 1,236 instances (100% of cases), meaning that the annotator did not identify any of the instances as being integral event participants. This is not too surprising given the nature of :other-role: it is intended as a last resort, but at present it is not used (needed) in any of the 8 languages in UMR 2.0.

AMR Role	#AMR	UMR Role	#UMR
:mod	1,236	:mod	1,236
:destination	5	:goal	5
		:recipient	0
:cause	2	:cause	75
cause-01	126	**cause-01**	38
		:reason	11
		:purpose	2
		:result	1
		:condition	1
:source	27	:source	17
		:start	7
		:cause	2
		:manner	1
:part	76	:part	76
:consist-of	13	:part	3
		:material	5
		:unit	3
		:group	2
Total	1485	**Total**	1485

Table 2: Mapping of Persian AMR to UMR roles. Note that **cause-01** is an abstract concept rather than a role; switching between it and a role involves changes in the graph structure.

Moreover, as explained before, the conversion of the :part role is deterministic and all 76 AMR :part instances are unchanged in UMR.

Conversely, other roles demonstrate significant splitting, necessitating disambiguation. The limited sample of :destination (5 instances) maps entirely to :goal, suggesting a potential bias in the data sample towards inanimate endpoints. A larger corpus would be required to observe the expected split with :recipient for animate entities.

The :source role is observed as a complex non-deterministic role, as it mapped to :source and :start based on Section 4.2.3. However, while analyzing the data, we encountered unexpected[6] mapping of the :source role to :cause, :location, and :manner. For example, in sentence (13) from the Persian AMR corpus, the word 'همین' (this) is annotated as :source but it was converted to :cause in UMR.

(13) است همین از غصه‌م
ast hamin az qosse-am
is this from my sorrow
'My sorrow is from this'

In the case of the :cause role, we can see that the majority of mappings are to :cause and

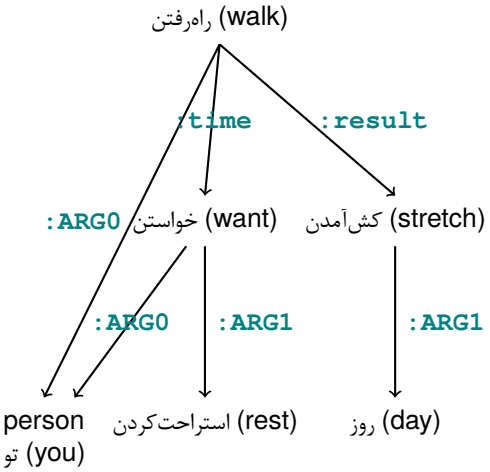
[6]By 'unexpected' we mean that it was not observed by Post et al. (2024) as an option in English, hence we did not list it in Section 4.1.

then cause-01. Moreover, contrary to the English conversion rules, there were some `:cause` annotations in the Persian AMR corpus that had to be mapped to `:result` (example (14)), `:condition` (example (15)), and `:purpose` (example (16)) in UMR.

(14)

کش‌خواهدآمد	روز	راه‌برو	می‌خواهی	استراحت	هروقت
keŝ xâhad âmad	ruz	râh boro	mixâhi	esterâhat	har vaqt
will stretch	day	walk	want	rest	when ever

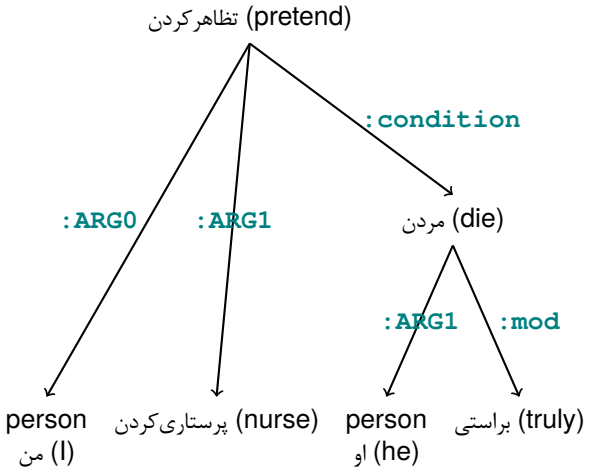
‘Whenever you want to rest, walk.
Then the day will stretch.’



(15)

می‌مرد	براستی	وگرنه	تظاهر می‌کنم	به پرستاری
mimord	be râsti	vagar na	tazâhor mikonam	be parastâri
die	truly	other wise	pretend	to nurse

‘I pretend to nurse.
Otherwise, He would truly died.’



(16) a.

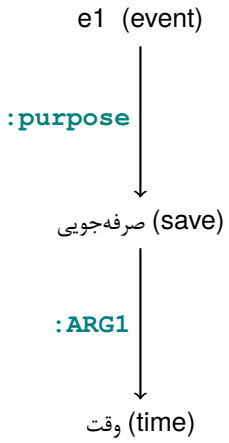
می‌فروشی؟	قرص	چرا
miforusî	qors	ĉerâ
sell	pills	why

‘Why do you sell pills?’

b.

در وقت	صرفه‌جویی	برای
dar vaqt	sarfe juyi	barây-e
time	save	to

‘To save time.’



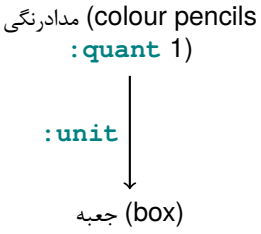
Furthermore, in cases where there is insufficient evidence to determine if the causer is animate or motivational (e.g., the `:ARG0` is unknown in the AMR graph), the annotator often defaulted to the more general role `:cause`. This conservative strategy is employed because `:cause` is semantically broader, it does not assume intention (which would be required for `:reason`), and it can accommodate both physical forces and abstract, non-motivational causes. For instance, in a sentence like “The event was caused by [an unknown factor],” where the factor’s nature is unclear, `:cause` serves as a safe, neutral default.

For the `:consist-of` role, we faced a similar issue as well and some AMR data were mapped to the `:unit` role. This occurred in contexts like “a box of colour pencils,” where the annotator interpreted the relationship not as material composition (`:material`) or integral parthood (`:part`), but primarily as quantification and containment—the box serves as a quantitative unit for the pencils.

(17)

مدادرنگی	جعبه	یک
medâd rangi	ja?bey-e	yek
colour pencils	box of	a

‘A box of colour pencil’



6. Conclusion and Future Work

This paper presented the first Persian UMR corpus created by systematically converting an existing Persian AMR resource using rule-based mapping combined with manual annotation. Like its AMR source, the UMR corpus is freely available for research.[7] We focused on split semantic roles, which present the greatest challenge for AMR to UMR conversion because a single AMR role can correspond to multiple UMR roles depending on context. This study provides the critical foundation for a comprehensive Persian UMR infrastructure. The immediate next step can be the systematic expansion of this initial corpus. This entails annotating the full range of UMR tags not covered in this study, particularly those for document-level semantics such as coreference chains, event temporality, and cross-sentence modality; also, the alignment between surface tokens and UMR nodes cannot be extracted from AMR data and will have to be obtained using other methods.

7. Ethics Statement

We are not aware of any ethical concerns related to this work. The corpus was manually annotated as part of academic research. In addition to the primary annotator, independent linguistically trained annotators contributed to annotation validation and agreement assessment. All annotation work was conducted voluntarily for research purposes, and no sensitive or personal data were involved, as the corpus is based on a publicly available literary text.

8. Limitations

At the present stage, the resource does not provide the full range of annotations specified in UMR guidelines; we list the main omissions in Future Work. This is a limitation, but given the complexity of UMR, the same limitation currently applies even to some datasets in the official UMR 2.0 release.

9. Acknowledgements

The work described herein was supported by the Charles University, project GAUK No. 394625. The second author was also funded by *LINDAT/CLARIAH-CZ* (Project No. LM2023062) of the Ministry of Education, Youth, and Sports of the Czech Republic. This research was also partially supported by SVV project number 260 821.

[7] http://hdl.handle.net/11234/1-6135

10. Bibliographical References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Julia Bonn, Skatje Myers, Jens E. L. Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H. Martin, Alexis Palmer, Martha Palmer, James Pustejovsky, Zdenka Urešová, Rosa Vallejos, and Nianwen Xue. 2023a. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 74–95, Washington, D.C. Association for Computational Linguistics.

Julia Bonn, Skatje Myers, Jens EL Van Gysel, Lukas Denk, Meagan Vigus, Jin Zhao, Andrew Cowell, William Croft, Jan Hajič, James H Martin, et al. 2023b. Mapping AMR to UMR: Resources for adapting existing corpora for cross-lingual compatibility. In *TLT 2023-21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023), Proceedings of the Conference*, volume 21. Association for Computational Linguistics.

Qin Gao and Stephan Vogel. 2011. Corpus expansion for statistical machine translation with semantic role label substitution rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 294–298, Portland, Oregon, USA. Association for Computational Linguistics.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, Alfio Gliozzo, Sairam Gurajada, Hima Karanam, Naweed Khan, Dinesh Khandelwal, Young-Suk Lee, Yunyao Li, Francois

Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G P Shrivatsa Bhargav, and Mo Yu. 2021. Leveraging Abstract Meaning Representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894, Online. Association for Computational Linguistics.

Simin Karimi. 2008. *A minimalist approach to scrambling: Evidence from Persian*, volume 76. Walter de Gruyter, Berlin, New York.

Gh. Karimi-Doostan. 1997. *Light Verb Constructions in Persian*. Ph.D. thesis, Essex University, England.

Paul R Kingsbury and Martha Palmer. 2002. From treebank to PropBank. In *LREC*, pages 1989–1993.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending VerbNet with novel verb classes. In *LREC*, pages 1027–1032. Genoa.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086, Denver, Colorado. Association for Computational Linguistics.

Azadeh Mirzaei and Amirsaeid Moloodi. 2016. Persian proposition bank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3828–3835.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 1130–1139.

Claire Benet Post, Marie C McGregor, Maria Leonor Pacheco, and Alexis Palmer. 2024. Accelerating UMR adoption: Neuro-symbolic conversion from AMR-to-UMR with low supervision. In *Proceedings of the Fifth International Workshop on Designing Meaning Representations@ LREC-COLING 2024*, pages 140–150.

Mohammad Sadegh Rasooli, Manouchehr Kouhestani, and Amirsaeid Moloodi. 2013. Development of a Persian syntactic dependency treebank. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 306–314.

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.

Zacchary Sadeddine, Juri Opitz, and Fabian Suchanek. 2024. A survey of meaning representations – from theory to practical utility. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2877–2892, Mexico City, Mexico. Association for Computational Linguistics.

Ziyi Shou, Yuxin Jiang, and Fangzhen Lin. 2022. AMR-DA: Data augmentation by Abstract Meaning Representation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3082–3098, Dublin, Ireland. Association for Computational Linguistics.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Jan Štěpánek, Daniel Zeman, Markéta Lopatková, Federica Gamba, and Hana Hledíková. 2025. Comparing manual and automatic UMRs for Czech and Latin. In *Proceedings of the Sixth International Workshop on Designing Meaning Representations*, pages 1–12, Prague, Czechia. Association for Computational Lingustics.

Reza Takhshid, Razieh Shojaei, Zahra Azin, and Mohammad Bahrani. 2022. Persian abstract meaning representation. *arXiv preprint arXiv:2205.07712*.

Nasim Tohidi, Chitra Dadkhah, Reza Nouralizadeh Ganji, Ehsan Ghaffari Sadr, and Hoda Elmi. 2024. Pamr: Persian abstract meaning representation corpus. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3):1–20.

Jens EL Van Gysel, Meagan Vigus, Jayeol Chun, Kenneth Lai, Sarah Moeller, Jiarui Yao, Tim O'Gorman, Andrew Cowell, William Croft, Chu-Ren Huang, et al. 2021. Designing a uniform meaning representation for natural language processing. *KI-Künstliche Intelligenz*, 35(3):343–360.

11. Language Resource References

Bonn, Julia and Bonial, Claire and Buchholz, Matt and Cheng, Hsiao-Jung and Chen, Alvin and Chen, Ching-wen and Cowell, Andrew and Croft, William and Denk, Lukas and Elsayed, Ahmed and Fučíková, Eva and Gamba, Federica and Gomez, Carlos and Hajič, Jan and Hajičová, Eva and Havelka, Jiří and Havenmeier, Loden and Kilgore, Ath and Kolářová, Veronika and Kučová, Lucie and Lai, Kenneth and Li, Bin and Li, Jingyi and Lopatková, Markéta and MacGregor, Marie and Mikulová, Marie and Mírovský, Jiří and Nedoluzhko, Anna and Myers, Skatje and Novák, Michal and O'Gorman, Tim and Pajas, Petr and Palmer, Alexis and Palmer, Martha and Panevová, Jarmila and Post, Benét and Pustejovsky, James and Sgall, Petr and Song, Jialin and Song, Li and Ševčíková, Magda and Štěpánek, Jan and Urešová, Zdeňka and Sun, Haibo and Sun, Yao and Vallejos Yopán, Rosa and Van Gysel, Jens and Vigus, Meagan and Wright-Bettner, Kristin and Wu, Jiawei and Xue, Nianwen and Xing, Dan and Xu, Keer and Xu, Zhixing and Yue, Liulu and Zeman, Daniel and Zhao, Jin and Zikánová, Šárka and Žabokrtský, Zdeněk. 2025. *Uniform Meaning Representation 2.0*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL). PID http://hdl.handle.net/11234/1-5902.

Takhshid, Reza and Azin, Tara and Shojaei, Razieh and Bahrani, Mohammad. 2021. *Persian AMR Dataset*. GitHub. PID https://github.com/Persian-AMR/Dataset.

Regression-Tested Compositional Semantics: A Graphical Development Environment for Glue and Description-by-Analysis

Mark-Matthias Zymla, Kascha Kruschwitz
University of Konstanz
{mark-matthias.zymla|kascha.kruschwitz}@uni-konstanz.de

Abstract

We present a platform for developing compositional semantic annotations for formal syntactic representations that allows users to interact with and explore annotations and to track their progress and quality. For this, we provide several forms of visualizations and take inspiration from research in linguistic treebanking. Thus, we contribute to the development of formal semantic parsers and corresponding meaning banks. The system is designed with a regression testing paradigm in mind and provides support for NLI so that the created semantic resources can be developed and validated in a task-driven environment. We defend this paradigm in comparison to modern approaches to semantic parsing that are mainly evaluated on the basis of gold standard annotations.

1. Introduction

Formal semantic parsing has received little attention in the wake of machine-learning and deep learning driven approaches. This is despite the fact that such approaches suffer from various shortcomings. For example, they do not really solve the problem of the inability to properly generalize to out-of-domain data, sometimes even inventing completely nonsensical analyses (see, e.g., Haug et al. 2023; Zhang et al. 2022, 2025). However, their continuing popularity suggests that the benefits of these methods outweigh their flaws relative to formal approaches.

We believe that this is not entirely justified, as formal approaches provide a level of transparency and reliability that is not matched by quantitative methods.[1] Thus, we provide a system for deriving semantic parsers built around formal methods in computational linguistics.

Our system deviates from popular approaches in semantic parsing, concretely, the comparison of analyses against a gold standard (see Zhang et al. 2025 for an overview of methods). Rather, we follow a task-driven approach, evaluating semantic representations in the context of automated reasoning, in particular, the natural language inference (NLI) task (MacCartney and Manning, 2009), an ingredient of many NLP tasks (Bos, 2009).

According to Bos (2009), building a platform for developing and evaluating formal semantic parsers in this way is a major undertaking requiring a level of expertise in various disciplines, such as formal logic, computer science, linguistics, and NLP. This is certainly true. Thus, we believe it is crucial to focus on making computational linguistic resources accessible, such that linguistic analyses can be developed with less concern for the more technical aspects of semantic parsing.[2] We present an attempt at this by providing a browser-based application for developing compositional semantic analyses derived from syntactic parses, such as, for example, illustrated for the grammars developed in the framework of Lexical Functional Grammar in Zymla et al. (2025b) or for Universal Dependency analyses in (Findlay et al., 2023).[3]

Our application supports prototyping individual analyses and *regression testing* (Chatzichrisafis et al., 2007), enabling users to test analyses at a larger scale. However, we acknowledge the flaws of formal semantic parsing as they become more critical as coverage of a grammar is extended. We hypothesize that testing these boundaries can shed more light on the linguistic aspects of reasoning. Thus, we propose an incremental semantic parsing paradigm that matches particular semantic analyses with particular inference patterns, such that these analyses are covered by separately applicable rules. More concretely, we assume that building multiple interoperable medium-sized grammars, targeting specific inference patterns, is more beneficial than a broad coverage semantic parser built on a single set of grammatical and semantic rules. The tools presented here serve to foster such grammars and to generate corresponding meaning banks that may be beneficial in downstream tasks.

This paper is structured as follows: In section 2, we lay out the necessary background. We present our system in section 3, and in section 4 we describe incremental parsing as our ideal vision for using the system. Section 6 concludes.

[1] We do not mean reliability in the sense of robustness but rather in the sense that, e.g., interactions between semantic operators can be deterministically predicted.

[2] As is achieved in a number of linguistic annotation tools, e.g., (Stenetorp et al., 2012; Liu et al., 2017).

[3] The system is available at https://github.com/Mmaz1988/xleplusglue/tree/dmr2026_regression_testing. Using XLE grammars requires access to XLE binaries for Linux managed via the University of Konstanz. Please contact the authors for assistance.

Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026, pages 183–197
11 May 2026.

2. Background

The main goal of this paper is to present a tool that facilitates semantic parsing and the generation of corresponding tree-, or rather, meaning banks, based on formal linguistics. Zymla et al. (2025b) present a qualitatively tested approach to formal semantic parsing for NLI based on XLE+Glue, a system developed in the context of Lexical Functional Grammar (LFG; Bresnan et al. 2015; Dalrymple 2001, 2023). As described there and in Zymla et al. (2025a), XLE+Glue combines three resources: i) Linguistic Graph Expansion and Rewriting (LiGER), a resource for description-by-analysis, ii) the Glue semantics workbench (GSWB), a resource for calculating Glue derivations, and iii) an interface to the Vampire theorem prover.[4] Thus, they provide a workflow that allows for semantic parsing that is tested in the context of natural language inference (MacCartney and Manning, 2009). These tools are built around Glue semantics, a formalization of the syntax/semantics interface, which we introduce in the next section.

2.1. Glue and description-by-analysis

Glue semantics or Glue has recently received renewed interest (Meßmer and Zymla, 2018; Dalrymple et al., 2020) and the underlying tools continue to mature (Zymla et al., 2025a). They have been used, for example, to model aspects of the prosody-meaning mapping (Butt et al., 2024) or to verify complex formal analyses (Przepiórkowski and Patejuk, 2023).

The main idea of Glue is that compositionality is guided by the resource-sensitive linear logic (Dalrymple, 1999), such that the meaning of an expression and the way in which it interacts with other pieces of meaning are captured separately. The corresponding basic semantic representations are called meaning constructors.[5] Importantly, Glue is compatible with various syntactic and meaning representations, providing a very general view of the syntax/semantics interface that captures many of its fundamental aspects, while abstracting away from potential syntactic and semantic idiosyncrasies of specific formalisms.[6]

There exist at least two ways to introduce meaning constructors to a syntactic framework. The first is called *co-description*, and is somewhat LFG-specific in that semantic representations are stored lexically (or sometimes, as part of syntactic rules; Dalrymple 2001). The tools presented here mainly concern the second possibility: *description-by-analysis* (DBA), which models the syntax/semantics interface as rewrite rules applied to syntactic structures (e.g., Andrews 2008). Figure 1 provides an example of how to use DBA rules to add compositional semantic information to a UD analysis. As shown there, rules are applied sequentially to some input, adding additional structure and introducing meaning constructors. These can then be used for semantic composition.

Bobrow et al. (2007); Crouch and King (2006); Crouch (2005) provide early approaches to semantic parsing with DBA based on LFG but do not employ Glue semantics per se. Lev (2007) more closely resembles the present approach in that it uses DBA to produce Glue meaning constructors. It serves as inspiration for XLE+Glue (Dalrymple et al., 2020; Zymla et al., 2025a), which we build upon in this paper. More recent approaches to DBA (Findlay et al., 2023; Zymla et al., 2025b) have also been built around (parts of) these tools. All these approaches to DBA use graph rewriting; thus, in principle, DBA can be applied to any linguistic representations that can be cast as directed graphs. Consequently, it synergizes well with the flexibility of Glue.

2.2. Semantic parsing for meaning banking

More generally, our tools contribute to grammar development (Butt et al., 1999) and treebanking (Marcus et al. 1993; or more specifically, meaning banking), which are essential aspects of traditional CL methods and have heavily influenced the design of linguistic annotations in the past.

Meaning banks exist for various formalisms, such as the DeepBank for HPSG (Flickinger et al., 2012) or the Groningen meaning bank for a version of DRT (Abzianidze et al., 2020). Further, the abstract meaning representation (AMR) effort provides annotations via the LDC (Knight et al., 2020) and unified meaning representations (UMR) built on the AMR formalism, providing corresponding data sets (Bonn et al., 2024). However, only a small portion of modern meaning banks still rely on formal parsers for data annotation, despite many meaning banks originating from these efforts (e.g., the predecessor of the PMB, the Groningen mean-

[4] The name XLE+Glue stems from the fact that these tools have been developed mainly in the context of providing semantic representations for the Xerox linguistics environment (XLE; Crouch et al. 2017). However, the system has been extended to work with UD parses by Stanza, and, more generally, other syntactic parsers may be substituted.

[5] We do not explain the details of Glue semantics here. A comprehensive overview can be found in Asudeh (2023), or the recent LFG handbook (Dalrymple, 2023). For a practical introduction, see Zymla et al. 2025b.

[6] For example, Glue semantics maintains the type-driven analysis of quantifiers, without requiring additional syntactic mechanisms, like quantifier raising, or storage-based methods (Dalrymple et al., 1999).

(1) Every dog sniffed **a tree**.

```
#n UPOS NOUN & #n LEMMA %l & #n SEM #s ==> #s GLUE [/x_e.%l(x)] : (#s_e -o #s_t).

#d DEFINITE Ind & #d ^(DET) #n SEM #s & #n ^(%) #p SEM #q  ==>
#d SEM #t & #t GLUE [/P_<e,t>.[/Q_<e,t>.Ex_e[P(x) \& Q(x)]]] :
                        ((#s_e -o #s_t) -o ((#s_e -o #q_t) -o #q_t)).
```

$$\frac{\begin{array}{cc}\lambda x.tree(x) : & \lambda P.\lambda Q.\exists x[P(x) \wedge Q(x)] : \\ 11_e \multimap 11_t & (11_e \multimap 11_t) \multimap (11_e \multimap 7_t) \multimap 7_t\end{array}}{\begin{array}{c}\lambda Q.\exists x[tree(x) \wedge Q(x)] : \\ (11_e \multimap 7_t) \multimap 7_t\end{array}}$$

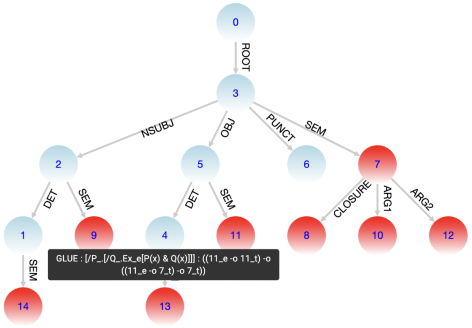


Figure 1: **Indefinite NP semantics example via LiGER:** DBA rules identify subgraphs in a UD parse to expand with semantic information. The first rule gives any matching dependent the semantics of a property (type $< et >$). The second rule identifies this property as the restrictor of the quantifier (identified by the variable #s = 11) and the semantics of the verb as the scope (identified by #q = 7). Together, they form the semantics for the quantifier attached to the corresponding node #t (= 13). They are combined to create a canonical quantifier of type $< et, t >$.

ing bank, relied on CCG parsers). Although LFG provides a variety of syntactic treebanks (e.g., Sulger et al. 2013), there do not yet exist correspondingly extensive meaning banks, making the present tools particularly interesting for this framework.

While we do not directly contribute to this research in the present paper, we assume that formal semantic parsing can lay the foundation for neuro-symbolic approaches which not only strive for best performance but also bridge the gap between formal and computational linguistics, by, for example, testing whether language models can efficiently learn from formal linguistic annotations (see, e.g., Lindemann et al. 2024; Li et al. 2021; Strubell et al. 2018). To achieve the necessary large-scale datasets, sufficiently powerful and robust parsers are needed. Importantly, these parsers should reliably capture complex syntactic and semantic interactions, particularly those that cannot be easily derived from surface patterns (e.g., scope ambiguities). However, permitting ambiguities can make formal parsing challenging (Bunt, 2008); thus, we borrow techniques from efforts to link grammar development and tree-banking to enable efficient management of ambiguities. Particularly, we build on the discriminant system presented in Rosén et al. (2006). Overall, we provide tools that allow for the generation of larger, more detailed datasets that can be used in the way envisioned above.

3. Supporting the development of compositional semantic representations

In this section, we present our application for supporting the development of compositional semantic representations via DBA using XLE+Glue resources. For this, our system provides possibilities to inspect and explore individual analyses and to perform regression testing, which allows for scaling up of the semantic parsers developed in this system. Before describing the system, we briefly discuss the underlying formal modeling.

3.1. Linguistic representations

XLE+Glue procedurally provides semantic parsing based on LFG. However, it also provides a formal representation of the form-to-meaning mapping, linking several linguistic annotation layers containing different levels of linguistic information, for example, c(onstituent) structure, f(unctional) structure, and s(emantic) structure. We uniformly store these different layers as interconnected or layered graphs (see Figure 2). Each graph is associated with a grammar and a sequence of rewrite (DBA) rules used to produce the graph. Thus, they are essentially a way of representing LFG's projection architecture (e.g., Asudeh 2006). Using a uniform graph format allows the representations to be easily extended with additional annotations (also inspired by, e.g., Ide and Bunt 2010). However, layered graphs can become very complex (see Figure 9). Thus, modularizing the ways of interacting with them is crucial and a goal of the present system.

To use our representations for NLI, we translate the semantic representation into the TPTP format for first-order logic (Sutcliffe et al., 2006), conjoining multiple premises to a single premise. Additional axioms and meaning postulates are added if necessary. We then use the Vampire theorem prover (Kovács and Voronkov, 2013) to derive *consistency* and *informativity* labels, which are heuristically mapped onto NLI labels (YES, NO, UNKNOWN). This is explained in detail in Zymla et al. (2025b).

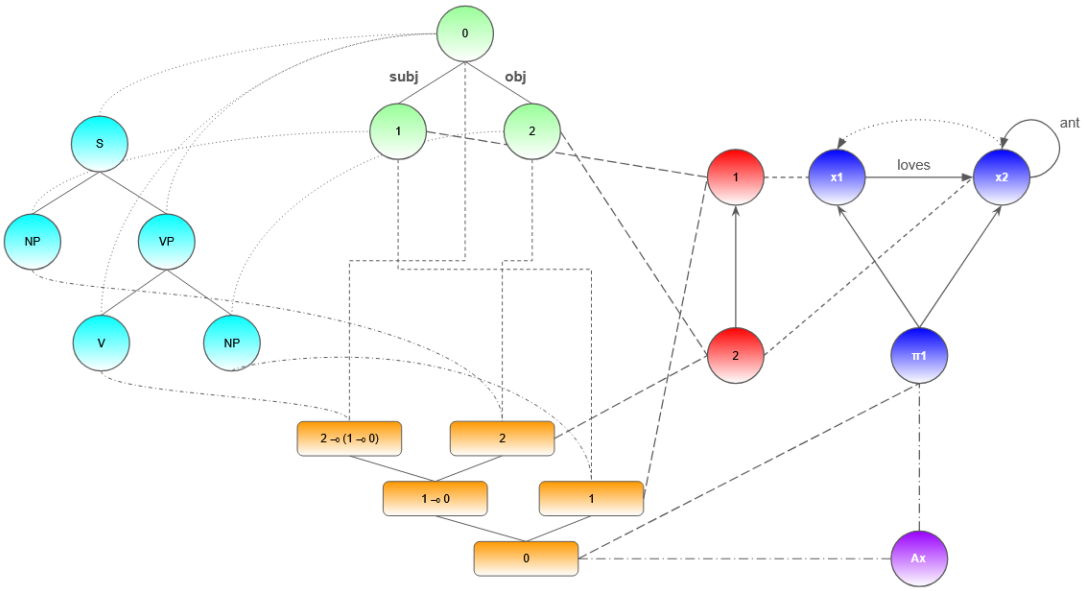


Figure 2: **A schematic layered XLE+Glue graph:** The graph illustrates an XLE+Glue analysis of the sentence *Sam loves herself*. Teal and green correspond to LFG's c(onstituent)- and f(unctional)-structure. Orange is the semantic composition graph, and red corresponds to s(emantic)-structure. The DRT representation is given in blue, and additional axioms are stored in a set associated with the purple node.

3.2. Individual analysis interface

Our application provides three different views: a view for the individual analysis of the mapping from syntactic input to semantics, a view for exploring (NLI) testsuites, and an explorative chat view for testing the NLI component qualitatively (see Zymla et al. 2025b). Here, we focus only on the first two as novel contributions we make to the system. Each view consists of multiple components that can be independently hidden, allowing users to focus only on particular tasks (e.g., designing rules, disambiguation of semantic representations, or debugging).[7] The individual view consists of two major components: a view for writing DBA rules and a view for inspecting Glue proofs, which act as user interfaces for LiGER and for the GSWB, respectively (see Figures 9 and 10 in the appendix).

The *DBA view* provides an editor window with syntax highlighting and visualizes the result of an annotation in an interactive graph window (a sample graph is displayed in Figure 1). During the derivation, all applied rules are collected in a list (i.e., a rule history), which is then displayed in the applied rules view (see Figure 3). Here, we provide the functionality to reapply rules incrementally to see how they affect the derivation. This serves the design and debugging of DBA rules in context. A common use of this view is to ascertain that rules are properly contextualized, meaning that rules

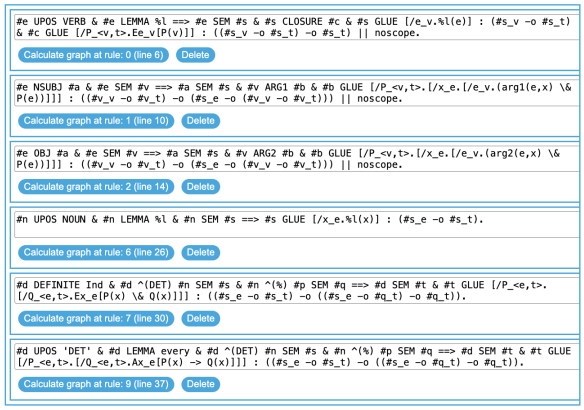


Figure 3: **Applied rules view (individual):** This view lists rules in the order they have been applied. It allows users to inspect the behavior of individual rules by calculating intermediate annotations.

interact with each other as anticipated. As DBA rules are processed sequentially, they (sometimes) form application chains, i.e., a sequence of rules where the output of a previous rule is required for the input of a subsequent rule. Wrongly modifying a rule in such a chain can easily break the whole analysis. Thus, the rule history helps track application chains, but also indicates whether individual rules have been applied as intended.

The *Glue view* presents the results of the DBA annotation in terms of added meaning constructors. It is also the main interface to the GSWB, which is used for calculating the semantic derivations. Cor-

[7] Comprehensive figures are provided in the appendix. In the paper, we only present aspects that have not been introduced elsewhere.

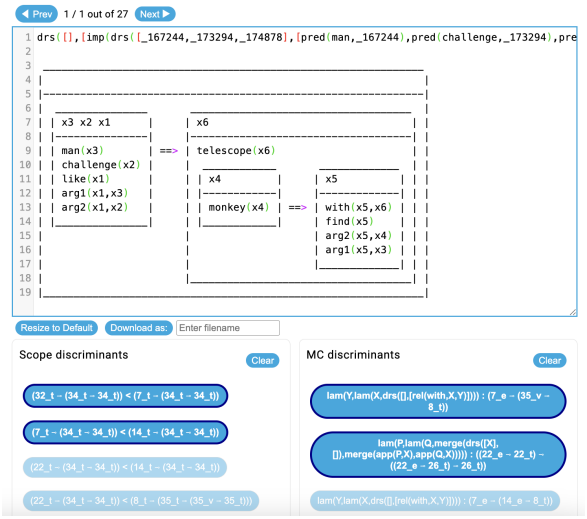


Figure 4: **Discriminant view:** Displays the results of a semantic composition, allowing users to disambiguate its results using two types of discriminants to filter out solutions that do not match them.

respondingly, it visualizes these derivations, such that they are represented as a graph or in a natural deduction format (Zymla et al., 2025a). For this paper, we added a discriminant component to this view that allows users to systematically disambiguate analyses. Figure 4 presents an example based on the following sentence:

(2) Every man who likes a challenge found every monkey with a telescope.

As indicated in the Figure, our parser produces twenty-seven solutions for this sentence, permitting all possible scopings between the different quantifiers as well as different interpretations of the prepositional phrase. The sentence is disambiguated to a reading where each man uses a possibly different telescope to find all the monkeys. First, we use the *MC discriminants* that indicate differences in the used meaning constructors for different solutions to only pick solutions where the telescope is used as an instrument, i.e., is associated with an event variable. Furthermore, we ensure that the relative clause is interpreted locally as its scope is fixed (partially) syntactically. Then, we can use *scope discriminants* to fine-tune the scopings between the individual quantifiers that are determined semantically to arrive at the desired semantic representation.

As the name suggests, MC discriminants are calculated based on different sets of meaning constructors that can be produced by a grammar for a given sentence. However, each set may still correspond to multiple solutions. Thus, these discriminants provide a less granular separation of results, whereas scope discriminants are calculated during the derivation for each individual solution, making them more fine-grained. After the derivation, equivalence classes for all potential discriminants are calculated to reduce the total number of discriminants. Sometimes, the same scope-taking operator may be instantiated differently in a different meaning constructor set, which can lead to malformed discriminants. Thus, we impose a disjointness condition on discriminants, such that each discriminant is uniquely applicable.

The discriminants behave commutatively, such that individual decisions are not dependent on each other. However, we keep track of choices to make uninformative discriminants unavailable during the selection process, allowing users to quickly reduce the number of readings. Importantly, this system not only allows for the reduction to a single solution but also provides finer control over which ambiguities should potentially be retained for a given derivation.

3.3. Resolving discriminants for NLI

We provide two different inputs for regression testing: i) sentence-based items and inference-based items, where inference-based items consist of a list of premises, a conclusion, and a gold label (illustrated in Figure 6). For both, we provide a list view for inspecting individual results, highlighting failed parses and NLI label mismatches. Further, sentences can be disambiguated using the discriminant view. This can be essential for eliminating uncertainty in NLI testing and is accordingly implemented as a human-in-the-loop process (similar, in spirit, to He et al. 2016). Consequently, the semantic parsing process can be paused after semantic composition to select discriminants. The following paragraph explains the need for this.

Figure 5 shows an inference result based on ambiguous inputs. It is based on example (3). There, multiple quantifiers and negation lead to a number of different possible interpretations.

(3) Every boxer with an injury lost a fight.
A boxer with an injury did not lose a fight.
→ NO

Without any constraints on quantifier scope, the first sentence has five solutions and the second sentence has eighteen solutions, leading to ninety different possible inferences. As can be deduced from Figure 5, many of these boil down to spurious ambiguity, but even categorically, the correct prediction of a contradiction represents a minority across the potentially resulting inferences. Thus, the correct prediction crucially requires human intervention for the given parser.[8]

[8]This also hinges on pragmatic factors that are difficult to completely capture in a purely formal setting. This

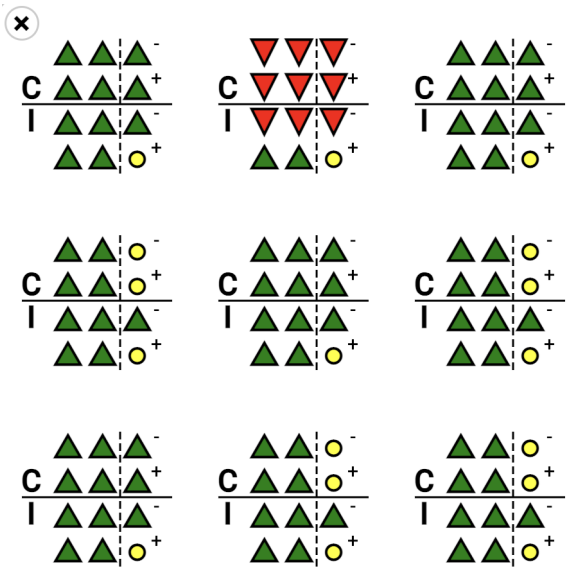


Figure 5: Inference output for ambiguous analyses based on example (3). The glyphs encode the results of a single potential inference and are based on *consistency* (C) and informativity (I) checking. Out of the nine shown possible results, only one (top center) correctly predicts the contradiction.

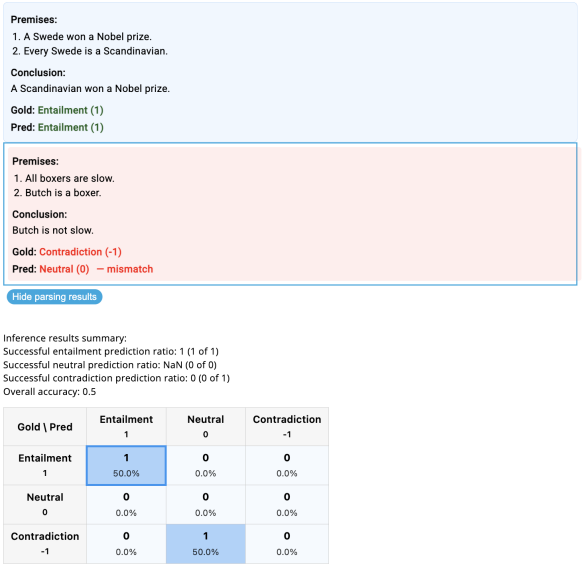


Figure 6: **Interactive confusion matrix:** Individual items are presented at the top. Mismatches in the NLI labeling are highlighted in red. In the confusion heatmap, classes can be selected to highlight individual items in blue accordingly.

3.4. Regression testing output

Parsing and inference results are summarized in overview reports. These reports contain information about the number of successful and failed parses and an accuracy overview regarding the NLI labeling (i.e., entailment, neutral, contradiction). This is visualized in an interactive confusion heatmap, which allows users to select classes, revealing the items associated with the corresponding predictions in blue (see Figure 6), facilitating error analysis.

In addition to inspecting and disambiguating items, we also extend this view with a visualization for DBA behavior across a testsuite. For each sentence, we calculate the application path (regardless of whether rules are dependent on each other or not). There, each rule corresponds to a node, and two nodes are connected if they are applied directly after one another. The global *applied rules graph* aggregates those paths into a single directed graph that highlights the complexity of the underlying annotation rules by highlighting disjunctions in the rule application. For example, in Figure 7, which is based on tense aspect rules, the complexity is fairly limited, mainly marking distinctions between past, present, and future tense at the top of the graph and aspectual distinctions between perfective and imperfective in the middle of the graph. The lower part of the graph is a little more complex as it further contextualizes some of those features (Zymla, 2024). Edges in the graph are associated with the sentences in the corresponding testsuite that pass through these paths. This is particularly useful when a testsuite is extended without prior individual testing. Furthermore, this view visualizes whether the rule set matches the syntactic input, as dangling nodes indicate rules that have not been applied, and which are, thus, not useful to the current testsuite.

3.5. Implementation

The present architecture was first described in Zymla et al. (2025b). As illustrated in Figure 8, the system spans two Java modules (LiGER and the GSWB) and a Python module that serves as an interface to the Vampire theorem prover. For the present paper, LiGER is extended to keep track of rule application paths during the annotation process and to calculate the corresponding histories and graphs. The GSWB is extended to calculate the different kinds of discriminants, described in the previous section. Furthermore, the Vampire interface is extended to permit batch processing. Finally, the frontend, which visualizes these components, is written in Angular and extended to present the newly added data structures.

All components are deployed as modular Docker containers. To use the system, a syntactic parser must be interfaced. We provide such an interface for the XLE, which is integrated natively as part of LiGER. Furthermore, we provide an external interface to UD by virtue of an additional Docker

is discussed in more detail in section 4.

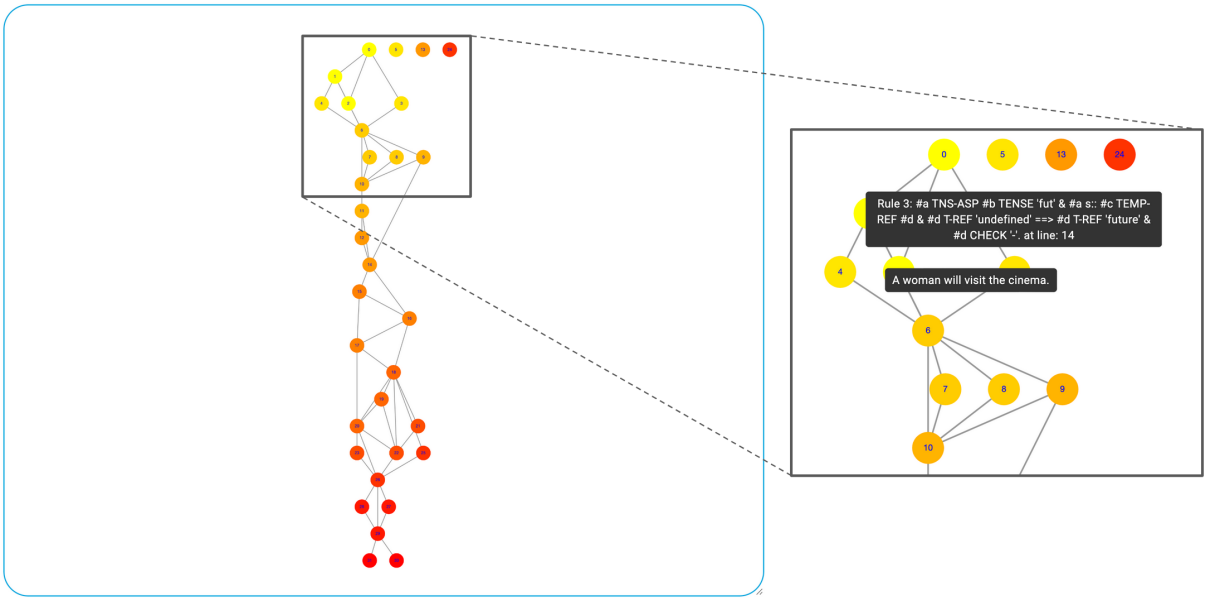


Figure 7: **Rule application view (testsuite):** This view captures the behavior of DBA rules across a testsuite. In the graph, each node corresponds to a rule, and each edge indicates which rules feed into each other. Hovering over nodes reveals the corresponding rules, and hovering over edges reveals which sentences follow the corresponding rule application path. The coloring of the nodes indicates the position of a rule in the sequence of all rules.

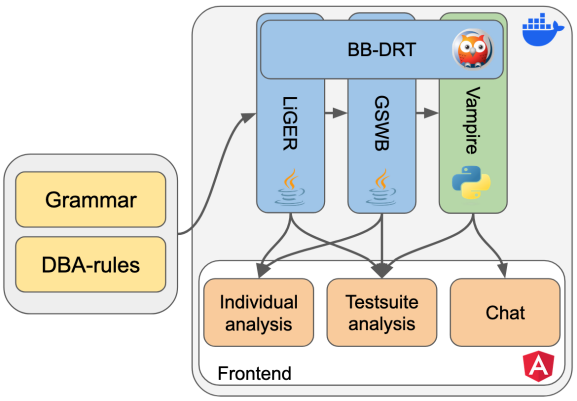


Figure 8: **System architecture:** LiGER is used to apply DBA-rules to a grammar. The resulting semantic annotations are passed on to the GSWB to resolve the compositional semantics. Both of these steps are visualized as part of the *individual analysis*, as well as the *testsuite analysis*, which optionally also takes reasoning with Vampire into account. The *chat* provides an explorative testing environment for the reasoning component.

container hosting an instance of the Stanza dependency parser (Qi et al., 2020).

4. Task-driven semantic parsing

There are good reasons to eschew formal semantic parsing in NLP applications, as maintenance is tedious and the systems are often still brittle. Furthermore, a one-fits-all broad coverage semantic parser is unlikely to be feasible, as we explain in more detail below. This may also have been a factor that led to the development of modular annotations such as those by the interoperable semantic annotation effort (ISA; Bunt 2015). In this section, we present some considerations regarding the question of how to approach formal semantic parsing from a research-driven computational linguistics perspective.

First, we propose to adopt ISA's "crystal growth" strategy to semantic annotation as the basis for developing formal semantic parsers. This is also more in line with formal semantic theory, which provides highly consistent analyses locally but often relies on approximate consistency in a broader context, such that interactions between unrelated phenomena are not always explicitly tested (but are testable). As an example, consider the classic *modus ponens* inference in (4).

(4)
A Swede won a Nobel Prize.
$\exists x, y[sw(x) \wedge prize(y) \wedge win(x, y)]$
Every Swede is a Scandinavian.
$\forall x[sw(x) \rightarrow \exists y[sc(y) \wedge x = y]]$

A Scandinavian won a Nobel Prize.
$\exists x, y[sc(x) \wedge prize(y) \wedge win(x, y)]$
→ YES

This inference falls out naturally from the interaction of the quantifiers and the identity predicate

be. However, adding, for example, semantic information about tense and aspect makes a formal analysis of the inference disproportionately more complex, as it would require an appropriate analysis of the persistence of states compared to events and how this interacts with tense and aspect. This would further require the design of additional axioms about temporal logic, also changing the inference process. A similar observation is made in Zymla et al. (2025b), who introduce a degree semantics inspired by Haruta et al. (2022) to deal with comparatives. This requires concessions during automated inference, as different proof search strategies are required compared to an example like (4) (without tense and aspect).

4.1. Building a foundation through interoperable semantic parsers

An interoperable approach advocates for first developing rules and testsuites, covering diverse phenomena (such as the ones mentioned above) in isolation but also in detail; however, with a shared set of assumptions about how to design annotations, allowing annotations of different phenomena to interact. For us, the common core is provided by Glue semantics and description-by-analysis, building on more general aspects of formal semantics.[9]

An interoperable semantic parser then consists of a (syntactic) grammar, a set of DBA rules providing semantic annotations, and a domain of testing. Given this approach, we can treat each pair of rules and test cases as building blocks for more comprehensive rule sets. The current system allows users to easily put building blocks together to investigate whether they provide a stable foundation for exploring more complex cases. Consider, for example, (5) which combines temporal and comparative semantics. If the individual building blocks are engineered properly, the analysis is expected to fall out compositionally when they are combined; however, if not, we can use the tools presented here to revise the individual building blocks and their interactions by investigating which rules break or stand in conflict with one another. We can also explore the emergence of unwanted ambiguities or other interactions that may obfuscate the reasoning process. Regression testing ensures that the arising more complex rule sets remain stable when introducing new phenomena and allows us to keep track of where the foundation may be weak and in need of revision.

[9]Furthermore, we assume a shared core grammar for syntactic analysis (e.g., an XLE grammar or a UD parser), but this is a design decision rather than a requirement of our approach.

(5)
Yesterday, Sam was faster than Jordan.
Today, Jordan is faster than Sam.

Sometimes, Sam is faster than Jordan.
→ YES

4.2. Exploring inference

While this remains to be tested, we hypothesize that stacking more and more rule sets on top of each other leads to less and less precise inference. This matches what we observe during development as grammars become more complex: i) more ambiguities are introduced, and ii) the proof search is more likely to time out (this is already observable for "simple" extensions like the one presented in Zymla et al. 2025b). Thus, an interoperable approach may give us further insights into how to best constrain semantic parsing to deal with particular inferences.

4.3. The trade-off between complexity and informativity

Formal semantic representations present a trade-off between complexity and informativity. More precisely, they focus on encoding certain entailments predicted by a given analysis, while leaving others unspecified. Thus, formal representations are, in a sense, fragmentary, only providing a partial picture; however, they still provide a precise view of certain reasoning patterns. They are distillates of aspects of natural language semantics.

On the one hand, this view seems sensible enough; however, on the other hand, in this view, formal semantic representations are prescriptive, capturing exactly those meanings we allow them to. This is why a task-driven approach to semantic parsing is so essential: only if we compare the output of our parsers to empirically grounded inference judgments can we ensure that semantic representations serve as falsifiable instantiations of linguistic theory. (We are, of course, not alone in this, see, e.g.: Haruta et al. 2022; Pulman 2018; Funakura et al. 2025).

Another point where we depart from currently popular methodologies in semantic parsing is that not all analyses should be fully disambiguated (cf. Abzianidze and Bos 2019). This is due to the fact that ambiguity may inform the annotation of NLI labels, particularly the neutral label, which can be most appropriate if a given sentence has two interpretations that lead to conflicting NLI labels during inference. This is another point where ambiguity management serves as a way of calibrating the boundary between logically and pragmatically driven inference.

If the output of a semantic parser is to be compared to empirically tested data, inferences should be tested under similar conditions, which also

paves the way for testing novel hypotheses such as the interaction between compositional ambiguity and inference. There, discriminant-based disambiguation enables us to separate spurious ambiguities from meaningful ambiguities, and it allows us to constrain meaningful ambiguities further. Essentially, we are not arguing for fully automated semantic parsing but rather for linguistically motivated and testable semantic parsing that bridges computational and formal methods. Given the overall approach to semantic parsing based on empirical adequacy that we advocate for, we, thus, also provide the foundation for exploring more fine-grained linguistic predictions computationally.

5. Limitations

We already mentioned some of the well-known limitations of formal approaches to semantic parsing. Furthermore, we have explained that NLI based on theorem proving can become computationally expensive as ambiguities grow and inferences become more reliant on more complex logics (e.g., temporal logic, degree logic). This can also result in long processing times when using the system.

Working with formal semantic parsers and, particularly, the process of disambiguation requires a certain level of competence in formal semantics. This alone can be seen as problematic, as there is a large strain of NLI work that argues that humans do not always reason logically, and that NLI labels should originate not from experts but from an empirically representative sample of annotators (see Manning 2006 for discussion). However, as addressed in the paper, we share the thought that NLI labels should be empirically motivated, and it stands to reason that at least some reasoning processes are best approximated by a logical system. However, this requires extensive testing of the kind of semantic parsers for which the tools here are envisioned.

The tools have been tested using XLE+Glue (Zymla et al., 2025a) during development. The corresponding GitHub page (see fn. 3) provides corresponding test sets based on Zymla et al. (2025b); however, they are limited in scope. Thus, while the tools have been tested for overall soundness, more detailed empirical testing remains for future work.

6. Conclusion

In this paper, we have presented a system that supports the development of task-driven formal semantic parsing at a larger scale by allowing for individual and regression testing. The focus of this system lies on LFG's description-by-analysis Glue approach, which provides a flexible and portable way to design meaning representations.

Furthermore, we discuss the role of such meaning representations and come to the conclusion that they are best observed in the context of automated reasoning, or NLU tasks more generally, because then they establish a firm connection between formal and computational linguistics and allow for the exploration of testable predictions.

We highlight a number of potential future directions with respect to this connection. Importantly, we suggest that an interoperable approach to semantic parsing is most likely to be promising. However, this comes with the caveat that a computational system must also be able to decide how to analyze a given problem. Thus, the present tools work best as a human-in-the-loop system, such that human experts produce meaning representations interactively.

Acknowledgments

7. Bibliographical References

Lasha Abzianidze and Johan Bos. 2019. Thirty musts for meaning banking. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 15–27. Association for Computational Linguistics.

Lasha Abzianidze, Rik Van Noord, Chunliu Wang, and Johan Bos. 2020. The parallel meaning bank: A framework for semantically annotating multiple languages. *Applied Mathematics and Informatics*, 25(2):45–60.

Avery D Andrews. 2008. The role of PRED in LFG + Glue. In *Proceedings of the LFG'08 Conference*, pages 47–67, Stanford, CA. CSLI Publications.

Ash Asudeh. 2006. Direct compositionality and the architecture of lfg. *Intelligent linguistic architectures: Variations on themes by Ronald M. Kaplan*, pages 363–387.

Ash Asudeh. 2023. Glue semantics. In *Handbook of Lexical Functional Grammar*, pages 651–697. Language Science Press, Berlin.

Daniel G. Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007. PARC's Bridge and Question Answering System. In *Proceedings of the GEAF 2007 Workshop*, pages 1–22.

Julia Bonn, Matthew J. Buchholz, Jayeol Chun, Andrew Cowell, William Croft, Lukas Denk, Sijia Ge, Jan Hajič, Kenneth Lai, James H. Martin, Skatje Myers, Alexis Palmer, Martha Palmer,

Claire Benet Post, James Pustejovsky, Kristine Stenzel, Haibo Sun, Zdeňka Urešová, Rosa Vallejos, Jens E. L. Van Gysel, Meagan Vigus, Nianwen Xue, and Jin Zhao. 2024. Building a broad infrastructure for uniform meaning representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2537–2547, Torino, Italia. ELRA and ICCL.

Johan Bos. 2009. Applying automated deduction to natural language understanding. *Journal of Applied Logic*, 7(1):100–112.

Joan Bresnan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. 2015. *Lexical-functional syntax*, volume 16. John Wiley & Sons.

Harry Bunt. 2008. Semantic underspecification: which technique for what purpose? In *Computing Meaning*, pages 55–85. Springer.

Harry Bunt. 2015. On the principles of semantic annotation. In *Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11)*.

Miriam Butt, Tina Bögel, Mark-Matthias Zymla, and Benazir Mumtaz. 2024. Alternative questions in urdu: from the speech signal to semantics. In *Proceedings of the LFG'24 Conference*, Konstanz. PubliKon.

Miriam Butt, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. 1999. *A grammar writer's cookbook*. CSLI Publications.

Nikos Chatzichrisafis, Dick Crouch, Tracy Holloway King, Rowan Nairn, Manny Rayner, and Marianne Santaholma. 2007. Regression testing for grammar-based systems. In *Proceedings of the GEAF07 Workshop*, pages 128–143, Stanford, CA. CSLI Publications.

Dick Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman. 2017. *XLE documentation*. Palo Alto Research Center.

Richard Crouch. 2005. Packed rewriting for mapping semantics to KR. In *Proceedings of the Sixth International Workshop on Computational Semantics (IWCS-6)*, pages 103–114, Tilburg.

Richard Crouch and Tracy Holloway King. 2006. Semantics via f-structure rewriting. In *Proceedings of the LFG'06 Conference*, pages 145–165, Stanford, CA. CSLI Publications.

Mary Dalrymple. 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. The MIT Press, Cambridge, MA.

Mary Dalrymple. 2001. *Lexical Functional Grammar*. Number 34 in Syntax and Semantics. Academic Press, San Diego, CA.

Mary Dalrymple, editor. 2023. *Handbook of Lexical Functional Grammar*. Number 13 in Empirically Oriented Theoretical Morphology and Syntax. Language Science Press, Berlin.

Mary Dalrymple, John Lamping, Fernando Pereira, and Vijay Saraswat. 1999. Quantification, anaphora, and intensionality. In Mary Dalrymple, editor, *Semantics and syntax in Lexical Functional Grammar: the resource logic approach*, pages 39–89.

Mary Dalrymple, Agnieszka Patejuk, and Mark-Matthias Zymla. 2020. XLE+Glue – a new tool for integrating semantic analysis in XLE. In *Proceedings of the LFG'20 Conference*, pages 89–108, Stanford, CA. CSLI Publications.

Jamie Y Findlay, Saeedeh Salimifar, Ahmet Yıldırım, and Dag TT Haug. 2023. Rule-based semantic interpretation for Universal Dependencies. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 47–57.

Dan Flickinger, Yi Zhang, and Valia Kordoni. 2012. Deepbank. a dynamically annotated treebank of the wall street journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

Hayate Funakura, Hyunsoo Kim, and Koji Mineshima. 2025. A theorem-proving-based evaluation of neural semantic parsing. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 295–306.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2022. Implementing natural language inference for comparatives. *Journal of Language Modelling*, 10(1).

Dag Haug, Jamie Yates Findlay, and Ahmet Yıldırım. 2023. The long and the short of it: DRASTIC, a semantically annotated dataset containing sentences of more natural length. In *Proceedings of the Fourth International Workshop on Designing Meaning Representations*, pages 89–98.

Luheng He, Julian Michael, Mike Lewis, and Luke Zettlemoyer. 2016. Human-in-the-loop parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2337–2342, Austin, Texas. Association for Computational Linguistics.

Nancy Ide and Harry Bunt. 2010. Anatomy of Annotation Schemes: Mapping to GrAF. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 247–255.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, and Nathan Schneider. 2020. Abstract meaning representation (amr) annotation release 3.0. LDC2020T02, Web Download. DOI: 10.35111/44cy-bp51.

Laura Kovács and Andrei Voronkov. 2013. First-order theorem proving and vampire. In *International Conference on Computer Aided Verification*, pages 1–35. Springer.

Iddo Lev. 2007. *Packed computation of exact meaning representations*. Ph.D. thesis, Stanford University.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2021. Improving BERT with syntax-aware local attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 645–653, Online. Association for Computational Linguistics.

Matthias Lindemann, Alexander Koller, and Ivan Titov. 2024. Strengthening structural inductive biases by pre-training to perform syntactic transformations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11558–11573, Miami, Florida, USA. Association for Computational Linguistics.

Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards Better Analysis of Machine Learning Models: A Visual Analytics Perspective. *Visual Informatics*, 1(1):48–56. Publisher: Elsevier.

Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In *Proceedings of the eight International Conference on Computational Semantics*, pages 140–156.

Christopher D. Manning. 2006. Local Textual Inference: It's Hard to Circumscribe, but You Know It When You See It — and NLP Needs It. Manuscript, Stanford University.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Moritz Meßmer and Mark-Matthias Zymla. 2018. The Glue semantics workbench: a modular toolkit for exploring linear logic and Glue semantics. In *Proceedings of the LFG'18 Conference*, pages 249–263, Stanford, CA. CSLI Publications.

Adam Przepiórkowski and Agnieszka Patejuk. 2023. Filling gaps with Glue. In *Proceedings of the LFG'23 Conference*, pages 223–240, Konstanz. PubliKon.

Stephen Guy Pulman. 2018. Second order inference in natural language semantics. *Journal of Language Modelling*, 6(1):1–40.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations*, pages 101–108.

Victoria Rosén, Koenraad De Smedt, and Paul Meurer. 2006. Towards a toolkit linking treebanking to grammar development. In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories*, pages 55–66.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh M Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinoğlu, I Wayan Arka, and Meladel Mistica. 2013. ParGramBank: The ParGram Parallel Treebank. In *ACL*, pages 550–560.

Geoff Sutcliffe, Stephan Schulz, Koen Claessen, and Allen Van Gelder. 2006. Using the TPTP language for writing derivations and finite interpretations. In *Automated Reasoning – IJCAR 2006*, volume 4130 of *Lecture Notes in Computer Science*, pages 67–81, Seattle, WA, USA. Springer.

Shuai Zhang, Lijie Wang, Xinyan Xiao, and Hua Wu. 2022. Syntax-guided contrastive learning for pre-trained language model. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2430–2440, Dublin, Ireland. Association for Computational Linguistics.

Xiao Zhang, Gosse Bouma, and Johan Bos. 2025. Neural semantic parsing with extremely rich symbolic meaning representations. *Computational Linguistics*, 51(1):235–274.

Mark-Matthias Zymla. 2024. *Tense and aspect in multilingual semantic construction*. Ph.D. thesis, University of Konstanz.

Mark-Matthias Zymla, Mary Dalrymple, and Agnieszka Patejuk. 2025a. Computational semantic tools for Glue semantics. In *Proceedings of the 16th International Conference on Computational Semantics (IWCS 2025)*, pages 189–207, Düsseldorf. Association for Computational Linguistics.

Mark-Matthias Zymla, Kascha Kruschwitz, and Paul Zodl. 2025b. An instructive implementation of semantic parsing and reasoning using lexical functional grammar. In *Proceedings of the 2nd Bridging the Gap between Human and Automated Reasoning Workshop (BriGap-2)*.

A. Appendix: UI figures

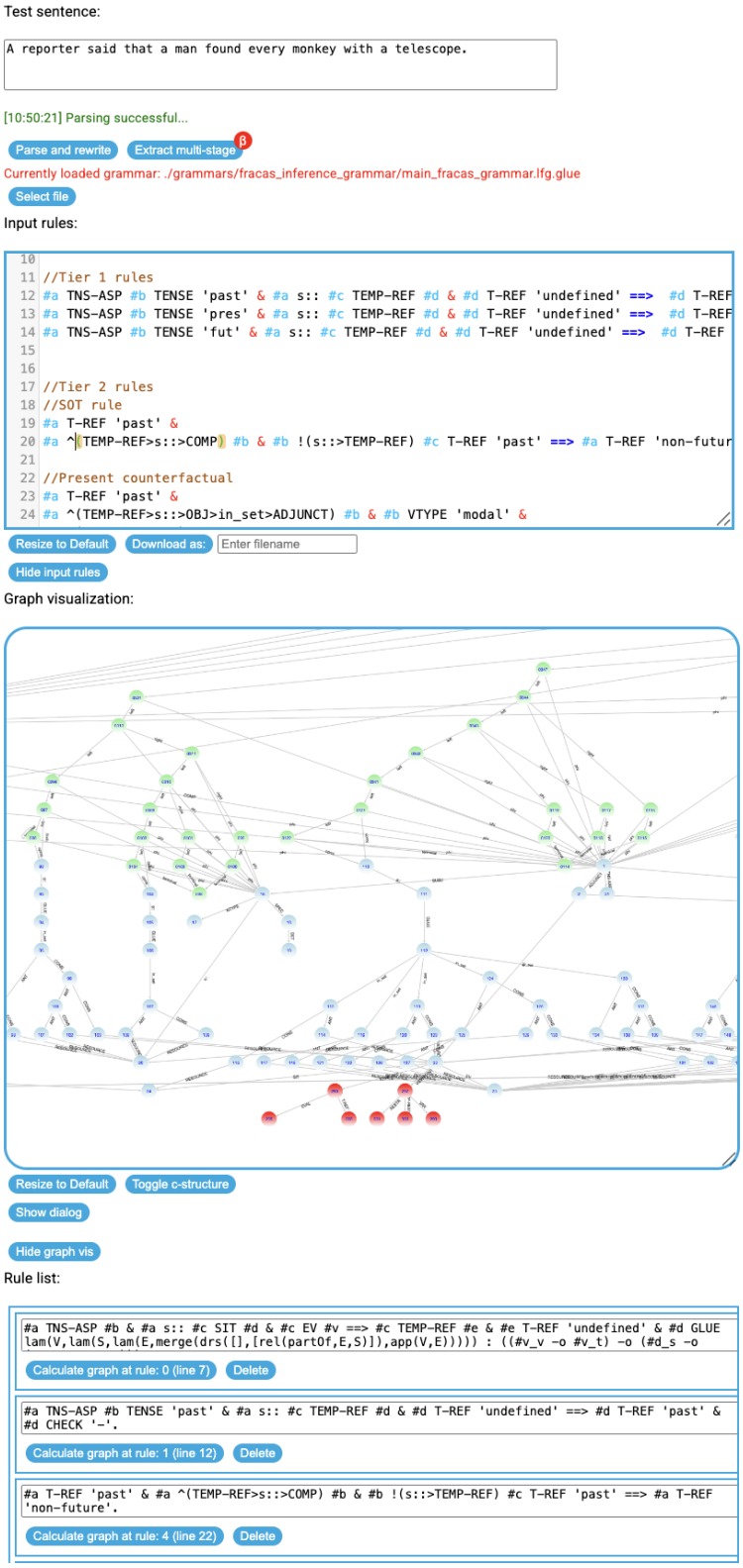


Figure 9: **Description-by-analysis view:** This view allows viewing of the specified DBA rules and the resulting annotated graph (here, the graph contains c-structure, f-structure, and semantic structure).Furthermore, it provides access to the history of applied rules.

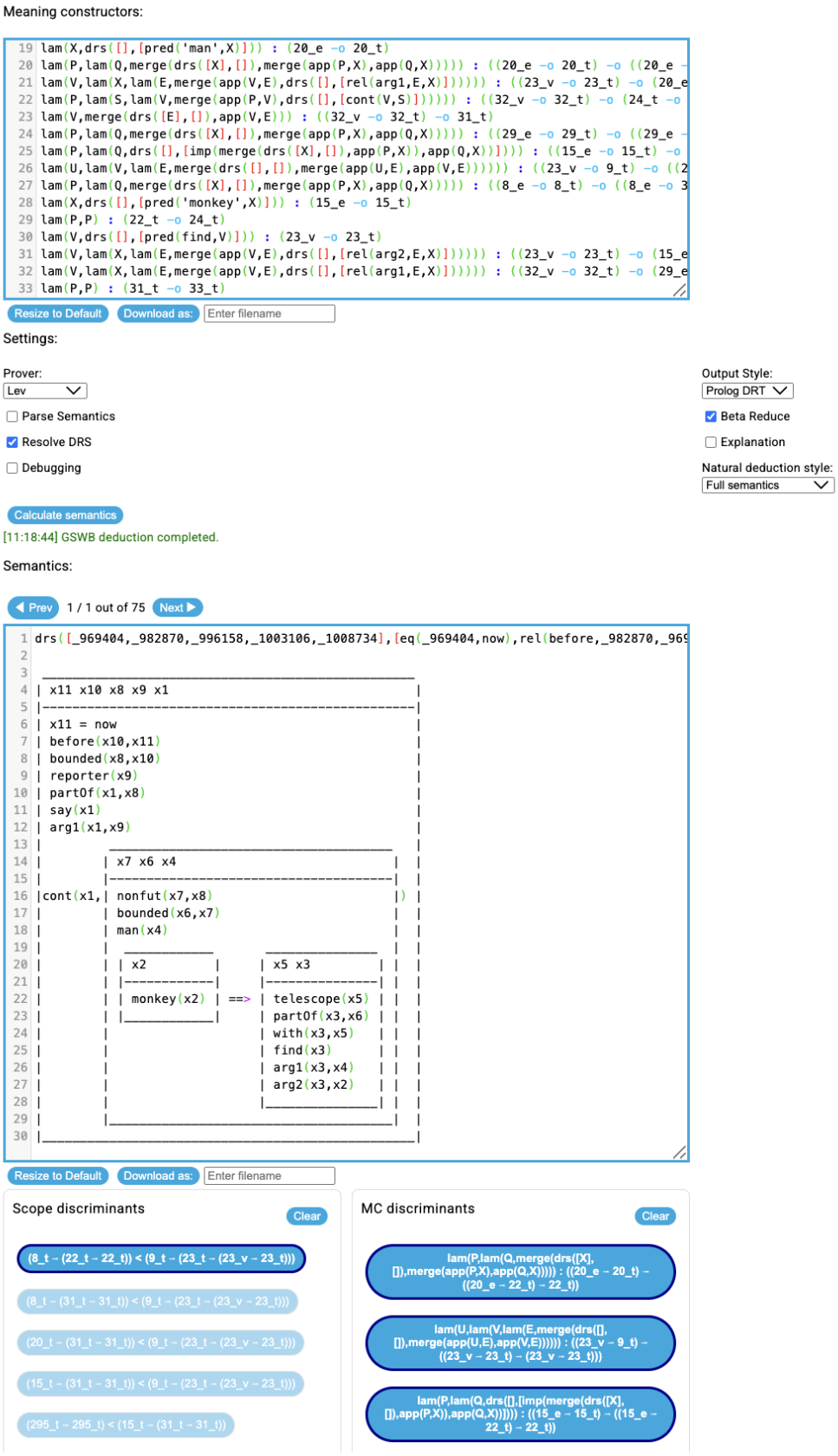


Figure 10: **Semantic composition view:** This view presents the meaning constructor sets available for a given sentence at the top and the resulting derivations at the bottom, which can be filtered using scope and MC discriminants. This view can be further extended to include information about the derivation.

Figure 11: **Full view:** This view combines the DBA view and the semantic composition view. The two major components can be scrolled individually. Most components can be hidden, including the two major ones.

Author Index