OntoLex 2006 Programme

8.40-9.00: **Welcome**

9.00-9.40: **Ontology-based Reasoning over lexical resources by means of ontologies** (Scheffczyk, Baker, Narayanan)
9.40-10.20: **Towards sounder taxonomies in wordnets** (Pedersen)
10.20-11.00: **Using EuroWordNet for the translation of ontologies** (Declerck et al.)

11.00-11.30: *Coffee Break*

11.30-12.10: **Linguistic Enrichment of Ontologies: a methodological framework** (Pazienza, Stellato)
12.10-12.50: **LingInfo: a Model for the Integration of Linguistic Information in Ontologies** (Buitelaar et al.)
12.50-13.30: **Tying the Knot: Ground entities, Descriptions and Information Objects for Construction-based Information Extraction** (Micelli, Aras, Porzel)

13.30-14.30: *Lunch*

14.30-15.30: **PANEL –Semantic Operability for lexical resources**

15.30-15.50: **Using various semantic relations in Word Sense Disambiguation** (Szarvas et al.)
15.50-16.10: **SmartIndexer – Amalgamating Ontologies and Lexical resources for Document Indexing** (Peter, Sack, Beckstein)
16.10-16.30: **How Linguistic Resources May help to recommend TV programmes** (Ludwig, Mandl, Schmidt)

16.30-17.00: *Coffee Break*

17.00-17.20: **Exploiting Linguistic Resources for building linguistically motivated ontologies in the Semantic Web** (Pazienza, Stellato)
17.20-17.40: **Metadata Cards for Describing Project Gutenberg Texts** (Reck)
17.40-18.00: **Open-class named entity classification in multiple domains** (Faulhaber et al.)

18.00-18.10: **Conclusions**

OntoLex 2006 Organisers

Alessandro Oltramari, LOA-CNR & University of Trento
Via Solteri 38, 38100 Trento (ITALY)

Chu-Ren Huang, Institute of Linguistics, Academia Sinica
Nankang, Taipei, Taiwan 115 (TAIWAN)

Alessandro Lenci, University of Pisa, Dipartimento di Linguistica "T. Bolelli
Via Santa Maria 36, 56100 Pisa (ITALY)

Paul Buitelaar, DFKI, Language Technology Lab
Stuhlsatzenhausweg 3, D-66123 Saarbrücken (GERMANY)

Christiane Fellbaum, Department of Psychology, Green Hall, Princeton University
221 Nassau St., Princeton, NJ 08544 (USA)

OntoLex 2006 Programme Committee

Table of Contents

Author Index

Ontology-based Reasoning about Lexical Resources

Jan Scheffczyk*, **Collin F. Baker***, **Srini Narayanan***

*International Computer Science Institute
1947 Center St., Suite 600, Berkeley, CA, 94704
{jan,collinb,snarayan}@icsi.berkeley.edu

Abstract

Reasoning about natural language most prominently requires combining semantically rich lexical resources with world knowledge, provided by ontologies. Therefore, we are building bindings from FrameNet – a lexical resource for English – to various ontologies depending on the application at hand. In this paper we show the first step toward such bindings: We translate FrameNet to the Web Ontology Language OWL DL. That way, FrameNet and its annotations become available to Description Logic reasoners and other OWL tools. In addition, FrameNet annotations can provide a high-quality lexicalization of the linked ontologies.

1. Introduction

Combining large lexical resources with world knowledge, via ontologies, is a crucial step for reasoning over natural language, particularly for the Semantic Web. Concrete applications include semantic parsing, text summarization, translation, and question answering. For example, questions like "Could Y have murdered X?" may require several inference steps based on semantic facts that simple lexicons do not include. Moreover, they require so-called open-world semantics offered by state-of-the art Description Logic (DL) reasoners, e.g., FaCT (Horrocks, 1998) or Racer (Wessel and Möller, 2005). The FrameNet lexicon (Ruppenhofer et al., 2005) has a uniquely rich level of semantic detail; thus, we are building bindings from FrameNet to multiple ontologies that will vary depending on the application. That way, we enable reasoners to make inferences over natural-language text.

In this paper, we report on the first step toward this goal: we have automatically translated a crucial portion of FrameNet to OWL DL and we show how state-of-the-art DL reasoners can make inferences over FrameNet-annotated sentences. Thus, annotated text becomes available to the Semantic Web and FrameNet itself can be linked to other ontologies. This work gives a clear motivation for the design of our proposed ontology bindings and defines the baseline for measuring their benefits.

This paper proceeds as follows: In Sect. 2. we briefly introduce FrameNet – a lexical resource for English. We present our design decisions for linking FrameNet to ontologies in Sect. 3. Sect. 4. includes the heart of this paper: A formalization of FrameNet and FrameNet-annotated sentences in OWL DL. In Sect. 5. we show how our OWL DL representation can be used by the DL reasoner RacerPro in order to implement tasks of a question answering system, based on reasoning. We evaluate our approach in Sect. 6. Sect. 7. concludes and sketches directions for future research.

2. The FrameNet Lexicon

FrameNet is a lexical resource for English, based on frame semantics (Fillmore, 1976; Fillmore et al., 2003; Narayanan et al., 2003). A semantic frame (hereafter simply frame) represents a set of concepts associated with an event or a state, ranging from simple (Arriving, Placing) to complex (Revenge, Criminal_process). For each frame, a set of roles, called frame elements (FEs), is defined, about 10 per frame. We say that a word can evoke a frame, and its syntactic dependents can fill the FE slots. Semantic relations between frames are captured in frame relations, each with corresponding FE-to-FE mappings. FrameNet currently contains more than 780 frames, covering roughly 10,000 lexical units (LUs) = word senses; these are all supported by more than 135,000 FrameNet-annotated example sentences, which are also used as training data for frame and FE recognizing systems (Litowski, 2004; Erk and Padó, 2005; Erk and Padó, 2006).[1]

Fig. 1 shows a portion of the Attack frame, which *inherits* from the more general frame Intentionally_affect (which in turn inherits from the frames Transitive_action and Intentionally_act). In addition, Attack *uses* the frame Hostile_encounter. The FEs of the Attack frame are mapped to their corresponding FEs in connected frames. For example, the FE Assailant is mapped to the FE Agent in the Intentionally_act frame.

3. Linking FrameNet to Ontologies for Reasoning

NLP applications using FrameNet require knowledge about the possible fillers for FEs. For example, a semantic frame parser needs to know whether a certain chunk of text (or a named entity) might be a proper filler for an FE – so it will check whether the filler type of the FE is compatible with the type of the named entity. Therefore, we want to provide constraints on fillers of FEs, so-called *semantic types* (STs). Currently, FrameNet itself has defined about 40 STs that are ordered by a subtype hierarchy. For example, the Assailant FE and the Victim FE in the Attack frame both have the ST Sentient, which in turn is a subtype of Animate_being and then Living_thing, Physical_object, and Physical_entity. It is obvious that FrameNet STs are somewhat similar to the concepts (often called classes) defined in ontologies like SUMO (Niles and Pease, 2001) or Cyc (Lenat, 1995). Compared to ontology classes, however, FrameNet STs are much more shallow, have fewer relations between them (we only have subtyping and no other relations), and are not

[1] For further information on FrameNet see http://framenet.icsi.berkeley.edu.

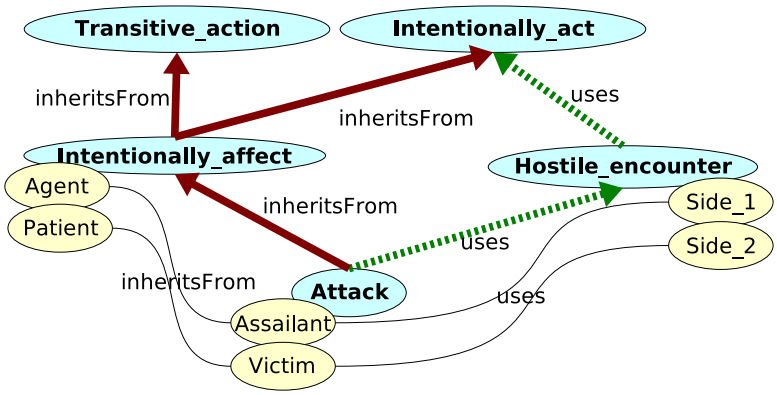


Figure 1: Abridged example frame Attack and some connected frames.

context specific. Naturally, in a *lexicographic* project like FrameNet STs play a minor role only.
Therefore, we want to employ the STs from existing large ontologies such as SUMO or Cyc; in this way we will gain a number of advantages almost for free:

- AI applications can use the knowledge provided by the target ontology.
- We can provide different STs suitable for particular applications by bindings to different ontologies.
- We can use ontologies in order to query and analyze FrameNet data. For example, we can measure the semantic distance between frames based on different target ontologies or we can check consistency and completeness of FrameNet w.r.t. some target ontology.
- The target ontologies would benefit from FrameNet, supplementing their ontological knowledge with a proper lexicon and annotated example sentences.

Compared to other lexicon-ontology bindings (Niles and Pease, 2003; Burns and Davis, 1999), our bindings offer a range of advantages due to specific FrameNet characteristics: FrameNet models semantic and syntactic valences plus the predicate-argument structure. FrameNet includes many high-quality annotations, providing training data for machine learning. In contrast to WordNet synset annotations, our annotations include role labelling. Frame semantics naturally provides cross-linguistic abstraction plus normalization of paraphrases and support for null instantiation (NI). Notice that a detour via WordNet would introduce additional noise through LU lookup (Burchardt et al., 2005). In addition, WordNet synset relations are not necessarily compatible with FrameNet relations.
The bindings from FrameNet to ontologies should be described in the native language of the target ontologies, i.e., KIF (for bindings to SUMO), CycL (for bindings to Cyc), or OWL (for bindings to OWL ontologies). This allows the use of standard tools like reasoners directly, without any intermediate steps. Also, arbitrary class expressions can be used and ad-hoc classes can be defined if no exact corresponding class could be found in the target ontology. We expect this to be very likely because FrameNet is a lexicographic project as opposed to ontologies, which are usually driven by a knowledge-based approach. Finally, the binding should be as specific as possible for the application at hand. For example, in a military context we would like to bind FEs to classes in an ontology about WMD or terrorism instead of using a binding to SUMO itself, which only provides upper level classes.[2]
The vital precondition for any such bindings is, however, to have FrameNet available in an appropriate ontology language (e.g., KIF, CycL, or OWL). A representation of FrameNet in an ontology language bears the additional advantages of formalizing certain properties of frames and FEs, and enabling us to use standard tools to view, query, and reason about FrameNet data. For querying, one could, e.g., use the ontology query language SPARQL. Next, we describe a formalization of a portion of FrameNet in OWL DL, which easily generalizes to more expressive ontology languages like KIF or CycL.

4. Formalizing FrameNet in OWL DL

Our major design decisions for representing FrameNet as an ontology are:

1. to represent frames, FEs, and STs formally as classes,
2. to model relations between frames and FEs via existential property restrictions on these classes, and
3. to represent frame and FE realizations in FrameNet-annotated texts as *instances* of the appropriate frame and FE classes, respectively.

Building on (Narayanan et al., 2003), we have chosen OWL DL as representation language mainly because better tools are available for it (particularly for reasoning) than for OWL Full or other similarly expressive languages. Our representation differs from many WordNet OWL representations, which represent synsets as *instances* and hence cannot use class expressions for ontology bindings.[3] Instead, WordNet bindings to SUMO employ a proprietary mechanism,[4] which cannot be used "out of the box" by ontology tools like reasoners.

[2]For examples of SUMO domain ontologies see www.ontologyportal.org.
[3]See, for example, http://www.w3.org/2001/sw/BestPractices/
[4]See www.ontologyportal.org.

In order to keep the size of our ontology manageable, we have chosen to split it into the *FrameNet Ontology* and *Annotation Ontologies*. The FrameNet Ontology includes FrameNet data like frames, FEs, and relations between them. Annotation Ontologies represent FrameNet-annotated sentences and include parts of the FrameNet Ontology that are necessary.

4.1. The FrameNet Ontology

Fig. 2 shows a simplified excerpt of the FrameNet Ontology. The subclasses of the Syntax class are used for annotations and are connected to frames and FEs via the *evokes* and *fillerOf* relations, respectively. Frames and FEs are connected via binary relations, e.g., the *usesF* property or the *hasFE* property, which connects a frame to its FEs. Consider our example frame Attack, which inherits from the frame Intentionally_affect and uses the frame Hostile_encounter. We model frame and FE inheritance via subclassing and other frame and FE relations via existential property restrictions (owl:someValuesFrom). Thus, the class Attack is a subclass of Intentionally_affect. In addition, we require that an instance of type Attack has at least one instance of type Hostile_encounter connected via the *usesF* property. The FEs of Attack are connected via an existential restriction on the *hasFE* property. FE relations are modeled similarly to frame relations.

Recall that class restrictions are inherited. Therefore, the class Attack inherits the restrictions $\exists$.*hasFE* Patient and $\exists$.*hasFE* Agent from the class Intentionally_affect. These restrictions are, however, subsumed by the restrictions on the Attack class itself because Victim is a subclass of Patient and Assailant is a subclass of Agent. Of course, OWL inheritance requires proper inheritance on the FrameNet data. We have implemented rigorous formal quality management for FrameNet that takes care of proper inheritance (Scheffczyk and Ellsworth, 2006).

Notice that our formal representation is incomplete. We would love to say that, e.g., an instance of Intentionally_affect has exactly one instance of type Patient connected via the *hasFE* property:

Intentionally_affect *hasFE* 1 Patient

This requires, however, so-called qualified cardinality restrictions (QCR) – a non-standard extension to OWL.[5] The workaround to require Intentionally_affect *hasFE* 2 or Intentionally_affect *hasFE* $\leq$ 2 does not work in our case due to inheritance: The Attack frame may have more than just 2 FEs. Indeed, we could define subproperties for the *hasFE* property (and *all* other properties), which would, however, clutter up our data significantly.[6] Therefore, we live with this incomplete modeling unless QCRs are accepted as a standard.

Our ST hierarchy is modeled as a simple subclass hierarchy. STs are attached to FEs via subclass relationships. So, the classes Victim and Assailant are both subclasses of the class Sentient. We intend to use this mechanism for linking FrameNet to other ontologies also. So we can use arbitrary OWL DL class expressions for our bindings and at the same time achieve a homogeneous formal representation that OWL tools can make use of.

One could use the FrameNet Ontology for querying and reasoning over FrameNet itself. For reasoning over natural language text, however, we must find a way to incorporate this text into the FrameNet Ontology. We do this by means of *Annotation Ontologies*, which we generate from FrameNet-annotated text.

4.2. Annotation Ontologies

FrameNet-annotated text provides textual realizations of frames and FEs, i.e., the frames and FEs cover the semantics of the annotated sentences. In ontological terms, FrameNet-annotated text constitutes instances of the appropriate frame and FE classes, respectively. From an annotated sentence we generate an Annotation Ontology, which includes parts of the FrameNet Ontology and fulfills all its class restrictions. In other words, the FrameNet Ontology provides a formal specification for Annotation Ontologies.

Consider an example sentence, which we derived from an evaluation exercise within the AQUINAS project called "KB Eval;" where sentences for analysis were contributed by various members of the consortium.

S *48 Kuwaiti jet fighters managed to escape the Iraqi invasion.*[7]

This sentence has three annotation sets:

1. The target word *invasion* evokes the Attack frame, where *Iraqi* fills the Assailant FE. The Victim FE has no filler, i.e., it is null instantiated (NI).
2. The target word *escape* evokes the Avoiding frame, with FE fillers *48 Kuwaiti jet fighters* $\rightarrow$ Agent, *the Iraqi invasion* $\rightarrow$ Undesirable_situation.
3. The target word *managed* evokes the Successful_action frame, with FE fillers *48 Kuwaiti jet fighters* $\rightarrow$ Protagonist, *to escape the Iraqi invasion* $\rightarrow$ Goal.

From this annotated sentence we first create a syntactic dependency graph and generate the appropriate frame and FE instances as shown in Fig. 3 A Span represents a chunk of text that can evoke a frame or provide a filler for an FE. We derive Spans, syntactic subsumption, and the relations to frames and FEs based on the annotations. For example, *invasion* evokes the Attack frame. Thus we (1) generate a Span that represents the text *invasion* and place it properly into the Span dependency graph, (2) generate the frame instance Attack_S (with type Attack), and (3) connect the Span to Attack_S via the *evokes* property. We proceed similarly with the FE filler *Iraqi* $\rightarrow$ Agent. Here we generate the FE instance Agent_S, connect it to its frame instance Attack_S via the *hasFE* property, and connect the Span representing *Iraqi* to Agent_S via the *fillerOf* property. Finally, we identify FEs that are evoked by the same Span via owl:sameAs.

[5] see http://www.w3.org/2001/sw/BestPractices/OEP/QCR/

[6] Even now, the FrameNet Ontology reaches a critical size of 100,000 triples.

[7] In the sequel we index the instances emerging from a sentence by its identifier, here S.

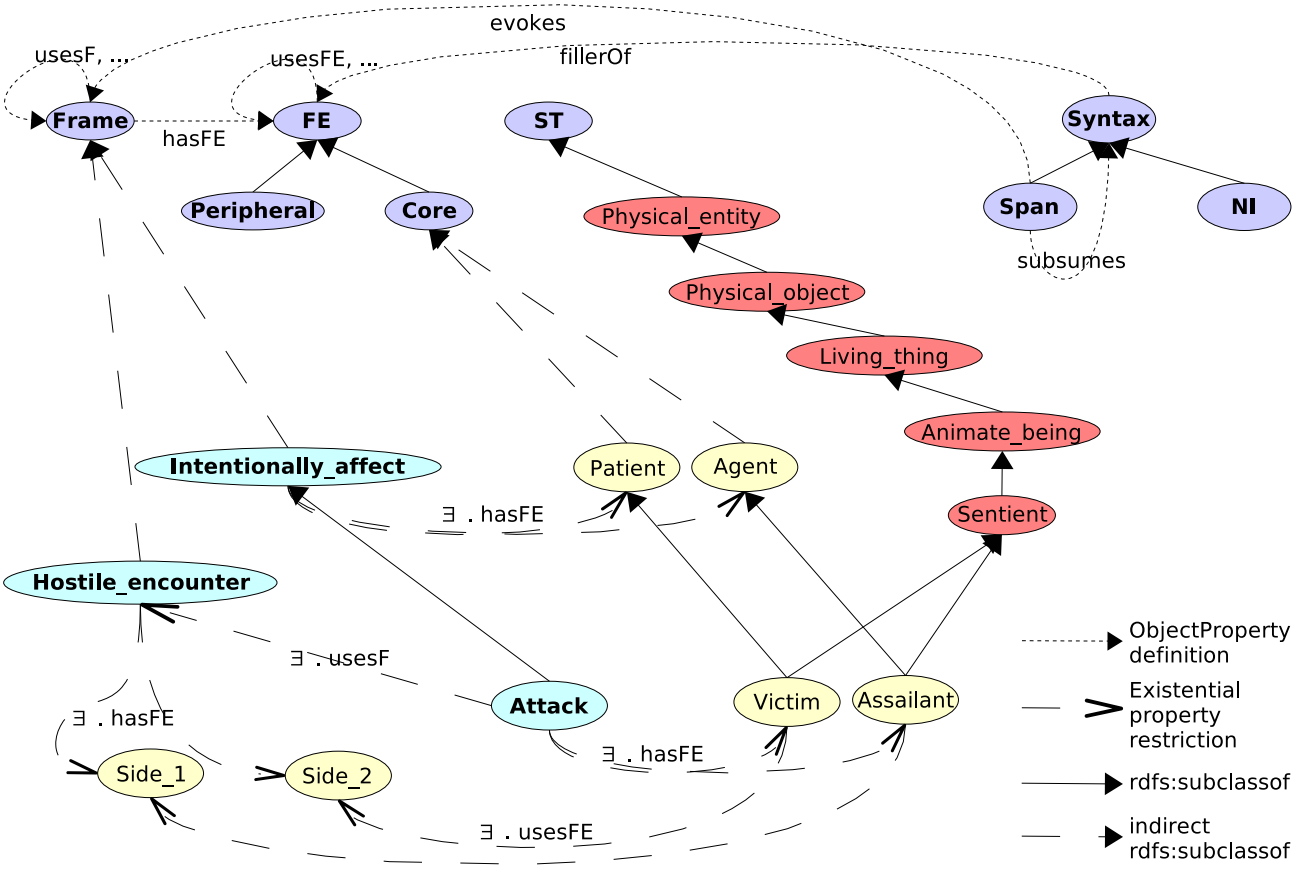


Figure 2: Part of the FrameNet Ontology for the Attack frame and some connected frames.

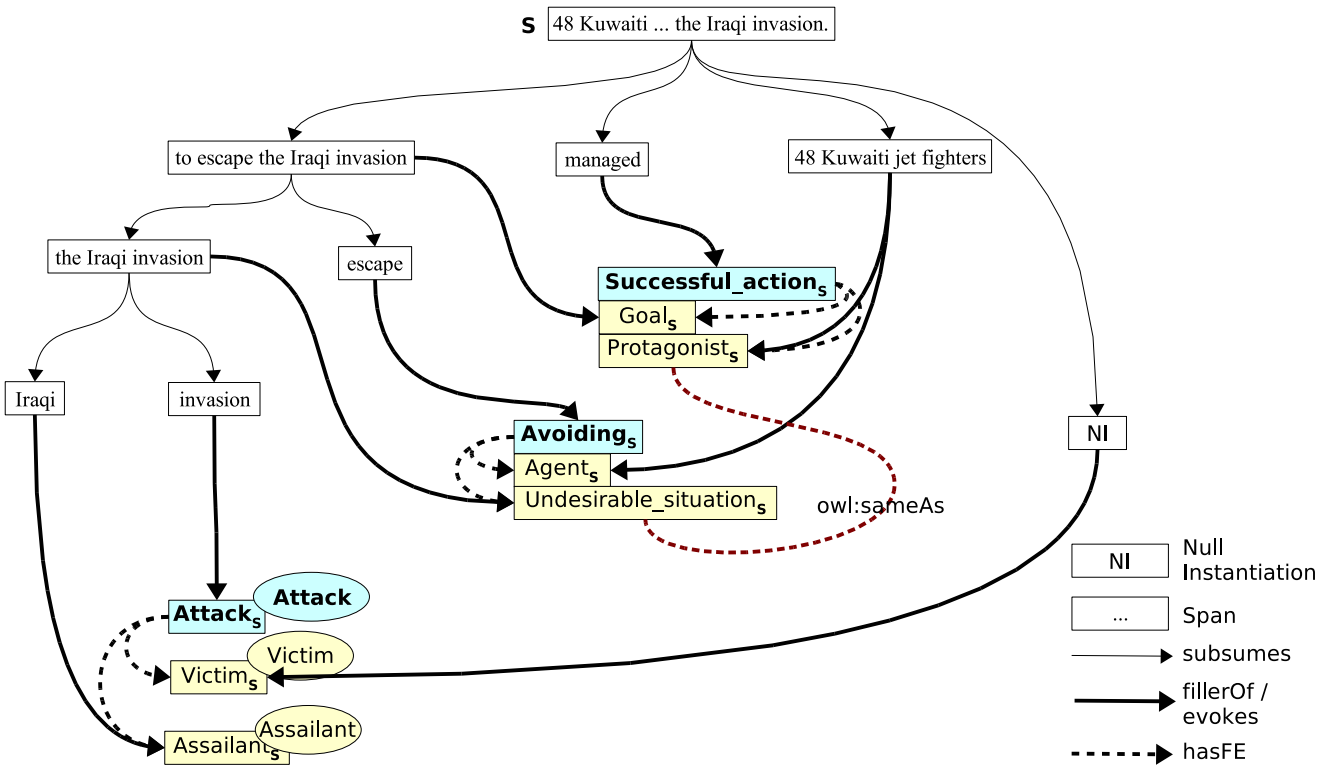


Figure 3: Annotation Ontology for: *48 Kuwaiti jet fighters managed to escape the Iraqi invasion.* (Step 1)

We can do this purely based on syntactic evidence. For example, the FE instances Protagonists_S and Agent_S are identified because the are both filled by the Span representing the text *48 Kuwaiti jet fighters*. This significantly aids reasoning about FrameNet-annotated text.[8]

The second step in generating an Annotation Ontology is to satisfy the class restrictions of the FrameNet ontology, i.e., to generate appropriate instances and to connect them properly. Thus, for a frame instance i of type C_i we

1. travel along each existential class restriction on a property pr to a class C_j ($\exists.pr\ C_j$),
2. generate an instance j of type C_j,
3. connect the instances i and j via the property pr, and
4. proceed with instance j.

Fig. 4 illustrates this algorithm for our example frame instance Attack. We generate the frame instance $\text{Hostile\_encounter}_S$ and its FE instances Side_1_S and Side_2_S, and connect Attack_S to $\text{Hostile\_encounter}_S$ via *usesF*. Similarly, we connect Assailants_S to Side_1_S and Victims_S to Side_2_S via *usesFE*. In addition, we identify the connected FE instances via owl:sameAs, which expresses the semantics of FE mappings: The Victim in an Attack *is* the Side_2 in a Hostile_encounter, i.e., their fillers are the same.

In addition to the class restrictions, we also travel along the inheritance hierarchy, which could be useful, e.g., for paraphrasing. Therefore, we generate the instance $\text{Intentionally\_affect}_S$ and its FEs. Clearly, we want to express that the act of attacking someone is also an act of intentionally affecting him (in a more general sense). We connect the instances via the somewhat artificial properties *inheritsF* and *inheritsFE* because there is no other way to relate particular instances in OWL.

Fig. 4 shows only a small *fraction* of the generated ontology. Because the size of the ontologies is crucial for DL reasoners we limit the number of generations. Also, Annotation Ontologies do not import the whole FrameNet Ontology but include only those classes that we generate instances for, i.e., classes connected to the evoked frame classes.

Next, we show a simple example, illustrating the use of Annotation Ontologies for reasoning.

5. Reasoning over FrameNet-annotated Text

We have investigated the potential of DL reasoners in question answering, which is a challenging application area for ontology text representation. For our current experiments we use RacerPro[9] (Wessel and Möller, 2005). Given a FrameNet-annotated question, we let RacerPro perform various reasoning tasks in order to identify compatible frames and FEs in potential answer sentences. If RacerPro succeeds, then the Spans bound to these FEs contain the answer, otherwise the question cannot be answered from the text.

Consider three example questions.

Q1 *How many Kuwaiti jet fighters escaped the Iraqi invasion?*

Q2 *How many Kuwaiti jet fighters escaped?*

Q3 *Did Iraq clash with Kuwait?*

Q4 *Was there a conflict between Iraq and Kuwait?*

Partial Annotation Ontologies for these questions are illustrated in Fig. 5.

Given the Annotation Ontology of the question, we let RacerPro perform the following queries, which can be formalized in nRQL.[10] In the following we will use question Q1 as an example of how the algorithm works.

1. For the question get the evoked frames instances, their FEs, and Spans.

   $\text{Avoiding}_{Q1} \rightarrow \text{Undesirable\_s.}_{Q1} \rightarrow$ the Iraqi invasion
   $\rightarrow \text{Agent}_{Q1} \rightarrow$ How many ...
   $\text{Attack}_{Q1} \rightarrow \text{Assailant}_{Q1} \rightarrow$ Iraqi
   $\rightarrow \text{Victim}_{Q1} \rightarrow$ NI

2. For each frame and FE instance determine the direct classes.

   $\text{Avoiding}_{Q1} \rightarrow$ Avoiding
   $\text{Undesirable\_s.}_{Q1} \rightarrow$ Undesirable_s.
   $\text{Agent}_{Q1} \rightarrow$ Agent
   $\text{Attack}_{Q1} \rightarrow$ Attack
   $\text{Assailant}_{Q1} \rightarrow$ Assailant
   $\text{Victim}_{Q1} \rightarrow$ Victim

   Notice that we get only one class because in this question a Span is the filler of at most one FE.

3. For each of the frame classes obtain frame instances *f* that are different from the ones in the question. Similarly, we look for corresponding FE instances *fe* that are connected to a frame instance *f* via the *hasFE* property.

   Avoiding $\rightarrow \text{Avoiding}_S$
   Undesirable_s. $\rightarrow$ Undesirable_s._S
   Agent $\rightarrow$ Agent_S
   Attack $\rightarrow \text{Attack}_S$
   Assailant $\rightarrow$ Assailant_S
   Victim $\rightarrow$ Victim_S

4. Get the Spans of the FE instances above and determine whether they are compatible with the Spans of the corresponding FEs in the question (we mark success by $\surd$ and failure by $\times$).

   $\text{Undesirable\_s.}_S \rightarrow$ the Iraqi invasion $\surd$
   $\text{Agent}_S \rightarrow$ 48 Kuwaiti jet fighters $\times$
   $\text{Assailant}_S \rightarrow$ Iraqi $\surd$
   $\text{Victim}_S \rightarrow$ NI $\surd$

[8] Alternatively, we could formalize a SWRL rule $\textit{fillerOf}(s,a) \wedge \textit{fillerOf}(s,b) \rightarrow \textit{owl:sameAs}(a,b)$. We do not do so because not all reasoners provide a SWRL implementation.

[9] see www.racer-systems.com

[10] We have to use multiple queries because class and instance queries cannot be intermixed in RacerPro.

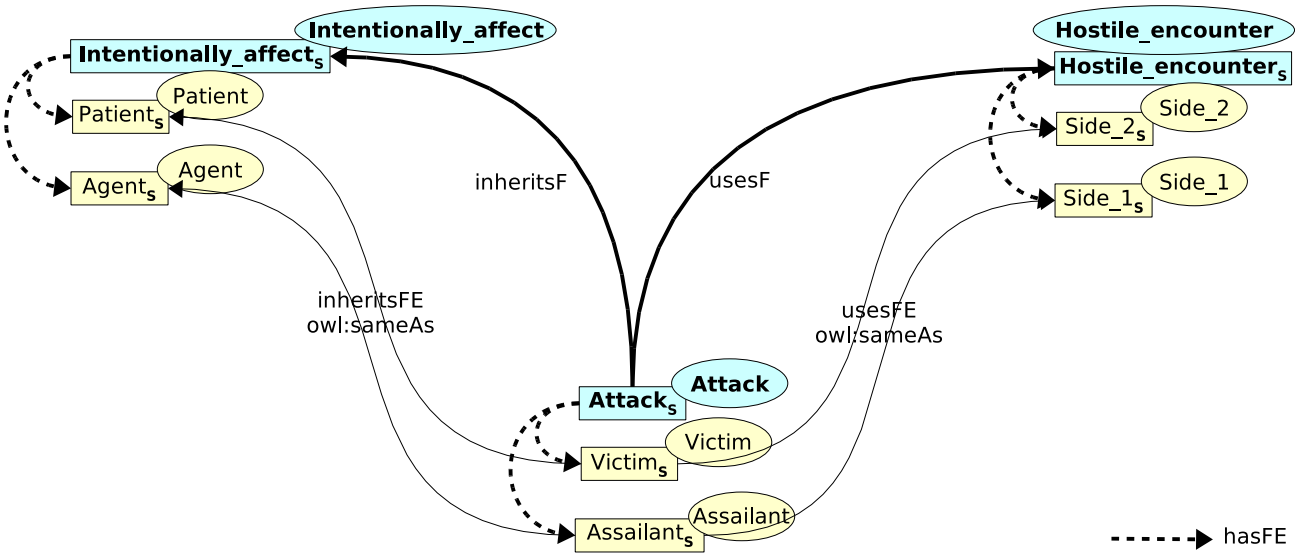


Figure 4: Connecting the Attack instance (Step 2 of Annotation Ontology generation)

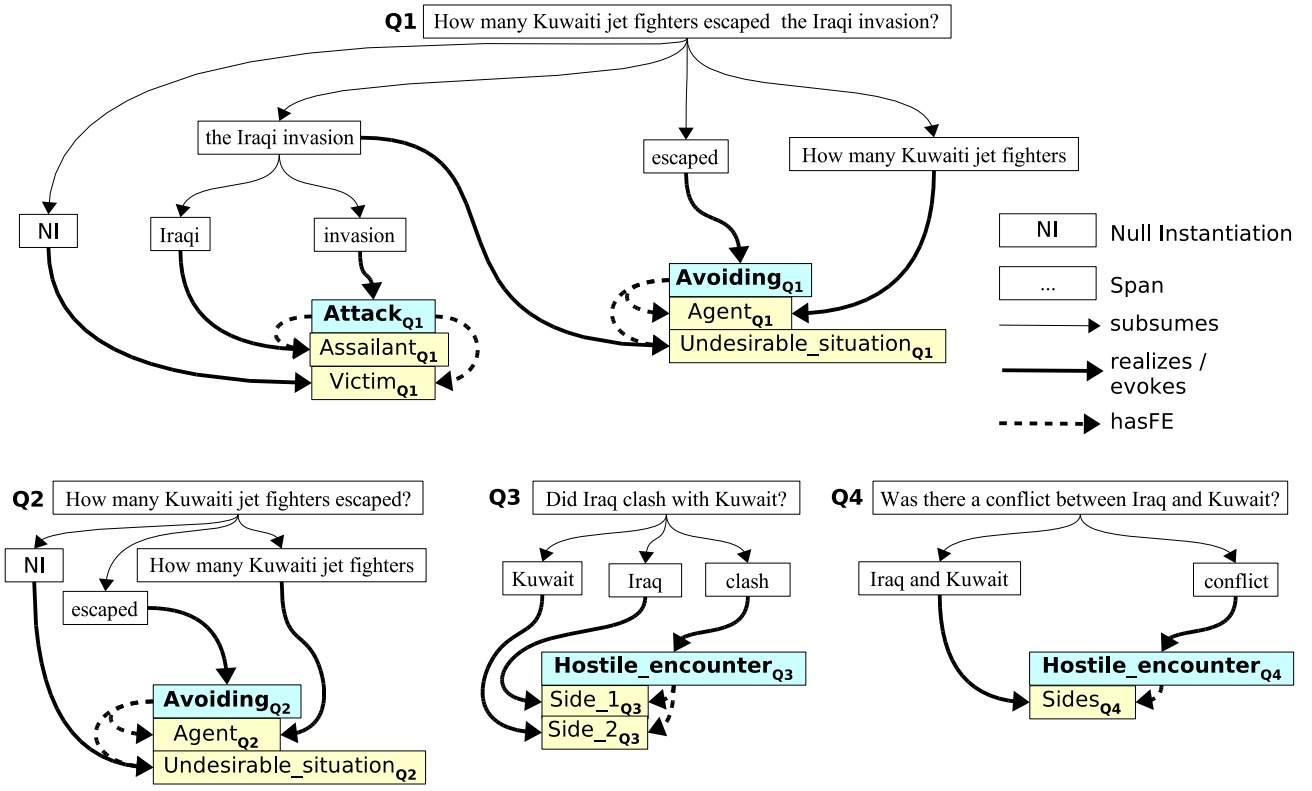


Figure 5: Abridged Annotation Ontologies for example questions

Since RacerPro is a reasoner (and no NLP tool), checking the compatibility of Spans is limited to checking syntactic equality. Therefore, the Span *48 Kuwaiti jet fighters* does not match the Span *How many Kuwaiti jet fighters*. We can, however, easily determine the Spans that are supposed to be compatible in order to yield an answer. Then Span compatibility can be determined by other NLP tools such as question type recognizers.

Question *Q2* is simpler than Q1 because we are asking for only one frame in which one FE is null instantiated. In this case our approach only using a reasoning engine yields the final answer:

Undesirable_s.S $\rightarrow$ the Iraqi invasion $\surd$
Agent$_S$ $\rightarrow$ 48 Kuwaiti jet fighters $\times$

Notice that here the Span *the Iraqi invasion* is in fact compatible with the corresponding null instantiated Span from the question. If we are asking for a null-instantiated FE we only check whether the corresponding FE has a proper filler.

Question *Q3* leverages our ontology structure in that it asks for the more general event of a Hostile_encounter. Our approach proceeds as follows:

1. Get evoked frames instances, FEs, and Spans:
   $\text{Hostile\_encounter}_{Q3} \rightarrow \text{Side\_1}_{Q3} \rightarrow \text{Iraq}$
   $\rightarrow \text{Side\_2}_{Q3} \rightarrow \text{Kuwait}$

2. Determine the direct classes for frame/FE instances:
   $\text{Hostile\_encounter}_{Q3} \rightarrow \text{Hostile\_encounter}$
   $\text{Side\_1}_{Q3} \rightarrow \text{Side\_1}$
   $\text{Side\_2}_{Q3} \rightarrow \text{Side\_2}$

3. Obtain corresponding frame/FE instances:
   $\text{Hostile\_encounter} \rightarrow \text{Hostile\_encounter}_S$
   $\text{Side\_1} \rightarrow \text{Side\_1}_S$
   $\text{Side\_2} \rightarrow \text{Side\_2}_S$

4. Determine compatible Spans:
   $\text{Side\_1}_S \rightarrow \text{Iraqi} \quad \times$
   $\text{Side\_2}_S \rightarrow \text{NI} \quad \times$

   RacerPro can infer the above because Side_1_S is the same as Assailant_S and Side_2_S is the same as Victim_S. However, the Span *Iraqi* is not compatible with *Iraq* and the null instantiated Span in the sentence is not compatible with the Span *Kuwait*. We should be able to determine that the adjective *Iraqi* is compatible with the noun *Iraq* by using the corresponding WordNet synsets: *Iraqi* should be connected to *Iraq* via the *pertainym* relation. From the fact that the Kuwaiti jet fighters escaped the Iraqi invasion we could infer that there was a clash between Iraq and Kuwait, which clearly needs world knowledge not present in FrameNet.

Question *Q4* is even more problematic than question Q3 because in Q4 the FE Sides is annotated, which is in an *Excludes* relation to both FEs Side_1 and Side_2. In FrameNet we find, however, no formal relation saying that the FEs Side_1 and Side_2 "make up" the FE Sides; also, it is unclear how the Span *Iraq and Kuwait* should be distributed to Side_1 and Side_2. Therefore, using only FrameNet, RacerPro would infer the correct answer for this example, but it would do so for *any* conflict.

6. Evaluation

Our example shows that in principle we can employ a DL reasoner for querying the FrameNet Ontology and Annotation Ontologies. However, even using a state-of-the-art DL reasoner like RacerPro, inference performance is not satisfying. For our small example, RacerPro takes several seconds for the final query. This is because our we are dealing with a large amount of data: The FrameNet Ontology contains about 100,000 triples, Annotation Ontologies contain on the order of 10,000 to 30,000 triples depending on the complexity and the amount of annotated text (even though we do not import the whole FrameNet Ontology). Moreover, checking Span compatibility requires other external tools. On the other hand, determining the direct classes of the frame and FE instances (Step 2) and getting the other instances of these classes (Step 3) can be done by a DL reasoner; especially since these tasks can require querying other ontologies.

We envision DL inference as a component of a lexical semantic reasoner. The DL component has to be integrated with other inferencing techniques for temporal, spatial, and event structure inference to adequately model the different dimensions of lexical semantics. For example, a model of predication must have the ability to capture linguistic aspect (modeling actions, state changes, resources, and event structure). This requires extensions to model situations, variables, and fluents and unification, which leads to full first order logic. The price for this expressiveness is of course less effective inference. An alternative approach is to not lose the efficiency of the DL reasoner for certain purposes, but to integrate it to special purpose representation and reasoning mechanisms for aspect and event structure, such as (Narayanan, 1999).

In previous work (Narayanan and McIlraith, 2003), we successfully explored one method of accomplishing this integration. We used an extension of OWL (OWL-S) that has a rich process ontology and was designed to model transactions and services on the Web.[11] Especially, OWL-S has an expressive process model ontology that provides a declarative description of the properties of the events we wish to reason about. The process model ontology makes fine-grained distinctions relevant to reasoning about event structure and the DL reasoner is able to perform consistency checks on the ontology.

As part of the integration, we implemented an OWL-S interpreter that translates OWL-S markups to the simulation and modeling environment KarmaSIM (Narayanan, 1999), which is able to reason effectively about events. This allows the system to integrate interactive simulations and use a variety of analysis techniques to model the temporal evolution of events and to perform inference related to linguistic aspect. We believe this mode of using OWL ontologies as structured interfaces (with special purpose ontologies) and using the DL reasoner for consistency checks will carry over to integrating with spatial and temporal reasoners.

7. Conclusion and Outlook

In this paper we have outlined our design to bind the FrameNet lexicon to multiple ontologies, thus providing filler types for FEs almost for free. Such filler types may then be used by various applications. As a first step toward such ontology bindings we have translated a crucial portion of FrameNet to OWL DL. The FrameNet Ontology provides a formal specification of FrameNet itself by means of ontology classes and existential restrictions on these classes. Annotation Ontologies are generated for specific FrameNet-annotated sentences thus filling and satisfying the FrameNet Ontology. That way, FrameNet and annotated sentences become available for reasoning. Also, we provide a solid basis for binding FrameNet to arbitrary OWL ontologies by using OWL itself for specifying the bindings. The resulting homogeneous ontology has a number of advantages over using proprietary techniques for specifying ontology bindings, particularly when it comes to tool support.

In the future we plan to link the FrameNet Ontology to several other ontologies in the OWL format, in order to restrict filler types for FEs. This is particularly useful for automatic frame parsing and role annotation in conjunction with

[11] see www.daml.org/services/

named entity recognizers. Also, we plan to evaluate the utility of DL reasoners in a fully fledged question answering system. Finally, we will translate FrameNet to other ontology languages such as KIF or CycL, in order to link FrameNet to SUMO or Cyc ontologies.

Acknowledgements

The first author enjoys funding from the German Academic Exchange Service (DAAD). The FrameNet project is funded by the AQUINAS project of the AQUAINT program.

8. References

A. Burchardt, K. Erk, and A. Frank. 2005. A WordNet detour to FrameNet. In *Proceedings of the GLDV 2005 Workshop GermaNet II*, Bonn.

K. J. Burns and A. R. Davis. 1999. Building and maintaining a semantically adequate lexicon using cyc. In Evelyne Viegas, editor, *Breadth and Depth of Semantic Lexicons*. Kluwer.

K. Erk and S. Padó. 2005. Analysing models for semantic role assignment using confusability. In *Proceedings of HLT/EMNLP-05*, Vancouver, Canada.

K. Erk and S. Padó. 2006. Shalmaneser – a toolchain for shallow semantic parsing. In *Proceedings of LREC-06*, Genova, Italy. to appear.

C. J. Fillmore, C. R. Johnson, and M. R. L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

C. J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, (280):20–32.

I. Horrocks. 1998. The FaCT system. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, number 1397 in Lecture Notes in Artificial Intelligence, pages 307–312. Springer-Verlag, May.

D. B. Lenat. 1995. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38.

K. Litowski. 2004. Senseval-3 task: Automatic labeling of semantic roles. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 9–12. Association for Computational Linguistics.

S. Narayanan and S. McIlraith. 2003. Analysis and simulation of Web services. *Comput. Networks*, 42(5):675–693.

S. Narayanan, C. F. Baker, C. J. Fillmore, and M. R. L. Petruck. 2003. FrameNet meets the semantic web: Lexical semantics for the web. In *The Semantic Web — ISWC 2003*, pages 771–787. Springer-Verlag, Berlin.

S. Narayanan. 1999. Moving right along: A computational model of metaphoric reasoning about events. In *Proceedings of the /National Conference on Artificial Intelligence (AAAI'99)*, pages 121–128. AAAI Press.

I. Niles and A. Pease. 2001. Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine.

I. Niles and A. Pease. 2003. Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE 03)*.

J. Ruppenhofer, M. Ellsworth, M. R. Petruck, and C. R. Johnson, 2005. *FrameNet: Theory and Practice*. ICSI Berkeley. www.icsi.berkeley.edu/˜framenet/book/book.html.

J. Scheffczyk and M. Ellsworth. 2006. Improving the quality of FrameNet. In *Proc. of the Wkshp. on Quality assurance and quality measurement for language and speech resources*, Genoa, Italy. to appear.

M. Wessel and R. Möller. 2005. A high performance semantic web query answering engine. In *Proc. International Workshop on Description Logics*.

Towards sounder taxonomies in wordnets

Bolette Pedersen[1], Nicolai Hartvig Sørensen[2]

[1]Center for Sprogteknologi – Københavns Universitet
Njalsgade 80, Copenhagen, Denmark
bolette@cst.dk
[2] Det Danske Sprog- og Litteraturselskab
Christians Brygge 1, Copenhagen, Denmark
nhs@dsl.dk

Abstract

Hyponymy constitutes the main organising mechanism behind semantic lexical networks such as standard wordnets. However, hyponymy in all its variants does not as such provide the best basis for a sound taxonomy – something which is required if we are to expect that wordnets can be fully integrated as valuable resources in for instance Semantic Web ontologies. The paper describes how we in the DanNet project strive towards the development of a sound taxonomy inspired by Cruse's three category model including *Natural kinds, nominal kinds*, and *functional kinds*, as well as by Pustejovsky's four-dimensional semantic model. On this approach, we provide guidelines and linguistic tests for determining whether or not a term should be considered and encoded as a sound taxonym in the lexical network, and we suggest that non-taxonomical terms are encoded as orthogonal to the taxonomy.

1. Introduction

Hyponymy constitutes the main organising mechanism behind semantic lexical networks such as standard wordnets. In these, the lexical hierarchy is given by defining hyponymy relations between particular senses of words or – in wordnet terminology – between *synsets* (cf. Miller 1998). Many wordnets are built on the basis of existing lexical data in terms of traditional lexicons. In these lexicons the hyponymy relations are implicitly expressed via the *genus proximum* of the definitions, and it seems straightforward to assume that this information can be imported into the network as a first organisation of the network taxonomy. Taxonomy is here defined in terms of hyponymy *and* mutual incompatibility between co-hyponyms (Lyons) as well as in terms of Cruse (1991) who describe 'simple' hyponymy by *An X is a Y*, whereas taxonomy is more restricted in that it can be described generically as *An X is a kind/type of Y*.

However, as pointed out by Guarino & Welty (2002) and several other formal ontologists, improperly structured taxonomies make models confusing and difficult to reuse or integrate, and this fact counts in particular for wordnets if we aim at integrating them as part of computational models and not only see them as lexicographical repositories organised in a slightly different way than traditional lexicons. Computational models make heavy use of inheritance mechanisms, and such mechanisms are easily messed up if the taxonomy is not sound. In fact, if we foresee an integration of lexical resources in terms of linguistic ontologies for the purpose of Semantic Web technologies in the future, a critical analysis of the nature of hyponymy in traditional lexicons and a more principled reorganisation of these in the lexical networks being built seem to be a highly relevant focus for the work of computational lexicographers.

In the Danish wordnet project DanNet we strive towards a principled organisation of the taxonomy inspired by Wierzbicka (1984) and further specified in Cruse's three category model (Cruse 2002) including as a minimum three subdividing categories: *Natural kinds*, *nominal kinds*, and *functional kinds* as well as by Pustejovsky's four-dimensional semantic model (Pustejovsky 1995). In the paper we exemplify the different categories of hyponymy on the basis of Danish lexical data and discuss why we believe that a categorising approach opens for a more sound and well-structured taxonomy better geared for Semantic Web and other applications.

2. The DanNet project

DanNet (cf. Pedersen et al. 2006) is a collaborative project between a research institution, Center for Sprogteknologi, University of Copenhagen, and a literary and linguistic society, Det Danske Sprog- og Litteraturselskab under The Danish Ministry of Culture. DanNet is being built on the basis of two Danish lexical resources developed at these two institutions respectively, namely a large corpus-based paper dictionary of modern Danish (Den Danske Ordbog, henceforth DDO, cf. Lorentzen 2004) comprising approx. 100,000 senses and SIMPLE-DK, a computational semantic lexicon for Danish comprising descriptions of 10,000 concepts in the so-called SIMPLE model (Semantic Information for Multifunctional, Plurilingual Lexicons, jf. Lenci et al. 2000 and Pedersen & Paggio 2004). These two resources are reused by automatically extracting and thereafter manually adjusting the genus proximum information as well as other central semantic relations such as part-of relations, purpose relations, synonymy and antonymy relations. The project runs from 2005 to 2007, in which period the plan is to achieve a wordnet of approx. 40,000 concepts of which 30,000 will be constituted by nouns.

3. Hyponymy in traditional dictionaries and in wordnets

It is a basic assumption in DanNet as well as in other wordnets that all synsets should be assigned a hypernym. This assumption, we believe, reflects the general intuition of the language user, who will answer in the affirmative when asked questions like *Is a birch a tree?*, *Is a jeep a car?* and also *Is a getaway car a car?* even though the last question differs from the first two in a way that we will discuss further below.

This assumption also constitutes the underlying basis for the lexicographer when choosing a genus proximum as a part of the sense definition in traditional dictionaries. For illustration, we find the following definitions of different trees in DDO:

(1) *kirsebærtræ: "**træ** der bærer kirsebær, og som har hvide el. lyserøde blomster der sidder enkeltvis el. i klaser"* (cherry tree: *tree* which carries cherries, and which has white or pink flowers that are situated separately or in clusters)

(2) *jeep: "mindre, firehjulstrukken **bil** med stærk motor som gør den velegnet til kørsel i ujævnt terræn"* (jeep: small, four-wheel driven *car* with a strong motor which makes it well suited for driving in rough terrain)

(3) *flugtbil: "**bil** der benyttes af en røver, en kidnapper e.l. under flugten fra den kriminelle handling"* (getaway car: *car* which is used by a robber, a kidnapper etc. during the escape from the criminal act)

The same interpretation underlies the use of the two inverse relations *has_hyperonym* and *has_hyponym* in wordnets. In Princeton WordNet, for instance, we thus find e.g. *birch* and *bonsai* as apparently equal hyponyms of *tree*. However, behind this description lies a clear generalisation over a diversity of linguistic data where *birch* and *bonsai* are not equal taxonyms. Actually, Miller (1998:35) acknowledges the fact that the hyponymic relation in WordNet represents more than one semantic relation; and that this underspecification represents a serious problem. Consider further the (partial) hyponymy hierarchy of the noun *bil* ('car') and (some of) its hyponyms in figure 1.

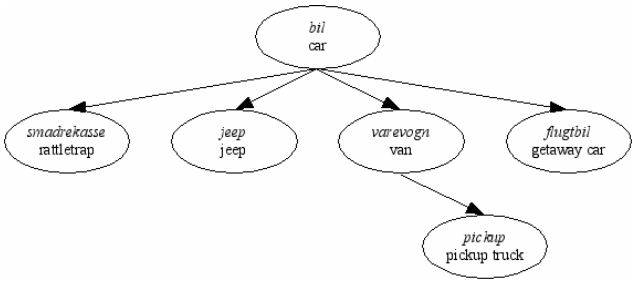


Figure 1: *Bil* ('car') and some of its hyponyms

Jeep ('jeep') and *varevogn* ('van') are internally incompatible: a car cannot be a jeep and a van at the same time. But a robber might use a jeep as a getaway car. And the jeep might at the same time happen to be a rattletrap. In other words, the ontological status of the involved car terms differ essentially.

Similarly, we find hyponyms like the (partial) hierarchy of the hyponyms of *træ* ('tree') in Figure 2.[1]

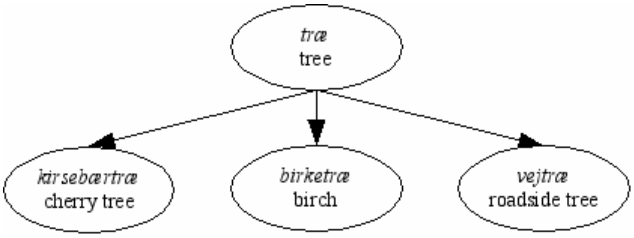


Figure 2: *Træ* ('tree') and some of its hyponyms

An instance of a cherry tree cannot at the same time be considered a birch. But both cherry trees and birches might in a certain context function as roadside trees.

As a first conclusion, it seems that the hyponymy relations involved in the Danish equivalents to 'jeep', 'van', and 'birch' and 'cherry tree', respectively, can be seen as *taxonomic* hyponyms. In contrast, the problematic cases of hyponymy involved in 'rattletrap, old car', 'getaway car' and 'roadside tree' must be seen as somewhat *non-taxonomic* hyponyms describing other dimensions of meaning than that of the classical taxonomy. However, we need to define the concept of taxonomy a little further as well as to achieve a deeper understanding of hyponymic relations as such. To this end we have – as already mentioned - been inspired by Cruse's three category model (Cruse 2002) comprising *Natural kinds*, *nominal kinds*, and *functionals*. It is our aim to make this three-category model function also as a practical device for the encoding of the wordnet.

4. Towards a structuring of hyponymy

4.1. Natural kind terms

On Cruse's approach (1991, 2002) natural kinds are defined as naturally occurring things like animals, plants and naturally occurring materials and substances like wood, stone and water. He states (2002:18) that 'the names of natural kinds behave to some extent like proper names in that they show referential stability in the face of quite radical changes in the speaker's beliefs concerning the referent'.

Put in another way, natural kinds generally possess what Guarino & Welty label *rigid properties* i.e. properties that guarantee identity through change (thus according to Guarino & Welty *person* possess rigid properties whereas *student* does not). Such entities are generally assumed to be good candidates for the skeleton of a sound taxonomy and therefore constitute a good starting point for building the lexical network.

For natural kinds it is generally true to say that *X is a kind of Y* as in *a birch is a kind of tree*, and likewise the two hyponyms of 'tree', 'cherry tree' and 'birch', are mutually incompatible as shown in the previous section.

[1] We here disregard the question of whether the hypernym of 'cherry tree' and 'birch' are better described as 'fruit tree' and 'leaf-bearing tree', respectively.

Thus they fulfill our restricted definition of taxonomy, and they obey the general rules of inheritance where a subsense inherits the characteristics of its supersense.

Another characteristics of natural kinds is that we are not able to define easily what distinguishes one natural kind from another. What distinguishes a cherry tree from a birch? - Well, bearing cherries springs to mind, but is clearly not sufficient, and how about distinguishing between a birch and an oak? The shape of the leaves and the color of the cortex are good suggestions, but these features do not completely describe and distinguish the trees from each other[2].

'Roadside tree', on the other hand, *can* be defined with a single feature, namely 'a tree situated in the roadside'. It also inherits the characteristics of a tree, but cannot be described as *a kind of tree*. 'Roadside tree' should therefore not be considered a natural kind term, even though it refers to a naturally occurring thing.

4.2. Functional kind terms

The second category is referred to as functional terms. Functional kind terms typically refer to artifacts whose function plays a central role in their definition. They share certain features with natural kinds, mutual incompatibility is the most general characteristics: a bus cannot be a car since buses and cars are both *types of* vehicles. Most often the hypernym itself denotes the basic function of the hyponym as in *a car is a vehicle*.

Also, just like we were not able to define easily what distinguishes a cherry tree from a birch, we are not able to define easily and uniquely what distinguishes a bus from a car, both being hyponyms of *motorkøretøj* ('motorized vehicle'). The size of the two vehicles differs, as well as the number of seats. But again this is not sufficient; we are not able to define the differences with a single feature.

As a consequence, we describe functional kind terms as taxonyms in DanNet just as we did with natural kinds.

Figure 1 gave examples of functional terms: *varevogn* ('van'), *jeep* and *pickup* ('pickup truck'). The two former terms are *types of cars*. The latter is a *type of van*.

To complicate the matter, many natural objects are often referred to by means of functional kind terms. For instance persons can be referred to with functional kind terms and thus expose taxonomy-like relations. Even if it does not sound right to say that 'a doctor is a type of person', we do find taxonomical structures *within* the professional domain so that we can say 'a surgeon is a type of doctor' and in these cases the hyponym inherits its basic function from it hypernym. Also, prototypically, co-hyponyms in the professional domain are somewhat mutually incompatible: typically you do not have two professions at the same time. For practical reasons, such functional kind terms referring to professions are therefore interpreted and encoded as taxonyms in DanNet even if they do not really fulfill the restrictions of rigidity over time.

[2] Actually, Ruus (1995:130) argues that some of these hyponyms are characterised by the fact that a *limited* set of features can distinguish them from each other. She uses Grandy's terminology and calls such hyponyms *contrast sets*.

4.3. Taxonomy and regular polysemy

In several cases, even clear-cut taxonomy relations seem to operate along different dimensions. Most often polysemy proves to play a central role in such situations, and particularly regular polysemy, i.e. cases where groups of words follow the same pattern of meaning shift. An often used example is the case of semiotic artifacts such as *bog* (book). Both *paperback* (paperback) and *kogebog* (cook book) are intuitively functional terms and thereby proper taxonyms of *bog* as can be illustrated in figure 3.

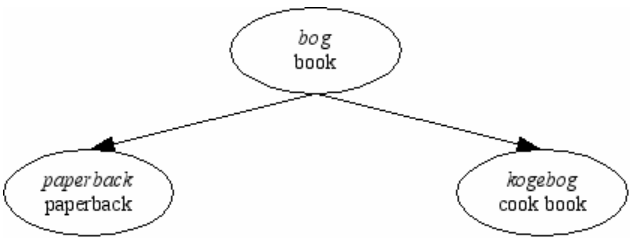


Figure 3: Hyponyms of *bog* (book).

However, they refer to different perspectives of *bog* namely the physical object and the semiotic content, respectively. In these cases we rely on the corpus-based sense distinctions established in DDO, where we find a subdivision into two senses:

*Bog*_sense 1: "*trykte el. beskrevne blade af papir indbundet el. på anden måde sammenhæftet i rækkefølge så de danner en helhed"* (printed or written pages of paper which have been bound or stapled in some way so that they constitute a whole).

*Bog*_sense 2: "*tekst der står på disse trykte el. beskrevne, indbundne el. sammenhæftede blade, el. som findes i nyere medier som lydbånd el. cd-rom; en bogs indhold"* (text written on these printed or written pages or in newer media such as audio tape or cd rom; the content of a book).

The acknowledgement of regular polysemy in such cases eases taxonomy building since *paperback* can now be seen as a taxonym to sense 1, whereas *kogebog* can be interpreted as a taxonym to sense 2 as depicted in figure 4. However, not all cases of regular polysemy prove to give rise to clear distributional differences in the corpus, and in such cases they are not established as distinct senses in DDO.

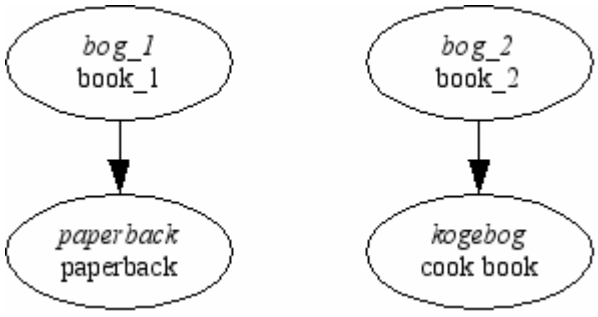


Figure 4: Two taxonomies for *bog* ('book')

4.4. Non-taxonomical hyponymy: Nominal kind terms

In contrast to natural kinds and functionals, nominal kinds cannot be described as *a kind of* or *a type of*. As a further characteristic, the relation between nominal kinds and their hypernyms can typically - in contrast to the two former categories – be captured in terms of a single differentiating feature (Cruse 2002:18) as we saw for *vejtræ* in section 4.1.

As we have already seen, we find nominal kind terms in both natural domains and artifactual domains. If we for instance look deeper into natural taxonomies as found with plants and natural substances, we see that several lexical terms denote non-taxonomic dimensions of plants and substances. We saw this in the case of *vejtræ* ('roadside tree'), but also in the case of natural substances, we find cases like *garvestof* ('tanning agent') which is defined as *vegetabilsk, animalsk el. mineralsk stof som kan optages af dyrehuder og derved omdanne dem til læder* (vegetal, animal or mineral substance which can be absorbed by animals' skin and thereby convert it into leather).

Since such dimensions of meaning really tend to mess up the natural taxonomies, it seems obvious *not* to consider such senses as taxonyms, but as nominal kinds orthogonal to the taxonomy.

To give additional examples of nominal kinds, let us consider lemmas with the hypernym *person* other than professions. DDO contains more than 4,000 lemmas with this hypernym (such as *passager* ('passenger'), *idiot* ('idiot'), *læser* ('reader'), *medlem* ('member') etc.) and generally they are all nominal kinds since they are not *kinds of persons*, but describe dimensions of persons with different characteristics highlighted, some of which are stable over time and some of which are only related to a specific situation. As mentioned, they are generally easily definable with a single feature or very few features, in other words we can define a passenger as a person that travels in a vehicle without being the conductor, a reader as a person that reads etc. An additional test that can be applied is the - however not fully waterproof - negative 'kind of' test. For example, it does sound odd to say that 'a passenger is a kind of person'.

4.5. Describing non-taxonomic hyponyms as orthogonal to the taxonomy

Although mutual incompatibility between co-hyponyms do occur with nominal kinds, the general characteristics of this category is that they *are* compatible: A car may very well be a rattletrap and a getaway car at the same time. Nominal kinds are also normally compatible with taxonomic co-hyponyms (and any taxonym of these): A car may be a van (or a jeep or a pickup truck) *and* a getaway car *and* a rattletrap at the same time.

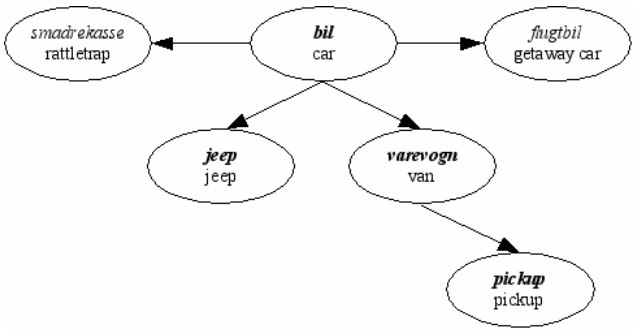


Figure 5: *Bil* ('car') with two orthogonal hyponyms

To illustrate this we may re-arrange figures 1 and 2 into figures 5 and 6 respectively. Note that in this representation for instance *vejtræ* ('roadside tree') in figure 6 is located in a different position than its co-hyponyms ('birch' and 'cherry tree') which reflects the observed fact of compatibility: Just like the term *bil* ('car') may be used instead of a *pickup* ('pickup truck'), so may *smadrekasse* ('rattletrap') and *flugtbil* ('getaway car') –

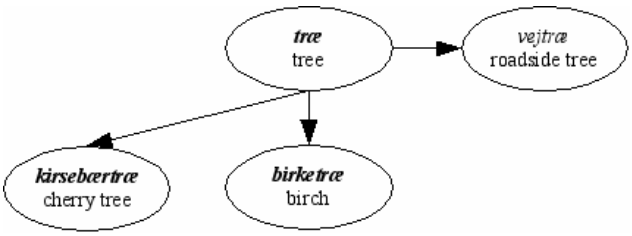


Figure 6: *Træ* ('tree') with one orthogonal hyponym

and in fact any hyponyms of these. The nominal kind terms may be regarded as being at the same level as their hypernym when considering the taxonomic hyponyms. In other words, the non-taxonomic hyponyms are encoded as orthogonal to the taxonomic hyponymy relations[3].

4.6. Linguistic tests for encoding

We have now identified two types of terms that expose taxonomic characteristics: natural kind terms and functional terms, and we have briefly sketched some of the English tests that can be applied to identify the two types. Also we have discussed nominal kinds and described how they differ from the former two groups. What we need now is to sum up the characteristics in terms of *linguistic tests* that can be applied in the practical encoding of Danish terms.

In Danish, a pair of tests can be identified corresponding to the English ones, thus in DanNet the test *X er en slags Y*, corresponding more or less directly to the English test *X is a kind of Y* is applied as a test for

[3] This was actually suggested to us by P. Vossen during a discussion at the DanNet seminar in Copenhagen 2005.

taxonyms. Likewise with *X er en type Y* corresponding to X *is a type of Y*. We do not find, however, that the two tests clearly distinguish functional terms and natural kind terms from each other.

An additional test for identifying taxonyms is the 'single-feature test' previously mentioned. If the 'single-feature test' does not apply, that is if we need an indefinite set of features to describe a concept from its superconcept, we see it as a sign of the fact that we are dealing with a taxonymic term.

One apparent weakness of this test, is to determine the semantic *width* of the one feature. We claim that the difference between a *tree* and a *roadside tree* is one feature that may be paraphrased as 'situated in the roadside'. But one may argue that what defines a *birch* in relation to a *tree* is also exactly one feature – a feature that might be paraphrased as 'being a birch'.

However, the difference between taxonomic and non-taxonomic (orthogonal) hyponyms can also be defined in terms of the fact that the latter are not restricting the set denoted by the hypernym, but rather the context in which it occurs. A taxonym like 'cherry tree' restricts the set of trees to only a certain kind of trees, while a 'roadside tree' makes no such claim. A 'roadside tree' exactly like its hypernym, 'tree', may denote any kind of tree as we saw in section 4.5. A linguistic test for nominal kinds could therefore be *X is any kind of Y which..* or in Danish *X er en hvilken som helst slags Y som..*

5. Experimental structuring of orthogonal hyponyms

In the previous sections we have described why and how we in DanNet distinguish formally between what we call taxonyms and other hyponyms. Both taxonyms and non-taxonyms are encoded as hyponyms in order to obey wordnet standards. However, an attribute on the hyponymy relation distinguishes the two.

This distinction ensures a more direct mapping of taxonyms into ontological classes as foreseen in formal ontologies such as SUMO (Suggested Upper Merged Ontology) which is used as the formal framework for several wordnets being built currently. Non-taxonyms are encoded as orthogonal to the basic taxonomy as illustrated in figure 5 and 6. However, a further organisation of the different dimensions that such orthogonal hyponyms can take would probably be fruitful. Such an approach is seen in some recent terminological work, as seen for instance in Madsen et al. (2004:94) where terms are subdivided by the dimensions that the terminologist find most crucial for the domain.

We might however also structure the orthogonal hyponyms more roughly in terms of qualia structure functions, and thereby get a better understanding of them as groups and maybe even be able to explain some of the incompatibility patterns that they exhibit.

For instance, if we look again at the 'person' domain, it includes a very large number of orthogonal hyponyms that do exhibit some degree of internal, mutual incompatibility: a beauty is not likely to be a beast, a genius is not likely to be an idiot (on the other hand who knows?).

As mentioned previously, the Danish SIMPLE lexicon serves as one of the basic resources on which we build DanNet. A central characteristics of the SIMPLE model is that it applies Pustejovsky's four-dimensional Qualia model as an underlying framework for the lexicon.

One of the fundamental assumptions behind this model is that lexical items vary in their internal complexity (cf. Lenci *et al.* 2000a:15-19 and 25-27). This can be understood in two ways: 1) how many dimensions of meaning are associated with an item and 2) how many senses an item incorporates. Pustejovsky's theory of lexical meaning (Pustejovsky 1995), relying on the qualia structure and on a highly structured lexicon in general, constitutes the backbone of the SIMPLE ontology since it proposes a strategy for accounting for exactly this internal complexity of meaning.

The notion of Qualia Structure as a basis for lexical networks is to some extent taken over in the EuroWordNet Ontology (Vossen (ed.) 1999) and particularly in DanNet. Reinterpretated in Cruse's words (Cruse 2000:118), the four qualia roles include:

- *the formal role* encompassing the dimension of seeing something as a kind,
- *the constitutive role* encompassing the dimension of seeing something as a whole consisting of parts (in SIMPLE a large range of semantic features and relations typically concerning the internal structure of the concept is expressed via this role[4]),
- *the telic role* encompassing the dimension of seeing something as having a certain function, and finally
- *the agentive role* encompassing the dimension of seeing something from the point of view of its origin.

A provisional division into the three qualia dimensions (we exclude here the formal dimension since it is already described via the hypernym) of some of the nominal kind terms already presented could be:

- the constitutive role: *idiot* ('idiot'), *geni* ('genius'), *smadrekasse* ('rattletrap')
- the telic role: *vejtræ* ('roadside tree'), *garvestof* ('tanning agent'), *flugtbil* ('getaway car')
- the agentive role: *fodgænger* ('pedestrian'), *cyklist* ('cyclist')

Note that nominal terms referring to artifacts can only fill out the telic role if it concerns a *non-prototypical* use, as seen in *flugtbil* (getaway car) – flight is not the prototypical purpose of a car.

Apart from providing a satisfactory systematisation of the nominal kind terms, a division into qualia dimensions may also explain why we in certain cases find incompatibility between these terms. It seems that terms belonging to the same qualia dimension are often incompatible or at least require a special context. You are not likely to be an idiot and a genius at the same time, nor are you typically a pedestrian and a cyclist at the same time. Incombatibility between dimensions, on the other hand, does not tend to occur: a person may very well be an idiot and a pedestrian at the same time.

[4] Examples of features are *sex*, *age* and *connotation*; whereas examples of relations are: *has_colour*, *lives_in* etc.

In spite of these interesting finding, dimensions on orthogonal hyponyms are *not* encoded in DanNet at its current stage. Further investigations are required in order to establish practically useful tests for the encoders and also the inheritance mechanisms need to be studied further.

6. Encoding of hyponymy in DanNet

As previously mentioned, we apply a strategy of substantial re-exploitation of existing lexicographical data in DanNet; therefore much energy has been invested in developing an efficient tool for flexible encoding. The encoding tool directly incorporates and makes immediately available the data from our existing resources, DDO and SIMPLE-DK, meaning that encoding in essence consists of accepting or adjusting pre-encoded information.

We work domain-wise since genus proximum information is supplied in a directly extractable field in both DDO and SIMPLE-DK. This means, for example, that fauna or vehicles are treated in one go, the encoder being automatically supplied with all the pre-encoded hyponyms of these two senses. Figure 7 shows a screen dump of the encoding of vehicles which is performed by means of a so-called wizard which suggests a hypernym to the lexical item based on the reused lexical resources – in this case the hypernym *personbil* ('car') is suggested as a hypernym to *stationcar* ('estate car'). Obviously, the encoder receives no guidance in the tool regarding whether to encode the hyponym as a taxonym or not since this information cannot be extracted automatically from neither DDO or SIMPLE. Here the elaborated tests come into play. Note that taxonyms are here visualised in circles, whereas nominal kinds are provisionally visualised by means of rhombs.

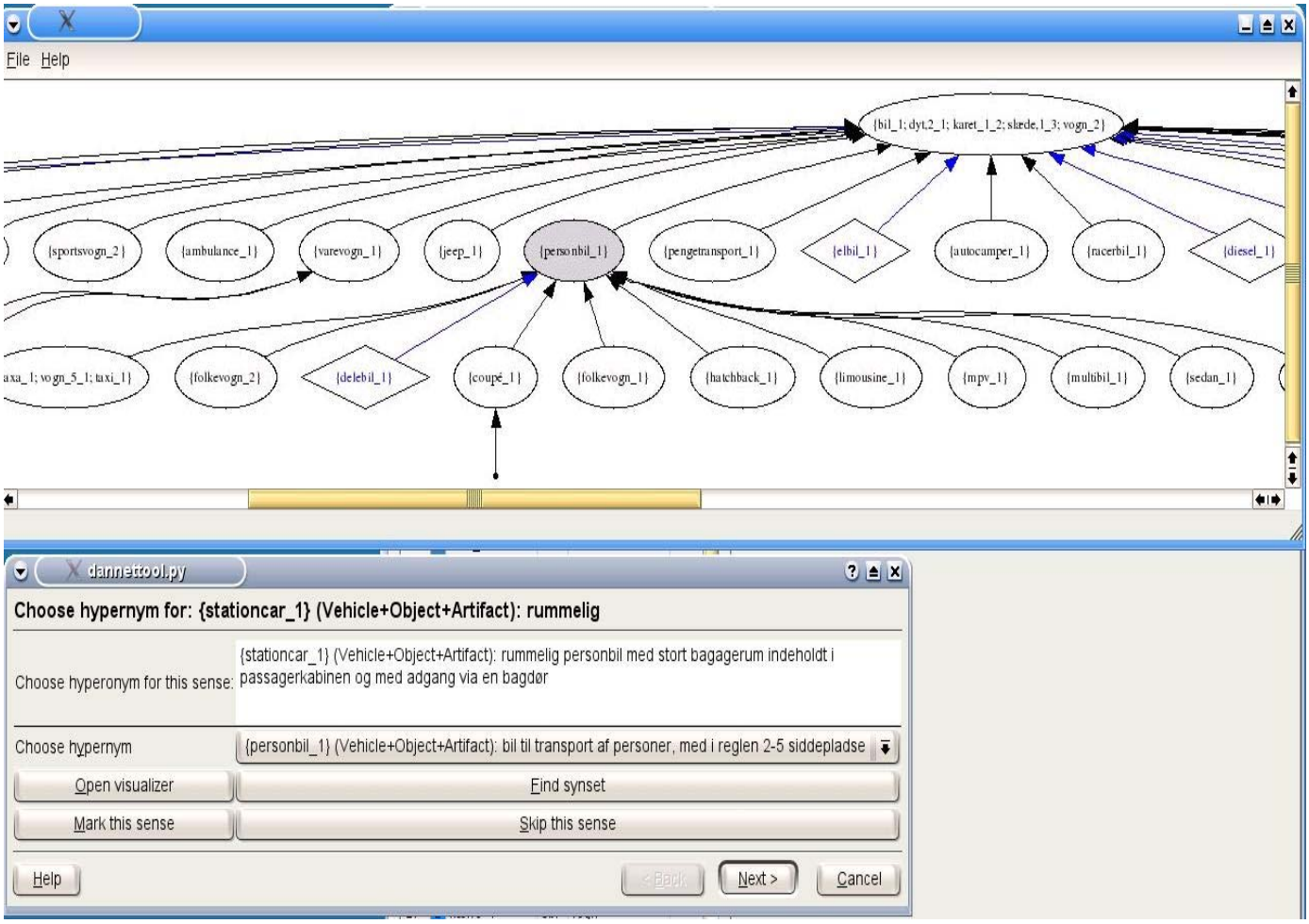


Figure 7: DanNet encoding tool

7. Concluding remarks

Lexical items prove to expose hyponymy of great variety. Classical wordnets generally generalise over this variation and do not distinguish between taxonomic and non-taxonymic types of hyponymy. In Miller (1998) this is acknowledged as a serious weakness of Princeton WordNet, and also several formal ontologist have pointed out this fact as a problem.

We believe that a distinction like the one we have presented in this paper – distinguishing between natural kinds, nominal kinds and functional kinds – provides a good starting point for a wordnet better tuned for future applications in for instance Semantic Web technology.

When mapping from lexical items in a lexical network to concepts in an ontology, hyponymic variety proves crucial since the ontological status of the hyponyms differs. Cruse (1990) tentatively states that lexical items may have meaning properties not accounted for by their associated concepts. Nominal kind terms tend to fall into this category, and it seems inevitable that computational lexicographers examine and describe these further (as we have already provisionally started upon in section 5) in future lexical networks.

References

Cruse, D. A. *Lexical Semantics*. 1991. Cambridge University Press, Cambridge.

Cruse, D..A. 'Prototype theory and lexical semantics'. In: A. L. Tsohatzidis (ed.), *Meanings and Prototypes: Studies in Linguistic Categorization*. Routledge, London.

Cruse, D. A. *Meaning in Language*. 2000. Oxford University Press.

Cruse, D.A. 'Hyponymy and Its Varieties'. 2002. In: R. Green, C.A. Bean, & S. H. Myaeng (eds.) *The Semantics of Relationships: An Interdisciplinary Perspective, Information Science and Knowledge Management*. Springer Verlag.

DDO = Hjorth, E., Kristensen, K. et al. (eds.). 2003-2005. *Den Danske Ordbog 1-6* ('The Danish Dictionary 1-6'). Gyldendal & Society for Danish Language and Literature.

Guarino, N. & C. Welty. 2000. 'Ontological Analysis of Taxonomic Relationships', in: A. Laender & V. Storey (eds.*) Proceedings of ER-2000. The International Conference of Conceptual Modeling*. Springer Verlag.

Guarino, N. & C. Welty. 2002. 'Identity and Subsumption'. In: R. Green, C.A. Bean, & S. H. Myaeng (eds.) *The Semantics of Relationsships: An Interdisciplinary Perspective, Information Science and Knowledge Management*. Springer Verlag.

Lenci, A., N.Bel, F.Busa, N.Calzolari, E.Gola, M.Monachini, A.Ogonowski, I.Peters, W.Peters, N.Ruimy, M.Villegas & A.Zampolli. 2000. 'SIMPLE – A General Framework for the Development of Multilingual Lexicons'. In: T. Fontenelle (ed.) *International Journal of Lexicography Vol 13*. 249-263. Oxford University Press.

Lorentzen, H. 2004. 'The Danish Dictionary at large: presentation, problems, and perspectives'. In: *Proceedings of the 11th EURALEX International Congress Vol. 1*. 285-294. Lorient, France.

Madsen, B.N., H.E. Thomsen & C. Vikner. 2004. 'Comparison of principles applying to domain-specific vs. general ontologies'. In: *Ontolex 2004 p. 90-95*. Lissabon.

Miller, G.A. 'Nouns in WordNet'. 1998. In: Fellbaum, C. (ed.) *WordNet – An Electronic Lexical Database p.23-47*, The MIT Press, Cambridge, Massachusetts, London, England.

Pedersen, B., Paggio, P. 2004. 'The Danish SIMPLE Lexicon and its Application in Content-based Querying'. In *Nordic Journal of Linguistics Vol 27:1 p.97-127*.

Pedersen, B.S., S. Nimb, N., J. Asmussen, N. Sørensen, L. Trap-Jensen, H. Lorentzen. 2006. 'DanNet - A WordNet for Danish'. *Proceedings from Third International Conference on Global Wordnets p.329-330*, Jeju, South Corea:. Pustejovsky, James 1995. The Generative Lexicon, Cambridge, MA. The MIT Press.

Ruus, H. *Danske kerneord*. 1995. Museum Tusculanums Forlag. Copenhagen.

Vossen, Piek (ed.). 1999. *EuroWordNet, A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, The Netherlands.

Wierzbicka, A. 1984. 'Apples are not a "kind of fruit"'. *American Ethologist 11:313-328*.

Using EuroWordNet for the Translation of Ontologies

Thierry Declerck[1], Asunción Gómez Pérez[2], Zeno Gantner[2], Ovidiu Vela[1], and David Manzano-Macho[2]

[1]DFKI GmbH, Language Technology Lab, Stuhlsatzenhausweg, 3,
66123 Saarbruecken, Germany
{declerck, vela}dfki.de

[2]UPM, Laboratorio de Inteligencia Artificial, Campus de Montegancedo,
28660 Boadilla del Monte, Spain
{asun, dmanzano, zeno}@fi.upm.es

Abstract

We describe the way we used EuroWordNet for providing ontologies, which are normally using concept labels in just one (natural) language, with multilingual facility in their design and use in the context of Semantic Web applications, supporting both multilingual semantic annotation and ontology extraction based on multilingual text sources.

1. Introduction

Ontologies in Semantic Web applications are used, among others, for providing semantic and content annotations of multilingual web pages. Therefore we dedicated work in the Esperonto project[1] for providing a strategy and a platform that supports the multilingual extension of ontologies existing in just one natural language, and in doing so to allow the semantic annotation of multilingual web documents using multilingual labels of ontologies.

In Esperonto we investigated the use of available multilingual language resources and basic natural language processing tools for providing for a supervised automated translation of labels in ontologies.

In this paper we describe the main type of multilingual lexical resources we have been considering in the Esperonto project: the lexical semantic approach, mainly the WordNet initiatives, and more especially the EuroWordNet (EWN) resources.

We describe then another type of multilingual information we are using for being able to translate labels of ontologies: the Wikipedia resource on the Web, which we use additionally to EuroWordNet. Wikipedia is not based on a lexical perspective but on a dictionary perspective that encodes knowledge of the world instead of knowledge of the words. We see thus in Wikipedia a real complementary multilingual resource to EWN and similar lexical semantic resources.

As a fallback position for the translation of ontologies, we use online general-purpose translation services

We then present the overall strategy for translating labels of ontologies and the actual state of the Esperonto implementation work on the platform for multilingual ontologies. And finally we sketch the evaluation strategy we have been thinking of, but which could not be implemented during the lifetime of the Esperonto project.

[1] Esperonto was a project of the Information Society Technologies (IST) Program for Research, Technology Development & Demonstration under the 5th Framework Program of the European Commission, with number IST-2001-34373. The project ran from 2002 to 2005.

2. Ontologies

(Domain) ontologies can be defined as a (possibly) complex data structure that introduces formal concepts and describes the relations existing between those concepts. The main goal of ontology is to formalize knowledge for ensuring a more compact description of it and a more efficient access to it. The concepts described by the (domain) ontologies do not have to rely on the words or terms in use in a particular natural language. But the praxis has been very often to label the concepts in using English terms.

A task of Esperonto was to provide for a platform that annotates natural language documents with knowledge encoded in ontologies. It might be here very useful to provide for a concept annotation in the language of the user. For achieving this, already available ontologies in one language (mostly English) should be "translated" into the language of the user. The term "language mapping" is probably more adequate here than the term "translation".

3. The Esperonto platform for supporting multilingual ontologies

The main functionality of the platform consists in loading (a subset of) ontologies and the selection of particular nodes/labels written in a particular natural language for translating them into another natural language. The actual graphical interface[2] of the tool supports the translation from English to German, Spanish or Catalan (the Catalan language is covered only by the Wikipedia resources and the on-line translation services), but the translation algorithm is suited for translations in any direction.

The user of the tool can do both things: modify and/or validate the proposed translations or write down her/his own translation. This supervision step is necessary since we cannot expect to have perfect translations of single terms that have a very specific meaning in ontologies, so that domain experts will always be involved to a certain extent in the translation process. In our case, the supervisor should ideally only have to chose and/or combine proposed translations proposed by the resources

[2] See the figures at the end of the paper

we include in the platform: EuroWordNet, Wikipedia and/or on-line translation services.

3.1. The general processing chain

The processing chain is the following:

1) If the concept label in the ontology is already available in the target language in our database, then just display it.

2) If this is not the case, then use first EuroWordNet (EWN) and check if the label is present in the WordNet of the source language (English in our case). If this is so, 2 cases are possible:

A. The label in the ontology is a multiple word unit (MWU): check if the multilingual index associated with the WordNet entry in the source language is pointing to an existing entry of the target language. Display the EuroWordNet entry of the target language if the matching is successful.

B. If this is not the case, check if the main words of the multi word unit are present in the EuroWordNet of the source and target languages (using again the multilingual index of EuroWordNet, which relates entries in the various languages). Display the results if the matching is successful. With "main words" we understand the words that are not to be considered as the so-called "stop words" (Determiners like 'the', prepositions like 'on' etc.). Main words belong in our case mostly to the class of nouns, but also to the class of adjectives.

3) If the EuroWordNet approach is not successful, use the same strategy described in 2) to the multilingual term resources of Wikipedia, which uses also an interlinking mechanism for relating entries in Wikipedia in the various languages available.

4) If 2) and 3) are not successful, use a fallback solution and access free accessible translation engines on the web and display their results, if any.

5) If no (satisfactory) result is displayed by the platform, the user can enter his/her own translation. In case satisfactory results are shown, the user can validate them, whereas the results can be edited for some improvement.

3.2. Some general considerations about the processing chain

As the reader can see, we give priority to the EuroWordNet resource. This is due to the fact that the EuroWordNet resources are organised in such a way that we expect a high quality in the resulting "translation" of a concept, since the multilingual index associated with a term in EuroWordNet has been built following semantic considerations and validated by language and/or domain experts.

EWN also offer glosses (in English) that give a definition to the terms listed in EWN. Those glosses can provide help when mapping a label in the ontology to EWN, if for example the glosses contain terms that are also appearing in the ontology.

The use of EWN turns out to be difficult when there is more than one possible entry in EWN that can be referred to from the label in the ontology (ambiguity problem). We are investigating here two approaches for using the glosses, a rule-based one and a statistical approach.

The rule-based strategy is twofold: 1) if in the glosses of the EWN terms of the target language, terms are occurring that are also present in the ontology to be translated, then the EWN entry having this gloss is a better candidate for the translation as the EWN entry in which gloss no such terms are occurring, and only the preferred translation will be displayed; 2) if the source and target EWN entries share the same or similar glosses (string matching), then the corresponding entry of the target language will be selected, discarding entries of the target languages that have distinct glosses as the entry of the source language.

The statistic approach is based on two gloss-based similarity measures in the Perl package WordNet::Similarity. This package implements two algorithms, called "The adapted Lesk" [13] and the "Vector" [14] algorithm. We went for a first evaluation of those algorithms, and can report that the Lesk algorithm performs better than the Vector one. But even for the Lesk algorithm we suggest a hybrid approach combining the Lesk algorithm with the rule-based approach. It seems that an implementation of this hybrid approach can offer a good solution.

But in any case, one has also to be aware that the EWN resource is far for being exhaustive and having an equal coverage for the different languages involved. Also not all the language specific WordNets do make use of the glosses with the same strength. So in our case, compared to the English WordNet, the German WordNet has not a large coverage in term of entries, whereas the Spanish WordNet is poorly "decorated" with glosses.

In the second place of the processing chain, we search the Wikipedia domain. Wikipedia is a Web-based multilingual dictionary resource developing quite fast and being currently extended to many languages. Wikipedia gives us an encyclopaedic view on the terms used in the ontology rather than the lexical semantic view of EuroWordNet. The definition article associated with the terms in Wikipedia can be considered as similar to the glosses in EWN, but are larger and more difficult to be processed for supporting the translation task. An advantage in using Wikipedia for supporting ontology translation is that the user can go to the Wikipedia articles and really check that the content associated with a term is the one he/she wants to have in the target ontology label.

In the actual implementation already some use is made of the structural organisation of the ontology. So the translation of terms is passed down in the taxonomy. Another use of the structural hierarchy consists in using it for guiding the translation process. Here an example for clarifying: consider the label "book" as a subtype of the label "publication". Knowing that the word "publication" is a substantive (it is encoded like this in the English EWN), the system can then filter out the verbal readings of the word "book" (in the case of booking a travel for example), and so not display to the user the Spanish verb "reservar" but only the nominal Spanish entries, like "libro"[3]. Also if the entry in the ontology contains the "to"

[3] Here we have to mention that EWN lists three types of word categories: Verbs, Nouns and Adjectives. An EWN entry can be part of more than one category, so the example of "book" that can be a verb and a noun. The ambiguity problem here is of purely syntactic nature. There are also semantic ambiguities, which are more difficult to cope with in our case.

prefix, our system is then choosing the verbal interpretation, applying simple syntax rules.

3.3. Some linguistic issues with EuroWordNet

There are some problems related to EuroWordNet (and partially to Wikipedia): all the terms are listed using the ground form of the words. So translating for example the English sequence "technical documentation" into Spanish, the following will be actually delivered by our system (using EWN) to the user: "tecnico" and "documentacion". Two words are given, since the multi word unit "technical documentation" is not in EWN, but each word alone is covered by EWN. We have two linguistic problems here, due to the word-by-word EWN based translation:

1) The word "tecnico" is the masculine form of this adjective. But the substantive "documentacion" bears feminine gender in Spanish. So the system has to generate the form "tecnica". This problem has been solved in Esperonto, automatically adding to the EWN data for Spanish (and for German as well) a (morphological) rule that generates the feminine gender of the adjectives in the case it is associated with a noun bearing the feminine gender (in German we also have to consider the neutral gender). Alternatively we can augment the EWN database with all the morphological forms that can occur in German and Spanish. We think that the rule-based approach is to be preferred, since it does not modify the EWN structure.

2) The second problem concerns the word order: the word-by-word translation of "technical documentation" is "tecnico documentacion". Once we have generated the right feminine form for the word "tecnico", we still have to provide for the right word order in Spanish, which is "documentacion tecnica". Here again a rule-based approach has been defined, applying to the proposed translation by EWN. In case this approach is failing, the user has still the possibility in the GUI to re-arrange the order of the translated words.

So at least two linguistic "interventions" are needed for solving this problem: provide for the right morphological forms of the translated words, and for the right word order. Formally the rules look like (whereas we subsume both Adjectives and Articles under the category "Modifier"):

a) If Gender(Head-Noun of EWN translated term) eq FEM => generate FEM-Form(Modifier of EWN translated term)
b) If Gender(Head-Noun of EWN translated term) eq NEUT => generate NEUT-Form(Modifier of EWN translated term)

These rules are meant to deal with the morphological properties of the terms (for Spanish and German). But the rule is not applicable to all Spanish adjectives, and therefore we constructed a list of the adjectives for which this rule does not apply. Dealing with the word order problem (relevant only for Spanish, since German and English have the same word order within nominal phrases):

a) If Sequence(translated terms) eq Adj-Noun => generate_sequence(Noun-Adj)
b) If Sequence(translated terms) eq Noun(1)-Noun(2) => generate_sequence(Noun-Prep-Noun)

The case a) is dealing with the improvement of the word-by-word translation of "technical documentation" -> documentacion tecnica". The b) case is dealing with the word-by-word translation of "message receiver" -> "recipiente del messaje".

But another linguistic "intervention" might also be very useful: parsing the glosses (in EWN) and definitions (in Wikipedia), in order to give to those a linguistic structure, which is more appropriate for detecting relevant expressions that can help the translation process of the ontology. So the platform for multilingual ontologies will be extended in order to search into linguistically annotated glosses and definitions, instead of pure text.

3.4. Linguistic issues with Wikipedia

Wikipedia is using only full form words. But in the Wikipedia "family" there is also now an open dictionary, which displays the ground forms of the word. An example is given in the following URL: http://open-dictionary.com/Arts, where the ground forms of the word "arts" are given in many languages. And quite interesting: the Wiki dictionary also links to the WordNet definition! So that we can close here a circle between the word based semantic net (WordNet) and the encyclopedic based semantic network. Here we still have implementation work to extract the morphological forms from the Wiki Dictionary and the links between Wikipedia terms and EuroWordNet terms.

4. Evaluation

We have been thinking about a first evaluation scenario that allows statements about the added value of the Esperonto platform for supporting multilingualism in ontologies. We will have to show that the use of a combination of language resources, as proposed in Esperonto, allows a higher degree of automation in the translation process of ontologies and a better quality of proposed translations submitted to the domain expert, as for example using only online translation services. The first evaluation will be something like defining a continuous line of using only:

1) EWN,
2) EWN+Wikipedia,
3) EWN+Wikipedia+SCHUG (for the analysis of Glosses and Definitions)
4) …

We should then be able to say how many words/terms can be translated without an active intervention of the domain expert, so that he/she cam just validate results of the translation process.

We will also use already available multilingual ontologies as test material for comparing the translation provided by the Esperonto platform and provide for measures in term of recall and precision.

We will also compare the results of the Esperonto platform with the output of the online translation services, whereas we will have to take in consideration the cases where either EWN/Wikipedia or the online translation services are not providing any results.

But we have to stress here the fact that only very few work has been done till now on the topic of evaluating ontology translation services, so that we enter here to a certain extent new land.

5. Conclusions

The actual state of the platform is offering choices for the translation of ontologies that is based on various type of information: lexical semantic (EWN), encyclopedic (Wikipedia) and usual translation services.

As the implementation of certain features that includes some linguistic processing and information is progressing, as well as the analysis of the whole ontology to be translated, we expect a higher degree of automation dealing with EWN and Wikipedia data that makes the platform a real alternative to sole translation services, since the platform is offering to a certain degree a knowledge driven translation that is supported by natural language analysis. The knowledge is the one accessed in EWN, Wikipedia and within the structure of the ontology being translated.

Some small qualitative and quantitative evaluations still have to be provided, comparing the results of the Esperonto ontology translation platform with already existing multilingual ontologies and with the output of translation services available on the web.

6. References

[1] http://www.globalwordnet.org
[2] http://www.esperonto.net
[3] http://www.cogsci.princeton.edu/~wn/w3wn.html
[4] http://www.hum.uva.nl/~ewn
[5] http://www.icsi.berkeley.edu/framenet
[6] http://www.ub.es/gilcub/SIMPLE/simple.html
[8] http://muchmore.dfki.de
[9] http://www.lsi.upc.es/~nlp/projectes/ewn.html
[10] http://nl.ijs.si/ME/
[11] http://en.wikipedia.org/wiki/Main_Page
[12] http://open-dictionary.com/
[13] S. Banerjee: Adapting the Lesk algorithm for word sense disambiguation to WordNet.Master Thesis, University of Minnesota, Duluth, 2002.
[14] S. Patwardhan: Incorporating dictionary and corpus information into a vector measure of semantic relatedness. Master Thesis, University of Minnesota, Duluth, 2003.

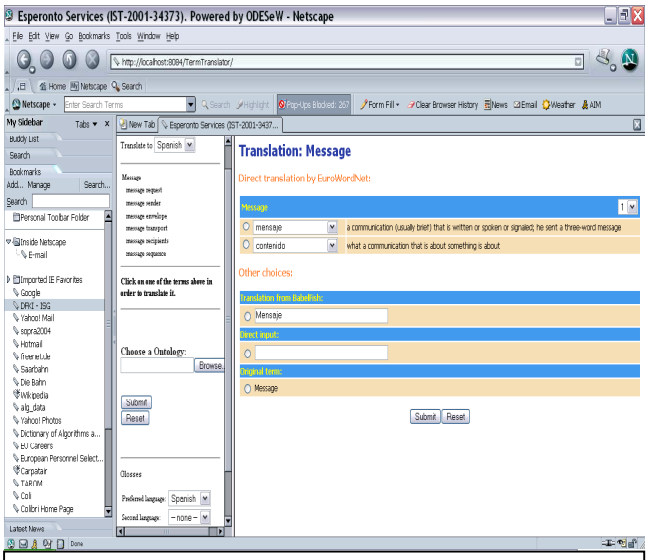


Figure 1: Spanish translations proposed by EWN

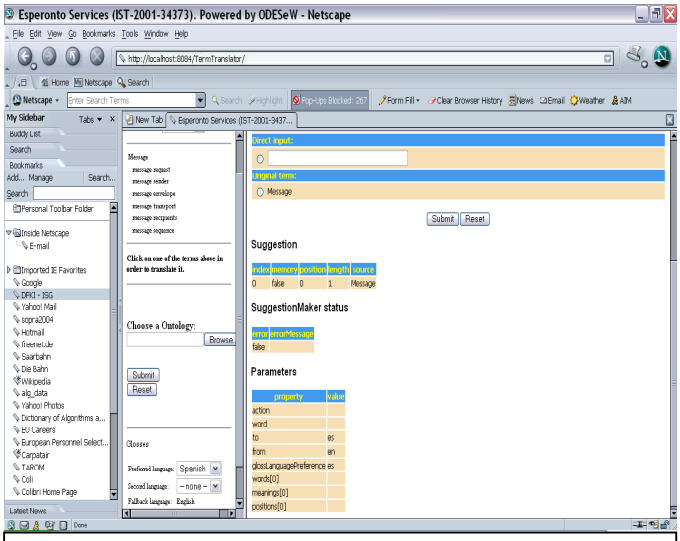


Figure 2: Debugging facilities of LabelTranslator

Linguistic Enrichment Of Ontologies: a methodological framework

Maria Teresa Pazienza, Armando Stellato

AI Research Group, DISP, University of Rome, Tor Vergata
Via del Politecnico 1 00133 ROMA (ITALY)
{pazienza,stellato}@info.uniroma2.it

Abstract

We introduce here a framework for adding Linguistic Expressivity to conceptual knowledge, as represented in ontologies. Both the multilingual aspects which characterize the (Semantic) Web and the demand for more easy-to-share forms of knowledge representation, being equally accessible by humans and machines, push in fact the need for a linguistically motivated approach to ontology development. Ontologies should thus express knowledge by associating formal content with explicative linguistic expressions, possibly in different languages. By adopting such an approach, the intended meaning of concepts and roles becomes more clearly expressed for humans, thus facilitating (among others) reuse of existing knowledge, while content mediation between autonomous agents gets far more chances than otherwise.

1. Introduction

The multilingual aspects which characterize the (Semantic) Web and the demand for more easy-to-share forms of knowledge representation, being equally accessible by humans and machines, depict a scenario where formal semantics must coexist side-by-side with natural language, all together contributing to the shareability of the content they describe.

These premises suggest that semantic web ontologies, delegated to express machine-readable information on the Web, should be enriched to both cover formally expressed conceptual knowledge and expose its content in a linguistically motivated fashion.

Even more could be done: revisiting ontology development process under this perspective, would in fact guarantee this scenario to become a suitable framework upon which even machine oriented task, like mediation and discovery, would benefit of this greater expressivity.

We introduced the expression "Linguistic Enrichment of Ontologies" to identify a series of different processes sharing the common objective of augmenting the linguistic expressivity of an ontology through the exploitation of existing Linguistic Resources (LRs, from now on). These processes strongly depend on the selected LRs but also on the task the ontology is dedicated to. In the following sections (sections 1.1-1.3) we describe some of the possible scenarios in which different enrichment tasks may be required, we then provide more information about one of these tasks (section 2) and will describe a framework (sections 3 and 4) for automatizing much of the work it requires. Finally (section 5), experimental evidence and quality of the suggested methods will be discussed.

1.1. Using a LR's semantic structure as a controlled vocabulary: semantic enrichment of ontologies

In this class of Linguistic Enrichment tasks, the semantic structure of a given LR (provided it has one), is used as a controlled vocabulary for the ontology and related application. What is required is just identification of *pointers* from ontological data to semantic elements of the linguistic resource. Access to pure linguistic information is then guaranteed by the links between the semantic and linguistic structure of the LR.

As a first example, consider an NLP ontology-based application, dedicated to whatsoever kind of text analysis task (e.g. Information Extraction), and which is strongly coupled with a semantic lexicon for extracting linguistic information from the text. The semantic pointers are needed to easily move from extracted, neutral, "linguistic information", which is processed in terms of lexical concepts, to "events" which are represented by the ontology.

As a further example, consider an agent society with knowledge mediators relying on a common form of knowledge. This common knowledge is represented by so called "upper ontologies", or "upper models" which contain a first stratification of general concepts. In a few cases (Beneventano et al., 2003), instead of an ontology, the semantic structure of an existing (WordNet: Fellbaum, 1998) linguistic resource has been adopted as a interlingua for guaranteeing communication between autonomous distributed agents.

1.2. Explicit Linguistic Enrichment

In case of no committed semantic agreement between autonomously developed information sources, no further solution exists for reaching semantic interoperability than relying on the very last form of *shared* knowledge representation: natural language. It is the form used by humans to pass from their own conceptualization of the world, to any form of shareable communication, being it spoken, written, or even related to formal representations of knowledge (also a good programming style ask for variables and functions being expressed through *evocative* labels). In-deed, stating direct links between ontological content (which is often scarcely modeled, upon a linguistic point of view) and linguistic expressions, may represent the only viable solution to increase the shareability of the represented knowledge.

Moreover, the improved range of expressions for denoting a concept and the (possible) presence of natural language descriptions for onto-logical data, facilitate reuse of existing knowledge, which is made more comprehensible also for humans.

1.3. Producing Multilingual Ontologies

Though English is commonly agreed to be a "lingua franca" all over the world, much effort must be (and is being) spent to preserve other idioms expressing different

cultures. Multilinguality has been cited as one of the six challenges for the Semantic Web (Benjamins et al., 2004). Exploitation of existing bilingual resources may thus help in the development of multilingual ontologies, in which different multilingual expressions coexist and share the same ontological knowledge. The multilingual enrichment process, mainly if considered upon already enriched ontologies, may beneficiate of a greater linguistic expressivity of the source data and thus exploit different techniques for obtaining proper translations for ontology concepts and roles.

2. Techniques for Semantic Linguistic Enrichment of Ontologies

In this work, we focus on the first of previously mentioned tasks: semantic enrichment of ontologies. This represents in fact a first necessary step through which all of the other tasks may be accomplished.

We thus designed a semantic enrichment process which can be run either semi-automatically, prompting ontology developers with suggestions to be supervised (approved, rejected or demanded), or executed as a totally automated procedure. These two options represent in fact desirable features for any application intended to support a linguistically motivated ontology development.

For our experimental setup, we adopted the terminology we defined in the Linguistic Watermark[1] (LW, from now on) framework (Pazienza & Stellato, 2006): a collection of interfaces for describing and manipulating linguistic resources. Through instantiation of these interfaces, ontology development applications may provide a uniform framework for accessing linguistic knowledge from different LRs, and use this content to enrich formal ontological data.

2.1. The Linguistic Enrichment Environment: adopted terminology

For sake of clarity, we will adopt from now on a terminology inherited from two well known standards for ontological and linguistic re-sources: OWL and WordNet.

OWL (Dean & Schreiber ed, 2004) has recently been accepted as a W3C recommendation for the representation of ontologies on the Web, so we have adopted its ontological model for our framework and will use its nomenclature for distinguishing ontological objects into *classes*, *properties* (*object properties* and *datatype properties*) and *individuals*. Frame based models for knowledge representation can equally be considered inside this framework, with *slots* taking the role of properties and *instances* acting as individuals of the OWL model. We adopt in fact the term *frame* to address any ontological object whose type needs not to be specified.

WordNet (Fellbaum, 1998) is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets (synsets), each representing one underlying lexical concept. Several wordnets exist for many other languages (Vossen, 1998; Stamou et al, 2002) which have thus favored a large diffusion of the model which inspired the original English version. As WordNet model closely matches the LW configuration which best fits for the semantic enrichment task, we will adopt terms like *synset* (or *lexical concept*, or *semantic element*), *sense* and *synonym*, under the meaning they assume in WordNet-like lexical databases.

We prefer in general to avoid use of term *concept* in any formal statement, as it is adopted in different communities with different meanings: a synset is a *lexical concept* in WordNet, while an OWL class implements a *concept* in Description Logics theory, furthermore, other ontology traditions use "concept" to mean every generic ontology construct, thus including properties and instances other than classes.

2.2. The Semantic Enrichment task

Objective of semantic enrichment task is to identify *semantic pointers* from ontological objects to semantic elements (e.g. synsets, for WordNet) of a linguistic resource.

Depending on their characterizing Watermark, not all LRs are exploitable for semantic enrichment of an ontology; in particular, only those resources whose model is compliant with the `ConceptualizedLR` (see Linguistic Watermark specifications) and at least one of `TaxonomicalLR` and `LRWithGlosses` interfaces, can be considered for this task.

Before detailing the model underlying our enrichment process, we describe a few empirical results we collected during our research. These results took the form of morphosyntactic and semantic evidences observed over several pairs of ontologies and linguistic resources.

All the reported examples refer to semantic enrichment of a DAML ontology[2] about baseball, downloaded from the DAML library of ontologies[3], using WordNet as a source for linguistic knowledge.

2.3. Taxonomy-Alignment evidences

In case the semantic structure of a given LR is organized as a taxonomy of broader/narrower linguistic concepts (the LR is a `TaxonomicalLR`), similarities between this taxonomy and that of the ontology may provide useful evidences for an enrichment task. The IS-A relation of ontologies (under the considered logic or frame based models) has well defined semantics, while taxonomical links of LRs may often bear informal and ambiguous relationships; nonetheless, an analysis of these similarities typically leads to interesting and reliable results.

The intuition behind this strategy is that if a semantic pointer links a frame-synset pair <*F*,*S*>, then other frame-

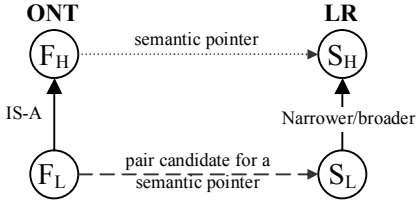


Figure 1: The sense-alignment square

[1] Linguistic Watermark is publicly available at: http://ai-nlp.info.uniroma2.it/software/LinguisticWatermark/

[2] http://www.daml.org/2001/08/baseball/baseball-ont for the original DAML version

[3] http://www.daml.org/ontologies/

synset pairs where the frame is more specific (more generic) that *F* and the synset is narrower (broader) than *S*, have a good probability of being linked through a semantic pointer. We call this phenomenon the "sense-alignment square".

In Figure 1, the semantic pointer between F_H and S_H already exists and represents an evidence for assessing a new semantic pointer over the pair $<F_L, S_L>$.

An example of this configuration is represented by the class labeled as *Hit* in the baseball ontology: this class has been eligible for 14 potential senses in WordNet. Of these 14 senses one is represented by the synset `noun.124696`, whose gloss states:

a successful stroke in an athletic contest (especially in baseball); "he came all the way around on Williams"hit"

This synset is more general than another Word-Net synset, `noun.39042`, which is described by the following gloss:

a base hit on which the batter stops safely at second base; "he hit a double to deep centerfield"

and which has among its synonyms the word "double". Finally, closing the alignment-square, *Double* is another class of the ontology, which is a subclass of *Hit*. Thanks to this evidence, both the *Hit*–`noun.124696` and the *Double*–`noun.39042` result as good candidates for being linked through a semantic pointer.

Analogously, a cross-link between a candidate pair and a semantic pointer represent a negative evidence for the candidate pair (Figure 2 below):

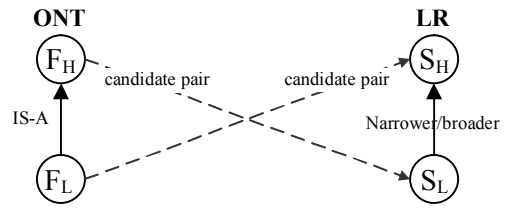


Figure 2: negative evidence for taxonomy-alignment

Though two taxonomical operators may present slight semantic differences, it is very unlikely for a configuration like this to exist so, in most of the cases, the candidate pair must not be connected through a semantic pointer (or the already existing semantic pointer should be verified).

The above examples represent situations involving a semantic pointer and a candidate frame-synset pair, however, in most of the cases, it will happen that there will be no direct cause-effect between an assessed pointer and a candidate pair. It is more frequent to face two (or chains of) candidate pairs each contributing to each other's plausibility: a proper model for representing evidences should take into account these mutual dependencies.

2.4. Evidences resulting upon analysis of glosses from the linguistic resource

Glosses offer natural language descriptions of concepts. Though their content is generally intended as an easy reference for human readability, it represents indeed a useful mean for discovering relations which have no explicit semantic counterpart in the resource they come from.

From the previous example, we can learn that a "double" is a kind of "base hit" (though the meaning of "hit" is not formally specified), even if the resource lacked of a taxonomical structure, binding the two concepts together in a broader/narrower relation.

A further example is represented by the class *Division*. WordNet offers 12 different senses for the term "Division". The gloss of the correct synset, noun.7741947, states:

a league ranked by quality; "he played baseball in class D for two years"; "Princeton is in the NCAA Division 1-AA".

Again, we could learn that a "division" is a "league", and League is one of the classes of the ontology. This case is however different from the previous one: in fact in the ontology tree, *Division* has not been conceived as a type of *League*. Nonetheless, a further analysis of ontological data reveals that Division appears in the restricted range of a property of class League. The co-occurrence of these two terms in the gloss, together with the presence of the range restriction binding the two classes labeled by the terms, suggests noun.7741947 as a potential candidate for Division.

There are however cases where a supposed interesting relation is not formally expressed in the ontology. An example is given by the class Out: we report here the gloss of its correct matching synset:

(baseball) a failure by a batter or runner to reach a base safely in baseball; "you only get 3 outs per inning".

we observe that "base" is a term appearing in the above gloss and that, at the same time, *Base* is a class in the ontology. Unfortunately, *Base* is not bound by any ontological relation to *Out*. Should this combination be discarded as a mere fortuity? May be not: the baseball ontology, with its 104 frames (considering classes and properties), may in fact be considered as a very domain-specific representation, where the sole presence of few concepts is enough to consider them semantically related in some way.

A final consideration: it may happen that glosses describing synsets which are candidate for enrichment of different ontology frames, contain common references to concepts of which no trace is present in the ontology. Oddly enough, the ontology about baseball which we used for our examples, contain no specific lexical nor conceptual reference to "baseball" itself! On the other hand, many WordNet definitions contain the word baseball in their glosses, so that, in those cases, it is quite easy for a human to immediately choose the right sense from the given set of candidates, just after a glimpse at the list of glosses. An automatic process should be able to discover even these "hidden" correlations and weight their effectiveness appropriately.

3. The Feature Model

To take into account all previous considerations, and to maintain a scalable approach towards new possible strategies and LW configurations, we adopted a probabilistic model based on a feature space which is produced upon the observed evidences.

We have thus defined a *Plausibility Matrix* M_P as a two-dimensional matrix on a $O \times L$ space, where *O* is the

cardinality of the ontological objects and L is the cardinality of the semantic data in the linguistic resource. Each element $M_P(i,j)$ of the matrix represents the plausibility that the ontological object i be matched with the lexical concept j.

Analogously, an Evidence Matrix M_E contains in each element $M_E(i,j)$ the set of evidences which contribute to the computation of element $M_P(i,j)$ in the Plausibility Matrix.

3.1. The Discovery Phase

The linguistic dimension in the two matrices is far broader than the ontological one. An efficient enrichment process should in fact consider a first *discovery* phase in which lexical anchors between the ontology and the LR are thrown. Each anchor represents a potential pointer from the ontology to the LR, and is discovered thanks to lexical similarity measures (use of string matching distances, possibly made smarter through knowledge of morphosyntactic properties of the natural language under analysis). In this phase it is important to drop as many anchors as possible, as they will represent the whole search space which is screened during the linguistic enrichment process. The trade-off is thus lightly biased towards *recall* rather than *precision*, as the latter, in this case, is only important for reducing the computational cost of the process. The result of the discovery phase is in fact a subspace L^A rep-resented by all synsets in L which have been anchored as potential targets for semantic pointers.

3.2. The semantic enrichment function

Once an L^A space has been extracted, we can then define the linguistic enrichment function f^{se} :

$$f^{se} : O \times L^A \mapsto [0..1] \tag{1}$$

This function maps pairs of elements from the ontology and the (restricted) linguistic resources into a confidence interval [0..1] representing the plausibility for assessing the presence of a se-mantic pointer between them.

The whole function f^{se} is realized through two main phases: by first the analysis of the linguistic and semantic similarities of the ontology and of the LR will lead the production of the *Evidence Matrix*; the *Plausibility Matrix*, based on the previously captured evidences, is then evaluated.

There may exist mutual dependencies between contributions of features (which we call dynamic) for different frame-synset pairs (as observed for taxonomy-alignment evidences and for some of the gloss-based evidences). For this reason, f^{se} is actually an iterative process $f^{se} = f^{se}(t)$; in particular computation of the plausibility matrix takes this general form:

$$M_P(t) = f\left(M_E, M_P(t-1), M_P(0)\right) \tag{2}$$

The Plausibility Matrix is thus not a single matrix, but a system which evolves over time, its content being the product of the observed evidences, of the system's history, and (possibly) of human intervention.

To adopt a smarter notation for addressing plausibilities of single frame-synset pairs, we define:

$$p(F,S,t) \stackrel{def}{=} M_P(F,S) \text{ with } M_P = M_P(t) \tag{3}$$

Finally, a *candidate pair* $<F,S>$ is a pair of elements $F \in O$ and $S \in L^A$, where $p(F,S,0) \neq 0$.

4. Instantiating f^{se}

The formulas in equations (1,2) are declarative forms representing classes of functions for realizing a semantic enrichment process, which are compatible with our model. In this section we present our realization of the semantic enrichment function, according to the two defined phases.

4.1. Computing plausibilities

In our experiments, we specified this function according to the following desiderata:

1. prizing candidate pairs characterized by positive evidences
2. punishing candidate pairs characterized by negative evidences
3. evaluate quantitative factors associated to different kind of evidences (representing the strength, or presence, of the evidence)
4. take into account inherent polysemy of every label associated to ontology concepts

The following equation has thus been conceived for computing elements of the Plausibility Matrix:

$$p(t) = \frac{p_0 + \left(1 - \prod_{i=1}^{n}\left(1-\rho(v_i,t)\right)\right)\cdot(1-p_0)}{1 + \left(1 - \prod_{i=1}^{m}\left(1-\rho(v_i,t)\right)\right)\cdot\left(\frac{1}{p_0} - 1\right)} \tag{4}$$

$p(t)$ is actually a smarter notation (to avoid abuse of indices) for $p(F, S, t)$, while $p_0 = p(0)$. p_0 value depends on τ_{high} and τ_{low}, two parameters representing the threshold over (resp. under) which a frame-synset pair must automatically be accepted (rejected), and on the ambiguity a (number of senses per word) of the term denoting F, according to the following formula:

$$p_0 \doteq \frac{\tau_{high} - \tau_{low}}{a} + \tau_{low} \tag{5}$$

For each evidence v_i, a weighted feature is then computed through the function $\rho(v_i,t)$, whose value depends on the type of evidence v_i and on the instantiation of its associated parameters. In the following section details are provided about how the structure of the different features v_i.

4.2. Extracting evidences

Following the experiences we summarized in section 3, we formalized methods for extracting interesting evidences and for mapping their con-tent into features for our f^{se} function.

First of all, we define the search space over ontological relations which is investigated for every class of evidences.

A *conceptual sphere* of a frame F over a set of relations R is a collection of frames linked to F through a relation $r \in R$. If r is a transitive relation, its closure may be limited to n allowed *hops*, depending on ontology's

size; *n* is called the *range* of the sphere wrt the *r* dimension.

The conceptual sphere for the Taxonomy-Alignment evidences has obviously been defined over the sole IS-A relationship, and its allowed range depends on the dimension of the ontology (for the average domain ontology, *n* is typically ∞, while its value must be restricted when dealing with very large – and deep – ontologies).

For gloss-based evidences we restricted the IS-A relation to cover only super concepts of the frame to be enriched; moreover, we considered both domain and range specifications of proper-ties, and range restrictions of properties for specific classes. Computation of the sphere also depends on the nature of the ontological object under analysis. In Figure 3 the algorithm for computing the conceptual sphere for classes, proper-ties and individuals has been shown.

4.2.1. Taxonomy-alignment evidences:

These kind of evidences assume the following form:

$$v \doteq \langle frame, synset, \mathrm{sgn} \rangle \tag{6}$$

where frame-synset is a *candidate pair* whose alignment influences the plausibility of the candidate pair which is being evaluated. The associated weighted features are computed through this formula:

$$\rho(v_i, t) \doteq \sigma_{TA} \cdot \mathrm{sgn} \cdot p(frame, synset, t-1) \tag{7}$$

where σ_{SA} is a coefficient related to this type of evidences and p(*frame*, *synset*, *t*-1) is the plausibility of the <*frame*, *synset*> pair at time *t*-1. sgn is 1 if *v* is a positive evidence, -1 if it is a negative one (as represented in figure 2, where $<F_H,S_L>$ and $<F_L,S_H>$ represent mutual negative influences, so that the plausibility of each pair is decreasing that of the other).

4.2.2. Gloss-mentioned Related Concepts:

The strategy for extracting these evidences is based on the intuition that the glosses of the candidate synsets which best define a given frame F, may contain linguistic references to other concepts contained in the conceptual sphere of F.

The extraction of this kind of evidences is de-scribed by the following algorithm:

```
for each Frame rc ∈ ConceptualSphere do
   MtchLvl ← match(rc, gloss),
   if MtchLvl ≠ 0
   Evidences ← Evidences ∪ evd(GR, rc, MtchLvl)
   end if
end for
```

where *Evidences* is the set of evidences related to a given <*F*,*S*> pair, *ConceptualSphere* is the conceptual sphere built around *F* and *gloss* is the gloss of *S*. `GR` is a tag denoting membership of the extracted evidences to this class of features. *MtchLvl* is a degree of lexical similarity between the term from the gloss and the label of the matching concept: this value is obtained on the basis

```
computeConceptualSphere(Frame frm, int DepthRange) SET OF Frame
input   frm: the class, property or individual which has been selected for linguistic enrichment
        DepthRange: the number of allowed hops along the IS-A relation for retrieving super concepts of frm
output  ConceptualSphere: the conceptual sphere surrounding frm
begin
FrameType type ← getOntoType(frm)
SET OF Frame ConceptualSphere ← {}
if (type = class  or type = property)
        ConceptualSphere ← ConceptualSphere ∪ getSuperConcepts(frm, DepthRange)
else    //obj is an instance
        Classes ← getClasses(frm)
        for each class ∈ Classes do
                ConceptualSphere ← ConceptualSphere ∪ {class} ∪ getSuperConcepts(class, DepthRange)
        end for
end if
if (type = class)
        for each property p, class c | frm.hasRestriction(p,c) or c.hasRestriction(p,frm)
                ConceptualSphere ← ConceptualSphere ∪ { c } ∪ { p }
if (type = instance)
        for each property p ∈ ( frm.getOwnRelationalProperties() ) do
                ConceptualSphere ← ConceptualSphere ∪ { p } ∪ frm.getOwnPropertyValues(p)
end if
if (type = property)
        for each class c ∈ ( domain(frm) ∪ range(frm) ) do
                ConceptualSphere ← ConceptualSphere ∪ {c}
end if
return ConceptualSphere
end
```

Figure 3: Algorithm for realizing the conceptual sphere for gloss-based evidences

of raw string matching distances and comparative morphological analysis of the two terms.

4.2.3. Gloss-mentioned Generic concepts:

Sometimes glosses of a candidate synset may disclose useful correlations between ontology concepts, which are unfortunately not captured by existing ontological relationships. In most cases nothing could be done and this phenomenon should simply be treated as a lack of information: the concepts can be recognized, upon human common sense, as potentially related (and they actually represent an evidence for a correct semantic pointer!), but they are not connected by any sort of relationship in the ontology (see related example in section 2.4)

Should the ontology be of modest size, offering a specification of a conceptualization of a very limited domain, it is nonetheless possible to consider each concept as somewhat related to the others. Under this hypothesis, given a <*F*, *S*> pair and a gloss *gloss* for synset *S*, this strategy considers as an evidence every occurrence of a term inside *gloss* which is also a label for a frame, even if no apparent relation with *F* exists.

```
for each term t ∈ gloss do
   Frame rc ← find(Ontology, t, MtchLvl),
   if rc≠ null
      Evidences ← Evidences ∪ evd(GG, rc, MtchLvl)
   end if
end for
```

Obviously, if both the previous strategies are applied, the results of the first one must be subtracted from those of the second one, which totally includes them. The second strategy is in fact less effective, on average, than the first one, and is generally used to augment the recall at the cost of a slightly minor precision. The evidences discovered by both strategies must thus be counted only on the first one, which has however a greater impact on the computation of the Plausibility Matrix

Both these two gloss-based features are defined by the following expression:

$$v \doteq \langle MatchingLevel \rangle \tag{8}$$

and their contribution to fse is:

$$\rho(v_i, t) \doteq \sigma_{GR/GG} \cdot MatchingLevel \tag{9}$$

4.2.4. Gloss-overlap between candidate synsets

Humans have the advantage of a wider knowledge about the world with respect to automatic processes. A user performing manual linguistic enrichment knows that the ontology is about baseball and therefore will probably check all the senses whose glosses report this term (see last example in section 2.4).

To reproduce such a behaviour, this strategy checks for possible term overlaps between glosses of synsets which appear as candidates for enriching concepts appearing each in the conceptual sphere of the other. Of course, overlapping terms must be properly filtered, to remove co-occurrences of articles, particles and very common words.

Instead of adopting large stop-lists, which may reveal to be incomplete, we exploit the whole set of glosses of the same resource which is used for linguistic enrichment, as a large corpus for statistically determining the distribution of terms. Thresholds may then be established for filtering very common terms which bear no informative evidence. Formally:

```
for each Frame rf_i ∈ ConceptualSphere do
   for each synset s_ij ∈ candidateSynsets(rf_i) do
      let rfgloss[i,j] ← s_j.getGloss()
   end for
   for each term t, t ∈ gloss and t ∈ rfgloss[i,j]
      let freq = LR.getGlossFrequency(t)
      if !filter(freq)
      Evidences ← Evidences ∪ evd(GO, rf_i, s_i, freq)
      end if
   end for
end for
```

As for taxonomy-alignment, even this third gloss-based strategy produces mutual influences among features: the collected evidences are in fact dependent upon the plausibility of candidate <*rc*, s_i> pairs. Their structure is in fact:

$$v \doteq \langle MatchingLevel, object, synset \rangle \tag{10}$$

and ρ assumes is computed this way:

$$\rho(v_i, t) \doteq \sigma_{GO} \cdot MatchingLevel \cdot p(object, synset, t-1) \tag{11}$$

MatchingLevel is in this case also dependant on the frequency of the observed overlapping term.

4.3. Frame-synset pairs as actors

SA and GO features (and, in general, *dynamic features*) form thus a network of mutual dependencies, where plausibilities of different candidate pairs depend on other pairs' plausibilities. Like in Conway's "Game of Life" (Berlekamp et al., 1982), correlated candidate pairs may associate into sort of "corporations" which tend to augment the *strength* (plausibility, in our case) of each of their members, thus lessening the chances of other candidates which, being cut away from these trust, are deemed to lose their run. At the same time, rare but not unusual "black sheeps" (represented by strong candidate pairs acting as bad evidences for others), may condemn whole sets of potential candidates to lose terrain in favour of others.

4.4. Reliability of gloss-based evidences

It emerges the risk, for gloss-based evidences, that they may be based on a mislead correlation of terms from glosses and labels for concepts, with the former indicating different meanings of those expressed by the related ontological concepts. Though sporadic occurrences of this phenomenon are indeed a possibility for each considered evidence type, their effects are generally cancelled out over large numbers of evidences, which, on average, present right correlations. In some cases the co-occurring terms bear in fact no polysemy at all, moreover, as a general consideration, several studies (Madhu and Lytle, 1965; Resnik, 1997) seem to support the hypothesis of a semantically conservative behavior of words wrt their use in a given specific context, so that even ambiguous

expressions tend to assume the same meaning if considered inside the same ontological framework.

5. Automatic Linguistic Enrichment: experimental results and final remarks

Fine tuning of evidence-typed σ-parameters has been performed over a collection of several small ontologies and/or portions of them. We then ran two experiments on two public domain ontologies, reporting performance in terms of standard precision & recall metrics.

We stress the fact that our framework foresees human effort both as a verification of automatic suggestions and as possible intervention on the enrichment process: a very few human decisions can in fact greatly affect the outcome of the automatic enrichment process, as they represent strong evidences (human choices are considered assessed semantic pointers, and have thus plausibility equal to 1) for correlation-based dynamic evidences.

Nonetheless, our experiments aim at evaluating the enrichment process also as a completely automatic procedure.

Recall has been measured towards the number of concepts which can be enriched with the considered LR. The linguistic resource thus determines the whole search space, and each evaluation of a linguistic enrichment process has only sense if considered wrt a specific LR. Regarding Precision, the "suggest and wait for confirm" threshold-based approach, which is well suited for a human centered process, has been given out for an immediate outcome of the highest ranking synset, chosen among all the candidate ones for every concept.

The first experiment has been performed on the baseball ontology chosen for our examples. The ontology, is composed of 78 classes, 26 properties and 13 individuals. Of these objects, 60 classes and 21 properties were considered for semantic enrichment (we performed the experiment limiting to the ontology schema, so we provide statistics only for classes and properties) during the discovery phase. The number of non ambiguous concepts (including both classes and properties) is 20 (~ 24,7% of the whole concept set) while the average ambiguity, (measured as the average polysemy of considered terms, wrt WordNet synset structure), was ~ 9,16. The oracle has been manually produced by two annotators which realized two documents reporting the most evocative synset for each concept. These documents have been compared and a final decision has been taken where discrepancies were found. The observed inter-annotator agreement has been however of 98.76% (one re-discussed decision out of the whole oracle).

The second experiment has been run on an ontology related to the university academic domain[4], developed in the context of the EU funded project MOSES (IST-2001-37244). This ontology has been built, in OWL language, over a preexisting DAML ontology[5] from the official DAML repository and finalized for representing the Italian university domain. As a consequence, while the original language in which concepts were expressed was English, many of the concepts added for describing the Italian academic institutions had only Italian labels. Though we plan for the future to define a two step enrichment process which is able to rely on multiple linguistic resources (for different languages) even for dealing with this kind of situations, we evaluated our algorithms over those parts of the ontology which were eligible for monolingual enrichment. More than half of the classes (100 out of 192) emerged during the discovery phase, while only a very small part of the properties (9 out of 100) have been discovered: this is probably due to the large amount of properties added during the customization to the Italian domain.

We report in the following table evaluation of the algorithm for both the experiments.

Ontology	Precision	Recall
Baseball Ont	80%	39,5%
Moses Italian	81,48%	42,72%

Table 1: Evaluation of linguistic enrichment over two publicly available ontologies

Detailed analysis of the test data on the first experiment revealed that, though only 40% of the original corpus (ontology) has been correctly enriched, another 50% contains the right choice as the second or third ranked suggestion. A similar observation holds for precision, where the remaining 20% wrong suggestions gave only some percentage points over the correct ones.

This reveals to be in line with the intended nature of the task, which is to be seen as part of a computer-aided, linguistically motivated approach to ontology development, more than a mere disambiguation problem.

6. Acknowledgements

This work has been partially funded under the GALILEO Cuspis Project (GJU/05/2412/CTR/CUSPIS) started inside the GALILEO Joint Undertaking User Segment, Call 1A: User Community/GNSS for Special Users Community, under the 6th Framework Programme of European Commission.

7. References

Beneventano D., Bergamaschi S., Guerra, F., Vincini, M: Building an integrated Ontology within SEWASIE system. In proceedings of the First In-ternational Workshop on Semantic Web and Data-bases (SWDB), Co-located with VLDB 2003 Ber-lin, Germany, September 7-8, 2003

Benjamins, V. R., Contreras, J., Corcho, O., and Gómez-Pérez, A.: Six Challenges for the Se-mantic Web. SIGSEMIS Bulletin, April 2004.

Berlekamp, E., Conway, J., and Gut, R.: The game of Life, *Winning Ways for your Mathematical Plays*, vol. 2, Academic Press, 1982, pp. 817-849

Dean, M. and Schreiber, G. editors: OWL Web On-tology Language Guide. 2004. W3C Recom-mendation (10 February 2004).

Fellbaum, C: WordNet - An electronic lexical da-tabase. MIT Press, (1998).

Madhu, Swaminathan and Dean Lytle, 1965. A figure of merit technique for the resolution of non-grammatical ambiguity. *Mechanical Translation*, 8(2):9–13

Pazienza, M.T. and Stellato, A.: Linguistic Enrichment of Ontologies: a methodological framework. *Second*

[4] http://www.mondeca.com/owl/moses/ita.owl

[5] www.cs.umd.edu/projects/plus/DAML/onts/univ1.0.daml

Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex2006), held jointly with LREC2006, Magazzini del Cotone Conference Center, Genoa, Italy, 24-26 May 2006

Resnik, P., 1997. Selectional preference and sense disambiguation. *In Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How*?, Washington, April 4-5, 1997

Stamou S., Oflazer K., Pala K., Christoudoulakis D., Cristea D., Tufiş D., Koeva S., Totkov G., Dutoit D., Grigoriadou M. (2002). BALKANET: A Mul-tilingual Semantic Network for the Balkan Languages. Proceedings of the International Wordnet Conference, January 21-25, Mysore, In-dia, 12-14.

Vossen. P: EuroWordNet: A Multilingual Data-base with Lexical Semantic Networks, Kluwer Academic Publishers, Dordrecht, 1998

LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies

Paul Buitelaar♦, Thierry Declerck♦, Anette Frank♦, Stefania Racioppa♦, Malte Kiesel♦, Michael Sintek♦, Ralf Engel♦, Massimo Romanelli♦, Daniel Sonntag♦, Berenike Loos♣, Vanessa Micelli♣, Robert Porzel♣, Philipp Cimiano∗

♦DFKI GmbH. Kaiserslautern/Saarbrücken, Germany
♣European Media Lab, Heidelberg, Germany
∗AIFB, University of Karlsruhe, Karlsruhe, Germany

paulb@dfki.de (contact address)

Abstract

To allow for a direct connection of this linguistic information for terms with corresponding classes and properties in a domain ontology, we developed a lexicon model (LingInfo) that enables the definition of LingInfo instances (each of which represents a term) for each class or property. The LingInfo model is represented by use of a meta-class, which allows for the representation of LingInfo instances with each class, where each LingInfo instance represents the linguistic features of a term for a particular class. Applications of the LingInfo model are in information extraction, dialogue analysis, and knowledge acquisition from text, i.e. in knowledge base generation and ontology learning.

1. LingInfo: Motivation and Design

To allow for automatic multilingual knowledge markup a richer representation is needed of the features of linguistic expressions (such as domain terms, their synonyms and multilingual variants) for ontology classes and properties. Currently, such information is mostly missing or represented in impoverished ways, leaving the semantic information in an ontology without a grounding to the human cognitive and linguistic domain.

Linguistic information for terms that express ontology classes and/or properties consists of lexical and context features[1], such as:

- *language-ID*: ISO-based unique identifier for the language of each term
- *part-of-speech*: representation of the part of speech of the head of the term
- *morphological and syntactic decomposition*: representation of the morphological and syntactic structure (segments, head, modifiers) of a term
- *statistical and/or grammatical context model*: representation of the linguistic context of a term in the form of N-grams, grammar rules or otherwise

To allow for a direct connection of this linguistic information for terms with corresponding classes and properties in the domain ontology, we developed a lexicon model (LingInfo) that enables the definition of LingInfo instances (each of which represents a term) for each class or property. The LingInfo model is represented by use of a meta-class `(ClassWithLingInfo)` and meta-property `(PropertyWithLingInfo)`, which allow for the representation of LingInfo instances with each class, where each LingInfo instance represents the linguistic features (`feat:lingInfo`) of a term for a particular class.

Figure 1 shows an overview of the model with example domain ontology classes and associated LingInfo instances. The domain ontology consists of the class `o:FootballPlayer` with subclasses `o:Defender` and `o:Midfielder`, each of which are instances of the meta-class `feat:ClassWithLingInfo` with the property `feat:lingInfo`.

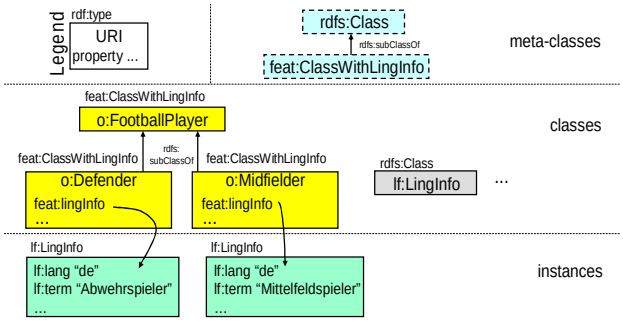


Figure 1: LingInfo model with example domain ontology classes and LingInfo instances (simplified)

Figure 2 shows a sample application of the model with a LingInfo instance (and connected 'stem', 'root' and other instances – for details see the complete LingInfo model in the appendix) that represents the decomposition of the German linguistic expression "Fußballspielers" ("of the football player"). The example shows `inst1` that represents the inflected (genitive) word form with stem "Fußballspieler" (`inst2`, "footballplayer"), which can be decomposed into "Fußball" (`inst3` , "football" with

[1] Morphosyntactic and syntactic features may be based in future versions on the (ISO-TC37/SC4-MAF and ISOTC37/SC4-SynAF) specifications. See also related documentation at the LIRICS project web site: http://lirics.loria.fr/documents.html

`semantics` "o:BallObject") and "Spieler" (`inst8` , "player), recursively continued for "Fußball" with "Fuß" and "Ball" (`inst5` and `inst7`, "foot" and "ball").

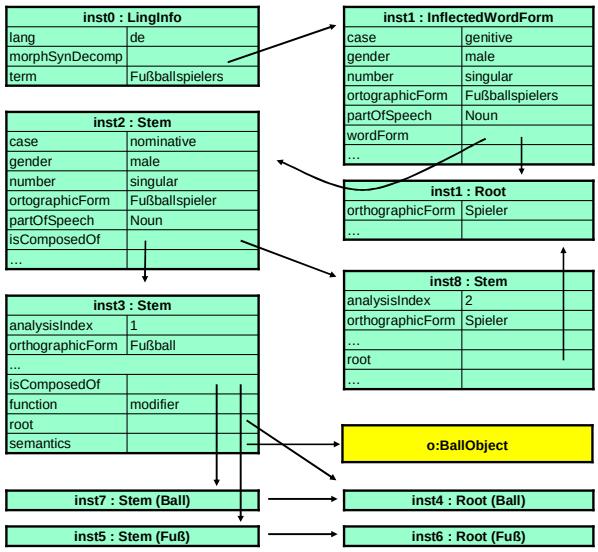


Figure 2: LingInfo instance (partial) for the morphosyntactic decomposition of "Fußballspielers"

2. Comparison with Related Work

2.1 Simple Knowledge Organization Systems

There is some overlap between the LingInfo model and the proposed SKOS[2] (Simple Knowledge Organization Systems) format for the formalized representation of thesauri. However, there is a technical and conceptual reason why SKOS does not fulfill the needs of our scenario[3].

On the technical side, SKOS uses sub-properties (`skos:prefLabel`, `skos:altLabel`) of `rdfs:label` together with `xml:lang` to attach multilingual terms to concepts. Furthermore, the RDFS specification[4] defines the range of `rdfs:label` to be `rdfs:Literal` and from the definition of `rds:subPropertyOf` follows that the range of `skos:prefLabel` and `skos:altLabel` is also (or a specialization of) `rdfs:Literal`. This is not sufficient in our scenario since we want to attach more linguistic information to classes than simple multilingual strings. This led us to the decision to use a meta-class `ClassWithFeats`, which allows us to attach complex information to classes with the properties `lingFeat` and `imgFeat`.

The conceptual problem we see with SKOS for the use in our scenario is that it mixes linguistic and semantic knowledge. SKOS uses `skos:broader` and `skos:narrower` to express "semantic" relations without clearly stating the semantics of these relations intentionally, and defines the sub-properties `skos:broaderGeneric`/`narrowerGeneric` to have class subsumption semantics (i.e., they inherit the `rdfs:subClassOf` semantics from RDFS).

Instead, the LingInfo model clearly keeps the linguistic and semantic, ontology-based knowledge representations separate: the ontology is represented using the semantic relations defined in RDFS or OWL-Full[5] with linguistic knowledge attached to classes and properties.

2.2 Wordnets and OntoWordNet

Our approach in effect integrates a domain-specific multilingual Wordnet into the ontology, although the Wordnet model does not distinguish clearly between linguistic and semantic information (Miller et al., 1995). Alternative lexicon models that are more similar to our approach include (Bateman et al. 1995) and (Alexa et al. 2002), but these concentrate on the definition of a top ontology for lexicons instead of linguistic features for domain ontology classes and properties as in our case. This is also the main difference with the proposed OntoWordNet model (Gangemi et al., 2003), which aims at merging the foundational ontology DOLCE (Gangemi et al., 2002) with WordNet to provide the latter with a formal semantics.

2.3 Lexical Markup Framework

Closest to our work are some recent initiatives of the ISO TC37/SC4[6] working group on the management of language resources, which was established in 2002 and continues the work from previous standardization initiatives, like EAGLES[7] (Expert Advisory Group on Language Engineering Standards) for morphological and syntactic annotation and ISLE[8] (International Standards for Language Engineering) for the representation of lexicon entries.

In the various initiatives of ISO TC37/SC4 the focus is on the more abstract level of meta-annotation and of frameworks supporting the creation and the exchange of annotations, data structures and resources. An important part of this work consists of the definition of procedures for the creation and maintenance of data categories for the various annotation frameworks. Data categories are formalized representations of the most relevant linguistic concepts, such as 'part of speech', 'lemma', etc.

The ISO TC37/SC4 standardization initiative that is most closely related to the LingInfo model is LMF, the Lexical Markup Framework, 'a common standardized framework for the construction of NLP lexicons' (Francopoulo et al. 2006). However, the main difference between LMF and the LingInfo model is again the level of division between linguistic and semantic knowledge. In LMF these are integrated in the same model by way of a lexical semantics slot, whereas in the LingInfo model all lexical semantics is to be found in the domain ontology - that is outside of the lexicon model per se.

As a further consequence of this approach, the LingInfo model allows also for the representation of non-linguistic, i.e. multimedia features (Buitelaar et al., 2005).

[2] http://www.w3.org/TR/swbp-skos-core-guide/

[3] In fact, the argumentation applies to all approaches based on `rdfs:label` and `xml:lang` for attaching multilingual labels to classes and properties.

[4] http://www.w3.org/TR/rdf-mt/

[5] OWL-Lite and OLW-DL do not support meta-classes and meta-properties (see http://www.w3.org/TR/owl-features/)

[6] http://www.tc37sc4.org

[7] http://www.ilc.cnr.it/EAGLES96/home.html

[8] http://www.ilc.cnr.it/EAGLES96/isle/

3. LingInfo in Context

3.1. The SmartWeb Project

The LingInfo model is developed and used within the SmartWeb[9] project on intelligent mobile information services for various domains, with a focus on soccer and the World Cup 2006 in particular. SmartWeb integrates question answering and ontology-based information extraction within a multimodal dialog system for a wide range of mobile devices. Information access to topical information available on the web is improved by adding machine-understandable semantics using a variety of techniques that range from semi- to fully automatic linguistic and semantic tagging to data-driven ontology learning.

LingInfo constitutes an ontology and linguistic knowledge base that provides for all other ontologies used in SmartWeb linguistic information (orthographic realizations, grammatical gender, stem and inflection) on ontology classes and properties for languages that are relevant to the SmartWeb scenario, i.e. German and English (and into some respect also French).

3.2. The SWIntO Ontology

A central component of the SmartWeb system is the integrated SWIntO ontology (Oberle et al., to appear), which consists of three layers: the upper model DOLCE (Gangemi et al., 2002), the domain-independent model SUMO (Niles and Pease, 2001) and several domain ontologies:

- **SportEvents** – As the soccer world cup 2006 will be the main application scenario, corresponding knowledge is modeled in the SportEvents ontology.
- **Navigation** – The SmartWeb user interfaces is based on mobile applications, e.g., by means of PDAs or by integration in cars or motorcycles. Navigation modeling is therefore a core requirement.
- **Discourse** - Multimodal web access is one of the core features of the SmartWeb sytem. It is therefore necessary to model user interaction in a generic way.
- **Multimedia** - The SmartWeb system will be able to display multimedia data such as live video streams. This data is described by means of an MPEG-based multimedia ontology.
- **LingInfo** – as described above

4. LingInfo Applications

The LingInfo model and instances are used in several components of the SmartWeb system, specifically of course in those components that are concerned with text analysis, i.e. in information extraction (IE) and dialogue analysis, and knowledge acquisition from text, i.e. in knowledge base generation and ontology learning.

4.1. Information Extraction from Text

The LingInfo model allows for the definition of *flexible interfaces* to linguistic processing components that ensure *consistency*. The SWIntO ontology, e.g., is interfaced with the IE system SProUT (Drozdzynski et al., 2004). Based on the information encoded in LingInfo, we automatically extract gazetteer entries for named entities, with back-references to the ontology. For terms associated with concepts, we recompile the relevant parts of the ontology, including LingInfo, into a type hierarchy used in the IE system. Thus, LingInfo information can be used to *consistently* identify and mark up (inflected) occurrences of domain-relevant terms.

The following example may illustrate this. It displays an excerpt of the SWIntO ontology that has been compiled into a type hierarchy defined in TDL[10], the representation language used by SProUT:

```
PlayerAction :< SportMatchAction.
SingleFootballPlayerAction :< PlayerAction.
FootballTeamAction :< PlayerAction.
GoalKeeperAction :< SingleFootballPlayerAction.
AnyPlayerAction :< SingleFootballPlayerAction.
```

Properties associated with these concepts are translated to TDL *attributes* of the corresponding *types*, e.g. the property `inMatch` of the SWIntO class `SportMatchAction` translates to the TDL attribute `INMATCH` that is inherited by all subtypes of the TDL type `SportMatchAction`. The SWIntO property `CommittedBy` that is defined for the SWIntO class `SingleFootballPlayerAction` translates to a corresponding TDL attribute `COMMITTEDBY` of the TDL type `SingleFootballPlayerAction`, and is again inherited by all its subtypes:

```
SportMatchAction := swinto_out &
     [INMATCH Football].
SingleFootballPlayerAction := swinto_out &
     [COMMITTEDBY FootballPlayer].
```

Multilingual (e.g. German) terms that are encoded as LingInfo instances are compiled into TDL lexical types:

```
"Teamaktion" :< FootballTeamAction.
"Spieleraktion" :< PlayerAction.
"Torwartaktion" :< GoalkeeperAction.
"Gesperrt" :< Banned.
```

SProUT extraction patterns can thus be triggered by lexical types, and define output structures that correspond directly to the classes and properties of the SWIntO ontology. For instance, the 'banned_player' rule below matches an extraction pattern for the SWIntO (SportEvents) class `BanEvent` with attributes `CommittedBy` and `InMatch` that is triggered for instance by the German LingInfo term "gesperrt".

Example sentences from the SmartWeb development corpus[11] to which this rule applies are as follows:

"… ist Petrow für die Partie gegen Schweden gesperrt."
("… has Petrow been banned for the match against Sweden")

"… ist David Trezeguet von der FIFA für zwei Spiele gesperrt worden."
("… has David Tezeguet been banned by FIFA for two matches")

[9] http://www.smartweb-projekt.de

[10] Type Description Language – see (Krieger and Schäfer 1994) for details

[11] See also http://www.dfki.de/sw-lt/olp2_dataset/

```
banned_player :>

@seek(player) & [IMPERSONATEDBY #player, INMATCHTEAM #team1]

(@seek(weekday_only) & [DOFW #dofw])? (token{0,2}
 @seek(soccer_institutions))?  token{0,3}
 @seek(game_teams) & [INTOURNAMENT #tour, TEAM2 #team2] morph & [STEM banned, SURFACE #event])

-> playeraction &
   [SPORTACTIONTYPE #event,
    COMMITTEDBY footballplayer &
            [IMPERSONATEDBY #player],
    INMATCH match &
            [INTOURNAMENT #tour, MATCHTYPE #match, TEAM1 #team1, TEAM2 #team2]].
```

4.2. Knowledge Base Generation

As described in (Buitelaar et al. 2006), the aim of the "SmartWeb Ontology-based Annotation" system (SOBA) is to automatically generate a soccer knowledge base, which is exploited in SmartWeb for knowledge-based question answering. The knowledge base is generated on the basis of information extraction with SProUT from freely available web documents on the soccer world cup – as described above. The web documents include structured as well as textual match reports and images with captions. All available text segments are linguistically annotated to extract semantic structures (class instances) that are compliant with the SWIntO ontology.

In extracting semantic structures, SOBA relies on the LingInfo model to avoid the creation of additional and redundant instances by comparing extracted names of players, countries etc. to LingInfo information of existing instances in the knowledge base.

4.3. Dialog Processing

The Smartweb dialogue integration framework (Reithinger and Sonntag 2005) integrates multiple natural language-intensive processing components such as SPIN (Engel 2005) for speech interpretation.

Usually, the rules for speech interpretation have to be written manually, but with the available LingInfo information we can generate part of the rules automatically. However, as the associated LingInfo information is not task-specific, the annotations are not always useful in a parsing context. To avoid an overgeneration of rules, so called generation rules allow a fine grained control over the rule generation. The generation rules have full access to the ontology and can exploit, e.g., the class hierarchy or the contained instances with LingInfo.

To resolve referential expressions, determiners (definite/indefinite) can be taken into account. This feature is provided by extending the LingInfo class with the property `RefProp`, which represents a definite/indefinite flag. A unit labeled as definite indicates the presence of an anaphoric reference which has to be resolved. This information is passed to FADE, which looks for the referenced item in recent user utterances, and resolves the reference. Additional syntactic information is used for disambiguation when several possible candidates for the referring expression exist.

4.4. Ontology Learning

In the ontology learning components of SmartWeb (Buitelaar et al., 2004; Schutz and Buitelaar, 2005), the representation of linguistic information for ontology classes and properties (relations) allows for the monitoring of any change in the domain model, for instance by tracking the use of soccer terms in subsequent versions of the SmartWeb development corpus.

The use of new terms or of new contexts for existing terms indicates an option for the extension or modification of the SWIntO ontology. For example, the term "Kneipe" ("pub") may be learned from a German text, as well as a potentially hyponymic relation with the term "Gebäude" ("building"). As the LingInfo information for the existing SWIntO class `Building` provides us with a corresponding LingInfo instance for the German term "Gebäude", this information can now be used to introduce a new class `Kneipe` (with a corresponding LingInfo instance for the German term "Kneipe") and integrate it into SWIntO as a subclass of `Building`.

4.5. Other Applications

Additional applications include the integration of the LingInfo model into ECtoloG (Micelli et al. 2006), an ontology that represents a formalization of construction grammar (Chang et al. 2002), and which allows only for one type of linguistic construction - i.e. pairings of form and meaning at different levels of abstraction. Since lexical constructions need linguistic information as provided by the LingInfo model, the LingInfo ontology was converted into OWL and integrated into ECtoloG. Therefore a meta-class `ClassWithLingInfo` (as sub-class of `owl:class`) was defined with the property `linginfo` that links ECtoloG classes and properties with LingInfo instances, enriching the ECtoloG classes with all necessary linguistic information as defined above.

An important challenge arising from this approach is that with the definition of a meta-class the ECtoloG ontology no longer confirms to OWL-DL but rather goes to OWL-Full, which thwarts the employment of Description Logic reasoners.

5. Lexical Acquisition for LingInfo

The LingInfo model enables *flexible interfaces*: by restricting the recompilation of LingInfo to core identifying properties (PoS, lemma, inflectional class), we can exploit a system's independent morphological

components, as in the case of SProUT, or we recompile the full range of information for systems that lack morphological processing components.

For this purpose, we are exploring different methodologies to (semi-) automatically instantiate a LingInfo model for a particular domain ontology with terms and corresponding linguistic information as described above. This is an incremental process, by which some information can be derived from annotated corpora. In this way, lexicons of tools used for annotation (e.g. Petitpierre and Russell 1995, Brants 2000, Lezius 2000) will be in effect tuned to respective domains and become fully integrated with the domain ontology.

Additionally, we can acquire syntactic information for domain-relevant terms from parsed domain corpora and/or existing syntactic lexica. The syntactic information can be defined in LingInfo, and exploited in information extraction tools. We are currently exploring the use of semantically annotated corpora, to acquire specific patterns between morphological and syntactic structures on the one hand and ontology classes on the other, based on the syntax-semantics links provided by LingInfo.

6. Current and Future Work

In current work, we are preparing the use of *deep parsing* to enhance the coverage and precision of concept recognition rules in the SProUT IE system, in particular for complex, non-local linguistic contexts that involve free word order, coordination, long distance constructions, etc. Via integration of argument structure information gained from deep parsing, SProUT recognition rules can refer to *deep syntactic* input structure, in particular, verbal arguments in non-local configurations. This will allow us to reliably identify concepts in linguistic constructions that are usually beyond the scope of shallow IE recognition systems. Our architecture for the integration of syntactic argument structure is designed as to permit integration of different parsers. The aim of future development in this area is the design of methods for semi-automatic acquisition of argument structure-based recognition rules, and the induction of argument-to-role mappings in the LingInfo model.

Other efforts are focused on the automatic enlargement of initial seed grammars in order to increase both their coverage as well as their inferential capabilities. For this a tight coupling to the Ontology Learning (described in Section 4.4) is vital to ensure consistency between the lexical semantics modeled via the grammar formalism and the descriptive conceptualization of the corresponding entities.

Further work is concerned also with pragmatic knowledge, which in a sense draws on all other knowledge sources cum contextual information. A first proposal on how to integrate such knowledge can be based on (Loos & Porzel 2004).

Acknowledgements

This research has been supported in part by the SmartWeb project, which is funded by the German Ministry of Education and Research under grant 01 IMD01 A. Thierry Declerck has been supported by the eContent project LIRICS[12] under EU grant 22236.

References

M. Alexa, B. Kreissig, M. Liepert, K. Reichenberger, L. Rostek, K. Rautmann, W. Scholze-Stubenrecht, S. Stoye *The Duden Ontology: an Integrated Representation of Lexical and Ontological Information* In: Proc. of the OntoLex Workshop at LREC, Spain, May 2002.

J. A. Bateman, R. Henschel and F. Rinaldi *Generalized Upper Model 2.0* Documentation Report of GMD / Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany, 1995.

T. Brants *TnT - A Statistical Part-of-Speech Tagger*. In: Proc. of 6th ANLP Conference, Seattle, 2000.

P. Buitelaar, M. Sintek and M. Kiesel *Feature Representation for Cross-Lingual, Cross-Media Semantic Web Applications* In: Proc. of the Workshop on Knowledge Markup and Semantic Annotation (SemAnnot2005) at ISWC05, Galway, Ireland, 2005.

P. Buitelaar, P. Cimiano, S. Racioppa and M. Siegel *Ontology-based Information Extraction with SOBA* In: Proc. of the International Conference on Language Resources and Evaluation (LREC), 2006.

N. Chang, J. Feldman, R. Porzel, and K. Sanders *Scaling Cognitive Linguistics: Formalisms for Language Understanding*. In: Proc. of the 1st International Workshop on Scalable Natural Language Understanding (ScaNaLU), Heidelberg, Germany, 2002.

W. Drozdzynski, H.-U. Krieger, J. Piskorski, U. Schäfer, F. Xu *Shallow Processing with Unification and Typed Feature Structures - Foundations and Applications.* In Künstliche Intelligenz, 1/2004.

R. Engel *Robust and Efficient Semantic Parsing of Free Word Order Languages in Spoken Dialogue Systems* In Proceedings of Interspeech 2005, Lisbon, Portugal, 2005

G. Francopoulo, M. George, N. Calzolari, M. Monachini, N. Bel, M. Pet, C. Soria *Lexical Markup Framework (LMF)* In: Proc. of the International Conference on Language Resources and Evaluation (LREC), 2006.

A. Gangemi, N. Guarino, C. Masolo, A. Oltramari and L. Schneider *Sweetening Ontologies with DOLCE.* In: Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW), Siguenza, Spain, pp. 166-181, 2002.

A. Gangemi, R. Navigli, P. Velardi *The OntoWordNet Project: extension and axiomatization of conceptual relations in WordNet.* In: Proceedings of ODBASE03, Springer, 2003.

H.-U. Krieger and U. Schafer *TDL---a type description language for constraint-based grammars* In Proceedings of the 15th International Conference on Computational Linguistics (COLING), pp. 893-899, 1994.

W. Lezius *Morphy - German Morphology, Part-of-Speech Tagging and Applications* In: Proc. of the 9th EURALEX International Congress, pp. 619-623, Stuttgart, Germany, 2000.

[12] http://lirics.loria.fr/

B. Loos and Porzel, R. *Towards Ontology-based Pragmatic Analysis*. In Proceedings of DIALOR 2005, June 9 - 11, Nancy, France, pp. 163-166.

V. Micelli, and R. Porzel *Tying the Knot: Ground Entities, Descriptions and Information Objects for Construction-based Information Extraction.* In: Proc. of OntoLex06. Genoa, Italy, 2006.

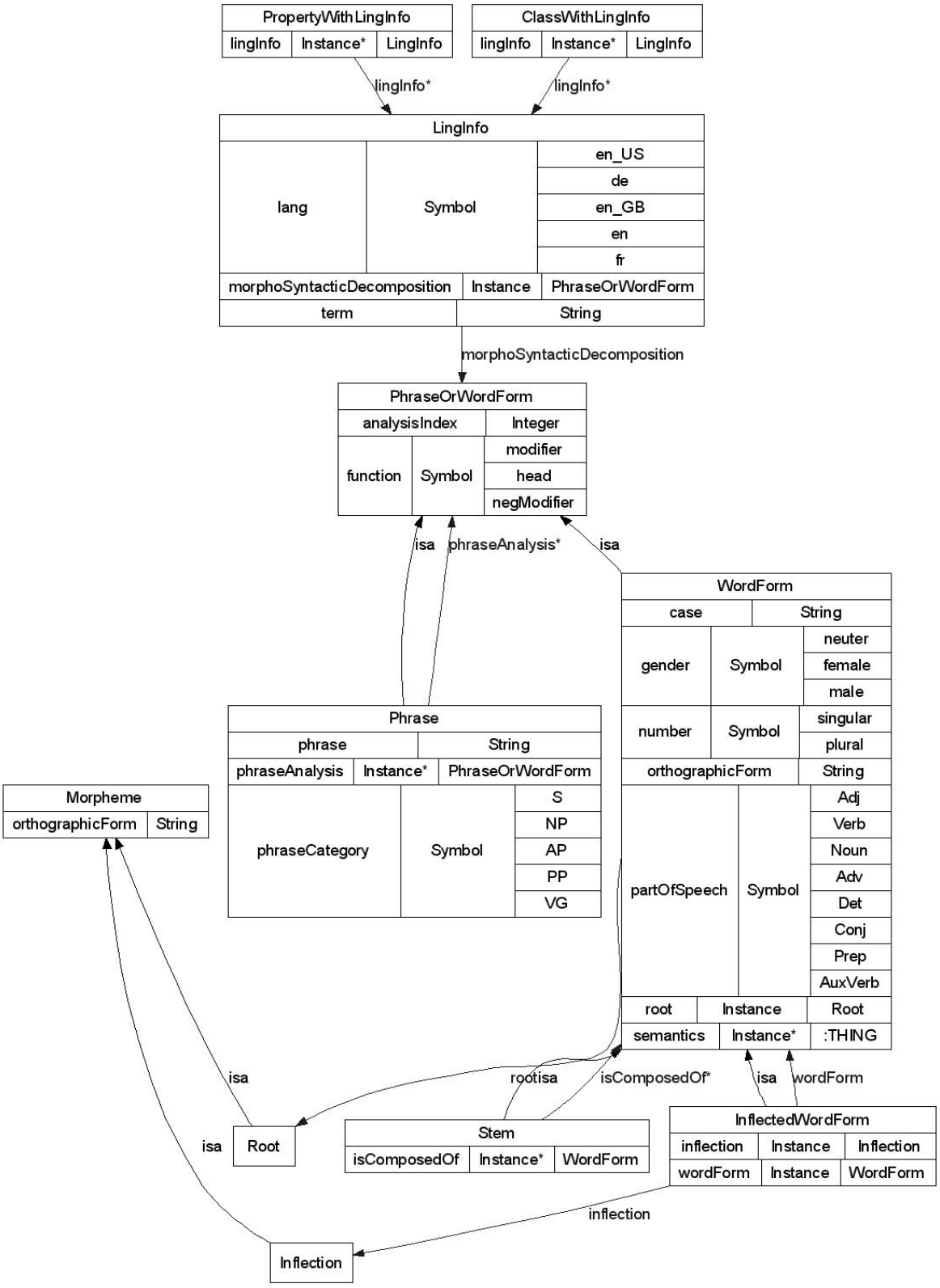
G. A. Miller *WORDNET: A Lexical Database for English*. Communications of ACM (11): 39-41, 1995.

I. Niles and A. Pease *Towards a standard upper ontology*. In: Proc. of the international conference on Formal Ontology in Information Systems (FOIS01), ACM Press, 2001.

D. Petitpierre and G. Russell *MMORPH - The Multext Morphology Program.* Multext deliverable report for task 2.3.1, ISSCO, University of Geneva. 1995.

N. Reithinger and D. Sonntag *An Integration Framework for a Mobile Multimodal Dialogue System Accessing the Semantic Web*. Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech) 2005.

Appendix: LingInfo Model



Tying the Knot:
Ground Entities, Descriptions and Information Objects for Construction-based Information Extraction

Robert Porzel, Vanessa Micelli, Hidir Aras and Hans-Peter Zorn
European Media Laboratory
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
{firstname.lastname@eml-d.villa-bosch.de}

Abstract

In this paper we present an approach to formalizing a construction grammar framework, called Embodied Construction Grammar and its integration into ground ontologies for the purpose of information extraction from natural language texts. For this we employ a common foundational ontology, i.e. the Descriptive Ontology for Linguistic and Cognitive Engineering, and two dedicated modules one called *Descriptions and Situations* and the other one *Ontology of Information Objects*. We will sketch out how to employ such models in agent-based semantic wrappers to extend their analyzing capabilities beyond structured via semi-structured sources to natural language data extracted from the web.

1. Introduction

One of the great challenges addressed in current research in the areas of information extraction, question answering and dialog systems, e.g. in the SmartWeb project (Reithinger et al., 2005), falls under the heading of *open-domain question answering* using the "web" as the corpus from which answers are extracted.

The approach taken in the SmartWeb project is to provide ubiquitous access to information on the web via multimodal user interfaces by combining knowledge-based and stochastic processing techniques to get the best of both worlds. In tune with the *Semantic Web* initiative (Berners-Lee, 2001) we seek to employ linguistic and ontological knowledge – wherever possible – and revert to statistical natural language processing (NLP) as well as standard information extraction (Cunningham, 2005) and question answering approaches (Moldovan et al., 2000) in the absence of formal and explicit knowledge. The SemanticWeb effort seeks to add machine-understandable semantics, i.e. to bring meaning to the internet making it possible for artificial agents to "understand" the information it contains, based on formal explicit models of certain domains of interest, i.e. formal ontologies (Gruber, 1993).

In this paper we take a closer look at an ensemble of ontology-based techniques to extract semantic information from structured, semi-structured and unstructured sources by means of so-called *semantic wrappers* (Arjona, 2002). Today structured and semi-structured content "hidden" inside HTML documents can be extracted reasonably well using increasingly automatic wrapper systems (Kushmerick et al., 1997; Simon and Lausen, 2005). A part of the extracted data records can be mapped to its semantic representation, e.g. RDF, resulting in instances of an underlying ontology. This, however, is not the case for unstructured elements such as natural texts, which are – if at all – further processed using shallow NLP methods. In any case, the semantically enriched data that has been extracted from the web can be stored in a semantic repository and then used for subsequent processing such as question answering as shown in Figure 1.

In the SmartWeb project we explore online and offline access to web data and analyze its structure and topicality. As mentioned above, while some success is achieved in extracting semantic information out of structured and semi-structured data, natural language texts, which notoriously contain non-literal, metaphorical context-dependent or otherwise underspecified linguistic expressions, still present a challenge for natural language understanding systems. What is, therefore, needed – especially for the natural language found on the web – as e.g. in news tickers or blogs, are models and processing technologies for linguistic knowledge that can deal with such "non-standard"[1] input.

2. Searching for Grammar Right

We decided to adopt construction grammar as it fulfills our demands on a grammar formalism: it is – in a sense – designed for robustness and partial analyses – which should be obligatory if one has to deal with real natural language data, as evidenced by the both "ungrammatical" and metaphoric title of this section. Construction grammar (Goldberg, 1995; Kay and Fillmore, 1999; Croft, 2001) is based on insights from cognitive and functional grammar (Fillmore, 1982;

[1] Ironically, it turns out that "non-standard" expressions are found quite more frequently than "standard" ones.

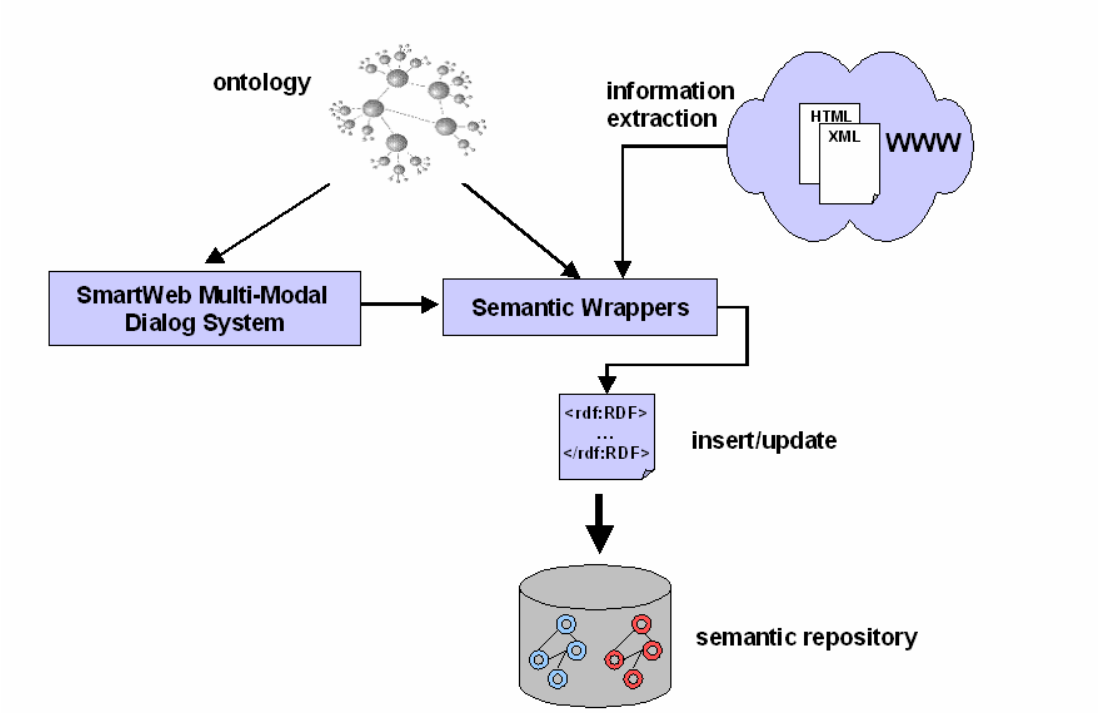


Figure1: Semantic translation of (semi-)structured data using semantic wrappers

Langacker, 1987; Lakoff, 1987) and eliminates the distinction between lexicon and syntax.
The only structure posited is that of a construction and is defined by Goldberg as follows: "C is a construction iffdef C is a form-meaning pair <Fi, Si> such that some aspect of Fi or some aspect of Si is not strictly predictable from C's component parts or from other previously established constructions." (1995:4).
That means that every level of a language can be mapped to corresponding constructions. Individual constructions can differ along three dimensions (Langacker, 2003) that are sketched out below:

- **Generality**: describes the extent to which the constructional schema is schematic rather than specific, e.g. highly specific constructions are holophrastic expression as found, for example, in early child language or idioms.

- **Productivity**: describes the extent to which a constructional schema is accessible for sanctioning new instances, e.g. so-called extensions via analogy or re-analyses.

- **Compositionality**: describes the extent to which the meaning and form of the whole are predictable from those of its parts in accordance with corresponding sanctioning schemas.

Especially in the light of language variation and change – and, therefore, robust and scalable natural language understanding – it is important to note that constructions can change their location over time or text in this three dimensional space in any direction. A language understanding system must consequently be based on a grammar that can reflect the topical status of the underlying constructions in this space. We, therefore, regard the construction grammar framework as the most suitable one for our purposes.

3. Formalization of Construction Grammar

Constructions, in general, are form and meaning pairings whereby a form can be any linguistic unit, from phonemes, morphemes to clauses, and a meaning is represented by a conceptual schema. In our attempt towards a formalization of construction grammar by means of formal ontologies we adhered to the basic ideas of embodied construction grammar (ECG) (Chang et al., 2002) — which presents a formal computational model of construction grammar, its main foci being on natural language understanding and later simulation processes (Bergen and Chang, 2002).
While basic morpho-syntactic knowledge is needed together with the corresponding ontological information, our research in this area seeks to capture and combine also semantic and pragmatic knowledge needed for deeper semantic analysis and inference of naturally occurring linguistic expressions.
The contribution of our approach hereby lies in analyzing unstructured data using a ground ontology which is linked to another ontological model of semantic and conceptual knowledge endowed with a construction grammar layer that links the two. Hereby, it is important to note the paramount importance of

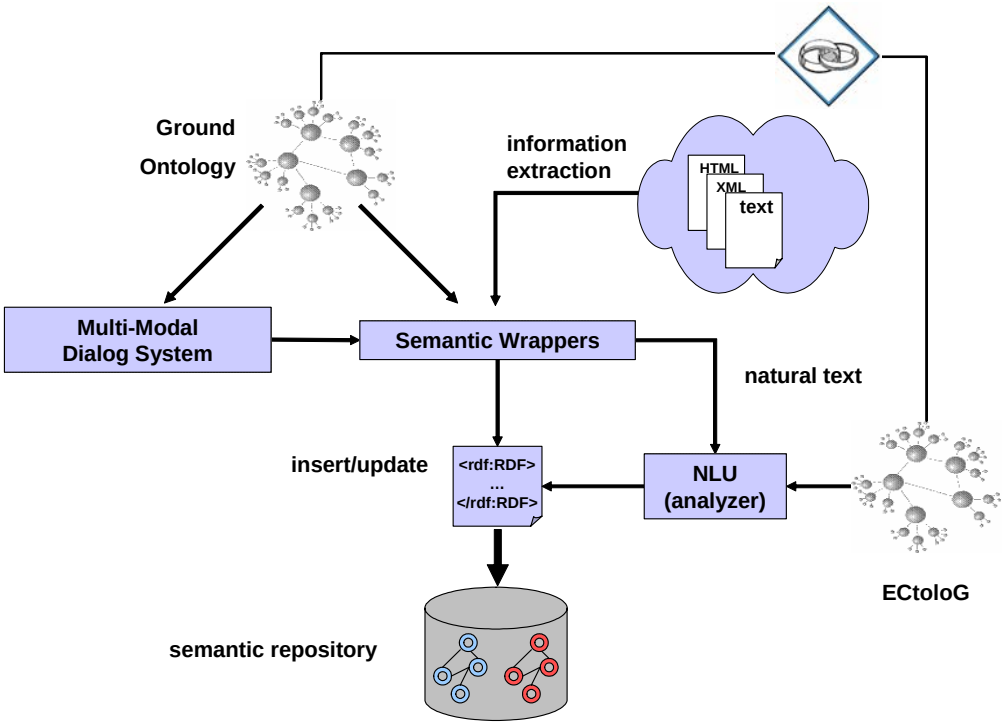


Figure 2: Natural text extraction and construction-based analysis.

using one and the same foundational ontological model. One of the central advantages of our ensuing ontological model (called *ECtoloG*) over the currently used ASCII-format of ECG lies in its compatibility with other ground ontologies developed within the Semantic Web framework.

We chose to integrate the construction grammar layer into a combination of two ontological modeling frameworks: one called *Descriptions & Situations* (D&S) (Gangemi and Mika, 2003) and the other one *Ontology of Information Objects* (OIO) (Guarino, 2006), which are extensions of the *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) (Masolo et al., 2003).

Gangemi and Mika (2003) regard D&S as an ontology for representing a variety of reified contexts and states of affairs and state that in contrast to physical objects or events, the extensions of ontologies by non-physical objects pose a challenge to the ontology engineer. The reason for this lies in the fact that non-physical objects are taken to have meaning only in combination with some other *ground* entity. Accordingly, their logical representation is generally set at the level of theories or models and not at the level of concepts or relations. It is, therefore, important to keep in mind that the meaning of a given linguistic expression emerges only through the combination of both linguistic and conceptual knowledge with so-called *encyclopaedic* ontological knowledge, as modeled in a descriptive ground ontology.

Therefore, it became possible for us to model conceptual schemas as *descriptions* which, in the D&S ontology, are described as social objects which represent a conceptualization (e.g. a mental object or state), hence they are generically dependent on some agent and communicable. Descriptions define or use concepts or figures and are expressed by an information object. Since descriptions are modeled to be expressible by information objects, e.g. constructions are modeled as those.

Information objects are defined by Guarino (2006) as social objects and are, therefore, realized by some entity. They are expressed according to some system for information encoding, e.g. formal expressions, linguistic, diagrammatic or iconic objects, or – as we propose herein – constructions. Consequently, they are regarded to be dependent on an encoding as well as on a concrete realization and can, as stated above, express a description (again defined as the ontological equivalent of a meaning or conceptualization). As described by Guarino (2006), information objects can be about any entity, and are interpretable by agents. From a communication perspective, an information object can play the role of "message" or, from a semiotic perspective; it plays the role of "expression".

Since constructions constitute form and meaning pairings, both poles need to be modeled in the ontology for every construction. In the remaining section we describe the modeling of lexical constructions. The form pole of each construction is modeled with help of the *realized-by* property. This property designates that a (physical) representation – as e.g. the orthographic form of the construction – realizes a non-physical object – in this case our construction. This property is also inherited by the class *information-object*, the superclass of constructions. What fills the range of that property is the class of *edns:physical-realization*. Therefore, we

define an instance of *inf:writing*, which then fills the form pole of the respective construction.
This instance has once more a relation which connects it to instances of the class *inf:word* which designate the realization of the instance of the *inf:writing* class.
This way of modeling the form pole of each lexical construction enables us to automatically populate our model with new instances of constructions, as will be described more detailed in section 4.
Analogous to the modeling of meaning in the original ECG the meaning pole is 'filled' with an instance of the class of *image schema*. This can be done with the help of the *edns:expresses* relation. This relation is defined, according to the specification of the D&S ontology, as a relation between information objects that are used as representations (signs) and their content, i.e. their meaning or conceptualization. In this ontology, content is reified as a description, which offered us the possibility to model image schemas as such, as already stated above.

4. Application

We already started to populate our ontology automatically with lexical constructions. The integration of the form side of these constructions and the consequent population of the ECtoloG presented little challenge. As future work the automatic extension process shall also include higher level constructions and the automatic learning of appropriate schemas, i.e. descriptions.
For the integration of the form side of new lexical constructions into ECtoloG texts initially need to be part-of-speech tagged. For this purpose, we are employing a tool that enables morphological analysis and synthesis. Therefore, we decided to make use of Morphy (Lezius, 2002) which complies with our demands. The analysis of the term *Fußballspielers*, e.g., yields the following information:

Fußballspieler SUB GEN SIN MAS
KMP Fuß/Ball/Spieler

This is to say that the stem of the term is *Fußballspieler*, its part-of-speech is noun, its case is genitive, its number singular, its grammatical gender masculine and the word is a composite of the three following stems *Fuß*, *Ball* and *Spieler*, denoted by the tag KMP. This information can then easily be integrated automatically into the ECtoloG. After tagging of whole sentences or texts, whole paradigms of nouns, adjectives and verbs are integrated into the ontology as instances of the class WORD.
When the agents come across unstructured natural language text, each phrase is extracted and passed to a constructional analyzer (Bryant, 2003). The analyzer performs semantic analysis by means of a semantic chunker and a chart, based on the constructions modeled via the formalism proposed herein. It analyzes semantics and syntax, finds constructions and schemas that represent the semantics of the sentence and yields to a so-called *semantic specification* of the sentence. This semantic specification is constituted by a co-indexed lattice of schema instances.
This specification is automatically transformed into RDF, using a production system called *2RDF (Zorn et al., forthcoming), so that it can be interpreted by the dialog system as sketched out in Figure 2.
This process involves the actual mapping and generation of additional instances to conform to the ontology axioms, instance merging and a validation step. While there are some more elaborate approaches for transforming semi-structured or even unstructured data to RDF, none of them fit our purpose, because they integrate either the wrapping or the parsing which we want to be separate. For example, the TARTAR system (Pivk et al., 2005) transforms arbitrary HTML tables into frame-structures (which can then easily be transformed to RDF). Unfortunately, TARTAR integrates the wrapping and analyzes of the semi-structured data with the RDF transformation. In a similar way the SPIN parser (Engel, 2002) implements a working memory based production system for parsing natural language directly to frames or RDF instances. Here the sources are natural language sentences where terminal symbols - words - are replaced by ontology concepts using also a production system approach.
As mentioned above the agent-based extraction system has access to several information sources (unstructured, semi-structured and structured ones) whose extraction results all need to be integrated semantically in a consistent fashion. Here, the ECtoloG system on the one hand and the ontology instance population via *2rdf on the other hand, need to do their work while staying semantically consistent with the ground ontology employed[2]. This is done by connecting the extraction entities or agents for each source and the NLU analyzer via one dedicated transformation entity.

5. The Model at Work

As already stated, there is no explicit separation between syntax and semantics in construction grammar. One of the most cited examples to demonstrate this necessity is Goldberg's (1995:29) example sentence:

(1) *he sneezed the napkin off the table.*

The whole meaning of this sentence cannot be gathered from the meanings of the discrete words. The direct object *the napkin* is not postulated by the verb *to sneeze*. This intransitive verb would have three arguments in a lexico-semantic theory: 'X causes Y to move Z by sneezing'. Goldberg states that the additional meaning of caused motion which is added to the conventional meaning of the verb *sneeze* is offered by the respective caused-motion construction.
In the same way our model enables the constructional analyzer described by Bryant (2004) to analyse a phrase such as:

[2] In this case we employ the SmartWeb Integrated Ontology (SWintO) as the ground ontology, which is also described in this volume by Buitelaar et al. (2006).

(2) *Ballack köpft das 1:0.*[3]

First of all the corresponding form sides of the lexical constructions for *Ballack*, *köpfen*, *das* and *1:0* and the needed flexional morphological constructions – that get us *köpft* from *köpfen* – need to be included. Next we need corresponding meaning sides of the constructions, which is straight-forward in the case of *Ballack*, who plays the functional role of a SoccerPlayer in our ontology, which, in turn, is impersonated by a NaturalPerson, and 1:0 that of a Score, which will become translated into a MatchResult in the ground ontology. More complicated is the verb *köpfen*, which introduces ambiguity, as it is also employed in German in the sense of *beheading*.

Here the fact that we use a parser with a chart comes into play. Both readings: a) that the "1:0" is the patient of a BeheadingSchema, which – via some steps – inherits parts of its semantics from a CausedMotionSchema, namely that which gets beheaded or b) that the "1:0" is the patient of a JointMotionSchema, i.e. the movement of Ballack's head causes another unspecified object (the ball) to move into another unspecified object (the goal), are found in the chart.

However, we find that the reading based on a ResultativeConstruction will be favoured due to the semantic clash of a Score being construed as the patient of a BeheadingSchema, which leaves that entity dangling and, therefore, creates a semantic specification which is less *dense*[4], than the semantic specification that employed the ResultativeConstruction to set up a scene with a JointMotionSchema.

6. Concluding Remarks

In this paper we have presented an approach to how to 'marry' a formalized construction grammar framework with ground ontologies using a common foundational ontology (DOLCE) and two dedicated modules (D&S and OIO). Additionally, we sketched out how to employ such models in agent-based semantic wrappers to extend their analyzing capabilities beyond structured via semi-structured sources to natural language data extracted from the web.

At the moment, the greatest problem we foresee is that of coverage, which we will touch upon below. Additionally, ways of finding agreement on both the constructional and schematic inventory constitute another obstacle on the way to a fully populated model. Here the debate in the linguistic research arena is still ongoing, in terms of finding systematic ways to determine the set of solidified constructions in any given language at a given time. Last but not least membership to any instance to a certain kind of construction, e.g. AdjectiveConstruction or DitransitiveConstruction, is not absolute but a matter of degree. This means that a lexical linguistic token (also called *construct* in most flavours of construction grammar) can be more or less adjectival or a clausal expression can be more or less ditransitive. For this, measures of entrenchment and collostructional strength (Stefanowitsch and Gries, 2003) need to be calculated based on the given corpora at hand.

Since any particular language[5] changes constantly and even varies across domains, users, registers etc., scalable natural language understanding systems must consequently be able to cope with language variation and change. Moreover, due to the fact that any natural language understanding system, which is based on some formal representation of that language's grammar, will always only be able to represent a portion (or subset) of what is going on in any particular language at the time. We, therefore, need to find systematic ways of endowing systems that intend to extract meaning from unstructured data as found on the web, with means of learning new forms, new meanings and, ultimately, new form-meaning pairings, i.e. *constructions*.

Acknowledgments

This work has been partially funded by the German Federal Ministry of Research and Technology (BMBF) as part of the SmartWeb project under Grant 01IMD01E and by the Klaus Tschira Foundation. The authors would also like to thank the reviewers for their helpful suggestions and Aldo Gangemi.

References

Arjona, J.L., Corchuelo, R., Ruiz, A. and Toro, M. (2002). A practical agent-based method to extract semantic information from the web. In: *Advanced Information Systems Engineering*, LNCS 2348, Springer, pp. 697–700.

Bergen, B. and Chang, N. (2002). *Embodied Construction Grammar in Simulation-Based Language Understanding*. ICSI Technical Report 02-004. Berkeley, USA.

Berners-Lee, T., Hendler, J., and Lassila, O. (2002) The Semantic Web. In: *Scientific American*, May.

Bryant, J. (2004). Scalable Construction-Based Parsing and Semantic Analysis *In the HLT-NAACL 2004 Workshop on Scalable Natural Language Understanding*, May 7, Boston, Massachusetts, USA, 33—40.

Paul B., Declerck, T, Frank, A., Racioppa, S., Kiesel, M., Sintek, M., Engel, M., Romanelli, M., Sonntag, D., Loos, B., Micelli, V., Porzel, R. and Cimiano, P. (2006). LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies. (this volume).

[3] A gloss-by-gloss translations reads "Ballack heads the 1:0", which means scored a goal by means of a head shot, which resulted in his team leading 1:0.

[4] See Bryant (2004) for details on the *semantic density* algorithm.

[5] The same can be said for dialects, chronolects, sociolects, ideolects, jargons etc., i.e. any solidified system of conventionalized form-meaning pairings.

Chang, N., Feldman, J., Porzel, R. and Sanders, K. (2002). Scaling Cognitive Linguistics: Formalisms for Language Understanding. In: *Proceedings of the 1st International Workshop on Scalable Natural Language Understanding* (ScaNaLU), Heidelberg, Germany.

Cunningham, H. (2005). Information Extraction, Automatic. In *Encyclopedia of Language and Linguistics, 2nd Edition*, Elsevier.

Engel R. (2002). SPIN: Language Understanding for Spoken Dialogue Systems Using a Production System Approach. In: *Proceedings of the 7th International Conference on Spoken Language Processing*. Denver, Colorado, USA 16.09. - 20.09.2002.

Fillmore, C. J. (1982) Frame semantics. In: *Linguistics in the Morning Calm*, The Linguistic Society of Korea, Seoul, pp. 111-138.

Gangemi, A., Mika, P. (2003) Understanding the Semantic Web through Descriptions and Situations. In: *Proceedings of ODBASE03 Conference*, Springer.

Guarino, N. (2006) Ontology Library. WonderWeb Deliverable D202, I STC-CNR, Padova, Italy. See: www.loa-cnr.it/Papers/ Deliverable%202.pdf (last access 04.04.2006).

Goldberg, A. (1995) *Constructions: A Construction Grammar Approach to Argument Structure*. The University of Chicago Press. Chicago.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, (5).

Kay, P. and Fillmore, C.J. (1999). Grammatical constructions and linguistic generalizations: the What's X doing Y? construction. *Language*, 75: 1-33.

Kushmerick, N., Weld, D., and Doorenbos, R. (1997). Wrapper induction for information extraction. In: *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, August 23-29, Morgan Kaufmann.

Lakoff, G. (1987) *Women, Fire, and Dangerous Things.* The University of Chicago Press. Chicago and London.

Langacker, R. (1987) W. *Foundations of Cognitive Grammar, Vol. 1*. University Press. Stanford.

Langacker, R. (2003). Construction Grammars: Cognitve, Radical, and less so. Plenary Paper, *8th International Cognitive Linguistics Conference*, Logrono, Spain, June 25.

Lezius, W. (2002) Morphy - German Morphology, Part-of-Speech Tagging and Applications. In: *Proceedings of the 9th EURALEX International Congress*, pp. 619-623. Stuttgart, Germany.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). Ontology Library". WonderWeb Deliverable D18, I STC-CNR, Padova, Italy. See: http://wonderweb. semanticweb.org/deliverables/documents/D18.pdf, 2003 (last access 04.04.2006).

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R. & Rus, V. (2000). The Structure and Performance of an Open-Domain Question Answering System. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong,1.-8. Oct. 2000, pp. 563-570.

Reithinger, N., Bergweiler, S., Engel, R., Herzog, G., Pfleger, N., Romanelli, M., Sonntag, D. (2005). A Look Under the Hood – Design and Development of the First SmartWeb System Demonstrator. In: *Proceedings of the Seventh International Conference on Multimodal Interfaces*, Trento, Italy.

Simon, K., Lausen, G. (2005). "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions". In: *Proceedings of the 2005 ACM International Conference on Information and Knowledge Management* (CIKM '05), Bremen, Germany.

Stefanowitsch, A. and Gries, S. (2003). Collostructions: Investigating the Interaction between Words and Constructions. In: *International Journal of Corpus Linguistics* 8(2), pp: 209-243.

Pivk, A., Gams, M, Lustrek, M. (2005) Semantic Search in Tabular Structures. *Informatica* 29.

Zorn et al., (forthcoming). From Structure to Semantics. Submitted to the *Mastering the Gap: From Information Extraction to Semantic Representation* Workshop to be held with European Semantic Web Conference 2006 Budva, Montenegro, June 11-14.

Using various semantic relations in Word Sense Disambiguation

György Szarvas[1], Dóra Csendes[1], András Kocsor[2], János Csirik[1]

[1] University of Szeged
Department of Informatics
Szeged, Hungary

[2] Hungarian Academy of Science
Research Group on Artificial Intelligence
Szeged, Hungary

{szarvas, dcsendes, kocsor, csirik}@inf.u-szeged.hu

Abstract

In this paper we discuss an ongoing research where we apply the various semantic relations encoded in the WordNet (Fellbaum, 1998) ontology to perform the semantic disambiguation of English texts. Our word sense disambiguation (WSD) aproach assumes that the proper senses of words share certain semantic relationships, and such relationships can be tracked in lexical databases (or ontologies) like WordNet. The novelty in our WSD algorithm is that it makes use of every relation type encoded in WordNet and Suggested Upper Merged Ontology (that has been mapped to the Princeton WordNet lexicon), while the other approaches studied only taxonomical relationships. We study different weightings to incorporate relations of different type in a semantic distance measure, and search for the sub-graph that has a minimal distance in WordNet, spanned by the words in a sentence being disambiguated. The vertices of the sub-graph identify the wordnet senses belonging to each word.

1. Introduction

The main task of word sense disambiguation is to identify the actual meaning of the word-forms based on their context. The set of possible meanings is usually available as an electronic dictionary or an ontology / semantically enriched linguistic database. This paper discusses our research concerning the usability of the different WordNet relations (Fellbaum, 1998) in word sense disambiguation.

Very often, supervised learning methods are used to select the most probable meaning of the word in a given context, which requires a corpus of manually disambiguated samples. Such a corpus for the English language is the freely available SemCor (Fellbaum, 1998) corpus. In our experiments, this corpus will be used for testing purposes and fine-tuning parameters.

The other common approach does not require the presence of a manually annotated corpus (McCarthy et. al., 2004; Patwardhan et. al., 2003; Pedersen et. al., 2005). Instead, disambiguation is based on the information encoded in an electronic dictionary (in most cases the definitions of the word-forms, its glosses and the relations defined among the dictionary-entries). These methods make use of overlaps among the glosses of words, and length of paths between the word-forms in the relational graph, treated as a semantic distance.

Previous semantic similarity-based approaches either make use of *"is-a"*/taxonomical relations only (see Leacock 1998; Resnik 1998; Sanfilippo 1997 or Pedersen et. al. 2005 for a detailed discussion), or do not give a numeric output (Hirst & St-Onge 1998) that can be used for distance calculations as our method does.

Of course there are systems (Mihalcea and Moldovan, 2001) for the resolution of lexical ambiguity which employ a variety of heuristics.

2. Experiments

In order to map the words of the text with their corresponding meaning in the WordNet ontology, we examined their distance in the ontology. The underlying concept of our method is that the words of a sentence form a semantically coherent structure within the graph of the ontology, i.e. meaning-mapping can be performed by finding the nearest system of word forms using a proper distance measure.

There are several methods that define semantic similarity by using distance within the graph, but in most cases they just consider the hierarchic (hyponym/hypernym) relations of the ontology (Lin, 1998) or were developed for other uses like an organised database of web sites (Maguitman et. al., 2005). The aim of our work was to examine the importance of the great variety of WordNet relations as regards their usability in resolving lexical ambiguities so that with an appropriate weighting we could gain a more efficient, graph-distance based heuristics for resolving lexical ambiguities (and perhaps also identify irrelevant relations).

Due to the lack of appropriate Hungarian resources, we used the English WordNet ontology for our experiments and the disambiguated SemCor corpus for evaluation. The methods we developed will be adapted to Hungarian as soon as the Hungarian WordNet is available.

The word sense disambiguation method we introduce lies between the two types mentioned above. The metrics used for disambiguation are similar to the latter, unsupervised method but we intended to make use of the relation types so far neglected in the disambiguation process and to analyse their impact on efficiency. To do this, we used tagged data for validation purposes. As far as the authors know, no distance-based word sense disambiguation method making use of all the relations stored in WordNet has been developed yet. The only one that exploits other than the hierarchical relations is the method of Hirst and St-Onge (Hirst and St-Onge, 1998), and its variants.

During our experiments we computed the sum of distances between the vertices of the sub-graph spanned by the words of the sentence, taking into consideration the possible meanings of each word.

The sub-graph with the lowest distance-sum shows the most probable meaning of the words in a given context. We might consider different relations with different weights in the calculation of the distance; in this case each possible weighting defines a word sense disambiguating heuristics. By tuning the weights we can search for the most appropriate weighting for resolving ambiguities. During the fine tuning process distances had to be computed on-line, which was quite a time consuming process. As soon as the best performing weights for WSD were given all the pair-wise distances of concepts in the WordNet ontology can be computed off-line to speed up the disambiguation process.

The pair-wise distances of the meaningful[1] words in a sentence can be arranged in a structure that we call the semantic distance matrix (an example of a semantic distance matrix can be seen in Figure 1). Disambiguation was performed by searching for the minimal sub-structure of the matrix. The distances were obtained by an algorithm described by the following pseudo code:

Pseudo code 1

For all the sentences in the text
    For all meaningful words in the sentence
    For al possible senses of the word
        Perform Dijkstra's alg.
    End
    End
End

Dikstra's algorithm will terminate as soon as all the senses of all the words in the sentence have been reached. Another speed-up can be achieved if inverse relations are given the same weights – in this case the distance between two concepts is always the same in both directions.

In the figure below, the semantic distance matrix defined by the words of the sentence "NOR COULD HE CALL UP (#3) MEMORY-PICTURES (#1) OF CLOSE (#2) FRIENDS (#1) OR RELATIVES (#1)." is displayed. Each row of the matrix shows the distance of every single potential meaning of the word to be disambiguated from the various alternatives of the other words. The aim here was to map a possible meaning to each word so that the sum of the elements in the intersection of rows and columns marked would be minimal. In the case of the example sentence above, the simplest heuristics, taking each relation into account with the same unit weight, creates an unambiguous optimum, and provides the correct mapping of meanings.

Distance	Relation	Synset ID	POS	Literals
0		95716	a	close#2
1	&	95719	a	chummy#1, buddy-buddy#1, thick#1
2	;u	37177	n	colloquialism#1
3	-u	30902	n	think#1
4	+	87705	v	think#1
5	@	87702	v	imagine#1, conceive of#2, envisage#1, …
6	~	87699	v	visualize#1, project#1, fancy#1, see#1, figure#1, …
7	+	31552	n	image#1, mental image#1
8	~	31595	n	memory image#1
9	~	31596	n	memory picture#1

Table 1: Semantic path between *close#2* and *memory picture#1*

[1] We call a word form meaningful if it is a literal in at least 1 WN synset with the proper POS.

0		82638	v	remember#1, retrieve#1, recall#1, call back#1, call up#3, recollect#1, think#3
1	+	30902	n	idea#1, thought#2
2	+	82745	v	think#4, opine#3, suppose#1, imagine#1, reckon#3, guess#1
3	+	53029	n	guesser#1
4	@	14	n	person#1, individual#1, someone#1, somebody#1, mortal#1, human#1, soul#1
5	~	53510	n	relative#1, relation#1

Table 2: Semantic path between *call up#3* and *relative#1*

+	Derivationally related form
;u	Domain of synset usage
-u	Member of this domain usage
=	Attribute
&	Similar to
@	Hypernym
~	Hyponym

Table 3: Types of relations used for the example sentence

As the reader can see, various types of relations play an important role in the semantic distance computation of concepts. If different relations are taken into account with different weights, the optimal solution of our disambiguating algorithm will be different as well. Hence, searching for the best performing weights for WSD is a straightforward task. Apart from obtaining better results, this also allows us to assess the role or importance of each relation in Word Sense Disambiguation. As these preliminary results show, incorporating relations different from hypernimy/hyponimy to compute semantic similarity can be beneficial to WSD and help us better understand the role each relation plays in resolving ambiguities.

	call up # 1	*call up # 2*	call up # 3	*call up # 4*	memory picture # 1	*close # 1*	close # 2	*close # 3*	*close # 4*	*close # 5*	*friend # 1*	*friend # 2*	*friend # 3*	*friend # 4*	*friend # 5*	relative # 1	*relative # 2*
call up # 1					8	9	8	9	11	9	6	8	6	7	8	6	5
call up # 2					9	8	7	8	10	8	5	6	5	6	7	5	6
call up # 3					**6**	5	**4**	5	8	5	**5**	7	5	5	7	**5**	6
call up # 4					9	8	8	8	7	8	6	6	6	7	8	6	6
memory picture # 1	8	9	**6**	9		10	**9**	10	12	10	**7**	9	7	7	9	**7**	8
close # 1	9	8	5	8	10						6	8	6	7	8	6	7
close # 2	8	7	**4**	8	**9**						**5**	7	5	6	7	**5**	6
close # 3	9	8	5	8	10						6	8	6	7	8	6	7
close # 4	11	10	8	7	12						7	9	7	8	9	7	8
close # 5	9	8	5	8	10						6	8	6	7	8	6	7
friend # 1	6	5	**5**	6	**7**	6	**5**	6	7	6						**2**	3
friend # 2	8	6	7	6	9	8	7	8	9	8						4	5
friend # 3	6	5	5	6	7	6	5	6	7	6						2	3
friend # 4	7	6	5	7	7	7	6	7	8	7						3	4
friend # 5	8	7	7	8	9	8	7	8	9	8						4	5
relative # 1	6	5	**5**	6	**7**	6	**5**	6	7	6	**2**	4	2	3	4		
relative # 2	5	6	6	7	8	7	6	7	8	7	3	5	3	4	5		

Figure 1: Semantic distance matrix of the sentence ”Nor could he call up memory-pictures of close friends or relatives.”. Insufficient meanings are italicised, while the semantic distance of sufficient meanings for each is shown in bold.

In our ongoing research we concentrate on the following tasks:

- The development of an effective algorithm for a mapping which defines the lowest total distance in matrices of the type illustrated in the figure above
- Examining the highlighted role of unambiguous (of single meaning) words and whether the highlighted role of these words might improve accuracy or speed
- Whether in defining the structure with the minimal total distance, it is better to examine the distance of the words of the sentence from all the other words, or whether it is enough to limit to a certain environment (is there a semantic cohesion between distant parts of the sentence or only between closely positioned words?)
- The effect of using the results of syntactic analysis to reduce the search of disambiguation (the search is limited not to the close environment of the word, but to the words related to it in the syntax tree)
- Using varying edge-weights in distance computations (an edge describing hyponym or hypernym relation for example, does not define the same semantic distance in the semantic hierarchy at the lower and the upper levels which describe more abstract concepts).

3. Summary

The approach of word sense disambiguation through WordNet synsets that we presented above has been a research trend for decades. One major group of the disambiguation techniques available in literature is based on algorithms built on semantic distance / similarity metrics implemented on WordNet structure as a tagged, directed graph; the method we introduced here is one of them.

The word sense disambiguation technique introduced by the authors is an improvement on the currently available ones in that it makes use of all the relations in the WordNet in examining the lengths of paths within the graph as each ontological relation establishes a semantic relationship between the particular concepts; thus taking them into account when calculating the distances is a reasonable one. As a result of the experiment we not only get new heuristics, but by the tuning of weights used in calculation of the distance metrics, we are also able to evaluate WordNet relations, and this shows how useful the relation type is in resolving the ambiguities of word meaning.

4. References

Martin, L.E. (1990). Knowledge Extraction. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 252-262.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* The MIT Press, USA

Hirst, G. & St-Onge, D. (1998). *Lexical chains as representations of context for the detection and correction of malapropisms.* In: Christiane Fellbaum (Ed.), WordNet: An electronic lexical database, pp. 305–332., MIT Press

Leacock, C. & Chodorow, M. (1998) *Combining local context and WordNet similarity for word sense identification*, in: C. Fellbaum (Ed.), WordNet: An electronic lexical database, MIT Press, pp. 265–283.

Lin, D. (1998). *An Information-Theoretic Definition of Similarity*. In Proceedings of the 15th International Conf. on Machine Learning, Madison, Wisconsin

Maguitman, A.G., Menczer, F., Roinestad, H., & Vespignani, A.. (2005). *Algorithmic Detection of Semantic Similarity.* Proceedings of the 14th International World Wide Web Conference, Chiba, Japan

McCarthy, D., Koeling, R., & Weeds, J. (2004). *Ranking WordNet senses automatically,* Technical Report CSRP 569. University of Sussex

Mihalcea R.F. & Moldovan, D.I. (2001). *A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation.* International Journal on Artificial Intelligence Tools, Vol. 10, No. 1-2

Patwardhan, S., Banerjee, S. & Pedersen, T. (2003). *Using Measures of Semantic Relatedness for Word Sense Disambiguation.* Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City

Pedersen, T., Banerjee, S. & Patwardhan, S. (2005). *Maximizing Semantic Relatedness to Perform Word Sense Disambiguation.* Research Report UMSI 2005/25 (http://www.msi.umn.edu/general/Reports/rptfiles/2005-25.pdf) University of Minnesota, Duluth

Resnik, P. (1998) *Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language*, Journal of Artificial Intelligence Research 11. 95–130.

Sanfilippo, A. (1997) *Using Semantic Similarity to Acquire Co-occurrence Restrictions from Corpora*. Proceedings of the ACL-97 Workshop on Automatic Information Extraction and Building of Automatic Resources for NLP Applications, Madrid

SMARTINDEXER – Amalgamating Ontologies and Lexical Resources for Document Indexing

H. Peter, H. Sack, C. Beckstein

Institut für Informatik
Friedrich-Schiller-Universität Jena
D-07743 Jena
Germany
{hpeter, sack, beckstein}@minet.uni-jena.de

Abstract

Document index compilation is a sophisticated task that requires text understanding capabilities. SmartIndexer supports the author in the process of index compilation. By providing information about the general structure of an index in combination with the lexical and semantic resources of WordNet, SmartIndexer gives suggestions for arranging potential index entries according to their semantic relationships and according to the requirements of the author. In addition, the process of index compilation can be reversed in the sense that an existing document index can be used for automated semantic annotation of the underlying document.

1. Introduction

The index is an essential part of any document, no matter if we consider a book, an issue of a magazine, a web page, or any other information source. It allows fast and efficient random access to any important topic within the document. The process of index creation is not trivial and thus requires extensive intellectual efforts: Appropriate headings must be chosen, index entries must be defined sophistically, synonymy, ambiguities and other relationships between index entries must be detected and handled properly. In the end, the creation of a sound index also affects the corresponding document because it provokes text restructuring and disambiguation of the used vocabulary.

Current indexing software (e.g. LaTeX's MakeIndex (Lamport, 1987) or MACREX(Calvert and Calvert, 1997)) supports the author only in mechanical indexing tasks, e.g. simple management or sorting of index entries. This type of software also does not assist the author in the much more complex and creative task of originating accurate and sound index entries. An entirely automated indexing process requires text understanding capabilities that are beyond the ability of prevailing computer systems.

Our goal was to develop an architecture – the SMARTINDEXER – that supports the author in the creative tasks of the indexing process. For this purpose, we designed an ontology (in the following referred to as *Index Ontology*), which contains general knowledge about index elements and their relationships. Index quality strongly depends on the amount of its inherent semantics. An index can be regarded as a network, where the index entries represent the nodes. Subentry relationship between two index entries as well as different cross-references among index entries constitute the arcs. This network embodies the semantic interrelationships inherent in the index. SMARTINDEXER facilitates the creation, expansion, and management of this network and thus, enables the generation of a high quality index.

Providing semantic relationships between words, as e.g. hyperonymy or meronymy, is the main task of the electronic lexical database WordNet (Fellbaum, 1998). SMARTINDEXER employs WordNet in connection with its Index Ontology to assist the author at the intellectually sophisticated indexing task. Supplementary, domain ontologies – if available – provide useful information about a document's subject. SMARTINDEXER can use these ontologies as significant input beyond the knowledge offered by WordNet. The paper is structured as follows: Section 2 and Section 3 introduce the reader to the basic principles of indexing. Section 4 covers the architecture of the SMARTINDEXER, while Section 5 gives a short overview of the SMARTINDEXER algorithm. In Section 6 a possible transformation of a document index into a domain ontology is shown. Section 7 concludes the paper with an outlook on ongoing and future work.

2. Index and Index Elements

According to the British Indexing Standard (Mulvany, 1994) an index is a systematic arrangement of entries designed to enable users to efficiently locate information in a document or specific documents in a collection.

Index entry: An index entry consists of a heading (or main heading) and at least one of the following components: a subentry, a reference locator (in the following referred to as locator), or a cross-reference. A heading is a term – normally a noun or a noun phrase – which reflects a concept in the document.

Subentry: A subentry is similarly structured as an index entry. It is composed of a subheading, one or more locators, and – only rarely – cross-references. The corresponding concepts of the subheadings are always related to the concept of the superordinated main heading. In the majority of cases subheadings represent subdivisions or more specific aspects of the main heading.

Sub-subentry: A subentry can have further index entries – so called sub-subentries. The above mentioned statements about subentries hold analogously for sub-subentries. In general it is not recommended to go beyond the level of sub-subentries.

Locator: Locators follow a heading and indicate that part of a document, where information related to the heading can be found. In printed media, reference locators are usually page numbers, section numbers, or line numbers.
Cross-reference: Cross-references establish a relationship between one heading and another. This makes it possible to connect scattered information within the index. A book index usually provides two kinds of cross-references: *see* references and *see also* references. The first kind is used for variant spellings, synonyms, aliases, abbreviations, and so on. *See also* references are used to guide the user to another closely related heading that supplies additional information.
A high quality index is an essential prerequisite for efficient information retrieval. Direct access to specific information within a document becomes hardly viable without an index.

3. Index Compilation

Compiling an index is an intellectually sophisticated process. The difficulty of that process lies in capturing the essence of a document by means of only a few short, expressive and predictable headings or heading phrases. Furthermore, synonyms, ambiguities, and various relationships between terms have to be detected and handled properly.
The index compilation process usually consists of the following six steps:

1. Terms are highlighted that are considered to be main headings or subheadings in the index. Each highlighted term is augmented with additional and more specific information. This information will be used in a subsequent step for the generation of subheadings.

2. A corresponding locator is assigned to each highlighted term.

3. Then, highlighted terms and locators are arranged in order within the existing index. There are several possible index orderings. The most commonly used index order is the alphabetical order.

The remaining three steps generate a consistent document index from the collected temporary index entries:

4. It has to be decided, which term is transformed from a set of synonyms or closely related terms into a main heading. Furthermore, appropriate cross-references have to be created that reflect the existing semantic relationships.

5. Then, an index level has to be chosen, where the index entries have to be placed.

6. Finally, it has to be verified that all cross-references relate to an existing index entry that offers a locator.

The mere mechanical aspects of index creation (step two and three) can be accomplished easily with current indexing software. However, the author usually does not obtain any support in the intellectual aspects of index creation. Thus, the goal of our SMARTINDEXER architecture is to assist the author in the creative tasks of index compilation – especially in steps four to six.

4. The SMARTINDEXER Architecture

The SMARTINDEXER architecture is based on a two component framework: the *Index Generator* and the *Ontology Processor* (for an outline of the SMARTINDEXER work flow see figure 1).
The Index Generator receives as input a potential index entry from an arbitrary word processing application (1). Additionally, an already existing document index is passed to the Index Generator (2). After a preprocessing step containing (among other things) spell checking and word stemming, the author has to mark up the sense carrying substring (SCS) of the potential index entry. Then, the SCS is passed over to the Ontology Processor (3). The Ontology Processor recalls the entire lexical field (LF[SCS]) of the SCS by means of WordNet (4,5). LF[SCS] contains synonyms, hyponyms, hypernyms, holonyms, meronyms, and sister terms of the SCS. After this lookup, the SCS is passed back to the Index Generator (6). The Index Generator uses the general knowledge about indexing represented by the Index Ontology for making suggestions about new potential index elements, as e.g. cross-references or subentries (see section 5. for a more detailed specification of the indexing algorithm).
In addition to lexical resources as e.g. WordNet, SMARTINDEXER can use different knowledge repositories. The author has the possibility to supply domain ontologies referring to the subjects discussed in the document to be indexed. If there is no suitable ontology available, standard WWW search engines as e.g. Google or specialized semantic search engines as e.g. Swoogle (Ding et al., 2004) can be used for searching better suited ontologies (see figure 2). Typically, domain ontologies describe domain entities and various relationships between them. In particular 'IS-A' or 'PART-OF' relationships are good candidates for possible index elements, especially for cross-references.
With the help of the Index Ontology the Ontology Processor filters the found relationships and transfers them to the Index Generator. Depending on this information the Index Generator suggests suitable index elements and lets the author decide which of them to include in the document index. Finally, the Index Generator returns the chosen index elements to the word processing application (7) that inserts the new index entry into the document index (8). In certain situations inserting a new index entry requires complex index rearrangement.
SMARTINDEXER is being implemented as a Java application independent of specific hardware or operating systems. For the management of semantic information provided by RDF, RDFS, and OWL ontologies, we use the JENA application programming interface (McBride, 2002). The preprocessing of possible index entries requires word stemming, which is performed with the Java implementation of the Porter stemming algorithm (Porter, 1980). In order to access lexical information provided by WordNet we use the Java Word Net Library (JWNL) (Didion, 2004).

5. SMARTINDEXER Algorithm

The SMARTINDEXER has to be able to detect relationships between index entries to properly assist the author with index compilation. This requires that the concept of a heading

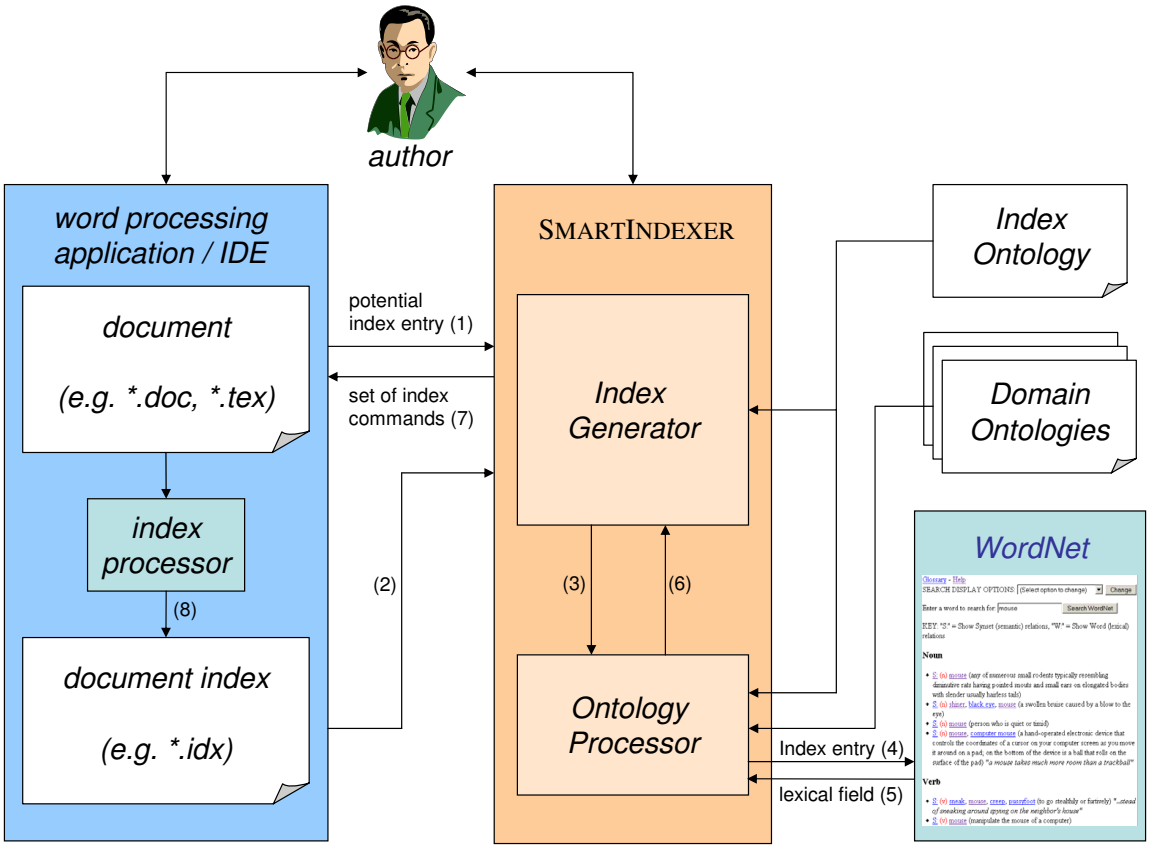


Figure 1: Indexing Process with SMARTINDEXER

is known. By knowing the concept of a heading SMARTINDEXER is able to identify relationships between index entries by the combined use of the Index Ontology and lexical resources as WordNet. The Index Ontology provides general knowledge about the components of an index and their relationships with each other.

As already mentioned, the Index Generator uses the general knowledge about index creation and the information offered by WordNet to make suggestions for a potential index entry i. First, the underlying concept of i is determined in a preprocessing step in cooperation with the author. The preprocessing comprises the following operations:

1. Perform spell checking of i and stop word removal from i.
2. Ask the author to mark up the sense carrying substring (SCS) of i.
3. Perform word stemming of i according to the porter stemming algorithm (this step is already provided by WordNet).
4. Use WordNet or available domain ontologies to determine the underlying concept of i, which will be used in the main index processing algorithm. This step has to be directed by the author.

Preprocessing must be guided by the author because the SMARTINDEXER algorithm is not able to determine the underlying concept of i. In order to realize this step in an autonomous way text understanding capabilities are indispensable.

WordNet contains so called synsets representing concepts that are identified with the help of so called sense keys. The sense key resulting from preprocessing of i is a prerequisite for the identification of the semantic relationships between i and the existing document index.

This main indexing process can be divided into two main steps: First, a possible position p of i within the already existing document index I has to be determined. Then, the new index entry i has to be inserted at position p either with its locator or as a cross reference. To accomplish both steps the semantic relationships between i and the existing index entries $j \in I$ have to be located. This can be achieved in the following way:

1. Determine the position p of the new index entry i within the existing document index I. This is done depending on the type of information available about i:
   - If i is a synonym of an already existing index entry $j \in I$, then the position p of the new index entry i can be the same as the position of j.
   - If there are already known subordinated relationships (e.g. hyponyms, meronyms) of the new index entry i and the already existing index entries $j \in I$, then i can be positioned at the position of j, while j has to be relocated below the new index entry i.

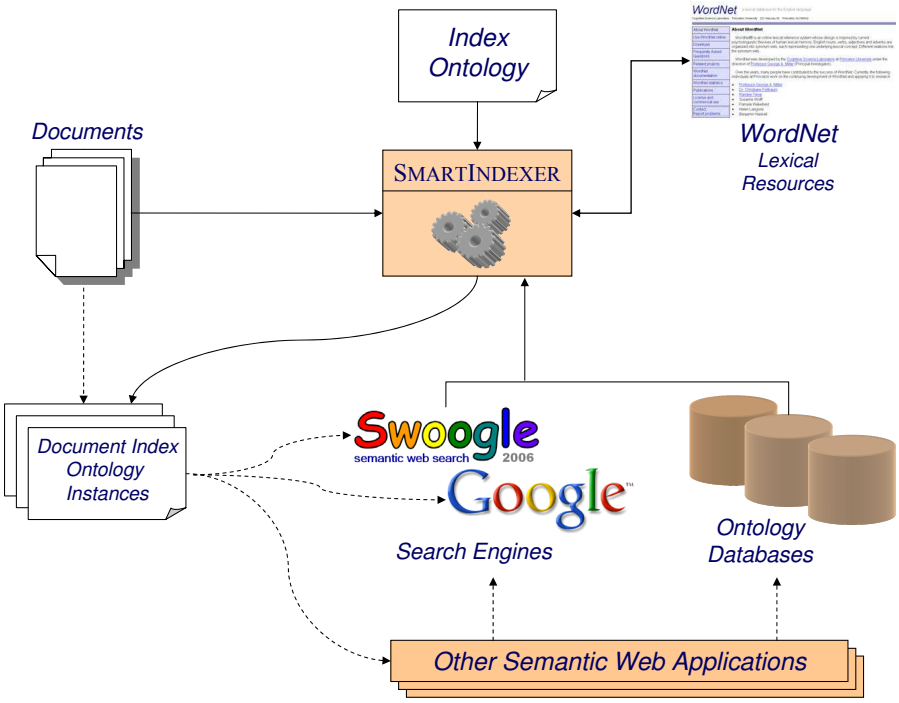


Figure 2: SMARTINDEXER Embedded in Semantic Web Framework

- If there are already known superordinated relationships (e.g. hypernyms, holonyms) of the new index entry i and the already existing index entries $j \in I$, then i can be positioned below the already existing superordinated index entry j.
- Otherwise, if there are already known associated terms (e.g. sister terms) of the new index entry i and the already existing index entries $j \in I$, they can be used to find a suitable position for the new index entry i in I. If i is a sister term of j, i can be positioned at the same index level as j.

2. Insert the new index entry i at position p with its locator or as a cross-reference:

   - *see* references can be already existing index entries $j \in I$, which have a synonymic relationship with i.
   - *see also* references can be already existing index entries $j \in I$, which have any semantic relationship with i.

SMARTINDEXER only gives suggestions, where to insert a new index entry into the existing document index. The final decision, where to supply the new index entry is up to the author.

The index compilation process is illustrated with the following example (see figure 3). The new index entry *mouse* has to be inserted into an existing index. After the preprocessing step SMARTINDEXER determines *mouse* to be a direct hyponym of the existing main heading *rodent*. Additionally, *mouse* also is determined to be a direct hypernym of the existing main heading *field mouse*. SMARTINDEXER suggests two possible arrangements to the author, who determines which of the proposed variants should be used. Choosing the second variant requires the rearrangement of already existing index entries. The new index entry *mouse* becomes a main heading, while *field mouse* and its subordinated index entries become subentries of *mouse*.

6. Embedding SMARTINDEXER within the Semantic Web Framework

A document index provides direct access to specific information within the document. It can be considered as a very condensed summary of the underlying document and thus, also providing access to essential concepts within the document.

By reversing the index compilation process, SMARTINDEXER can also be utilized to transform an already existing document index into an ontology that captures important semantic knowledge about the document. For this purpose, the already mentioned Index Ontology has to be considered to be a generic class framework for the index at large. Accordingly, a document index has to be considered to be a specific instance of the general Index Ontology.

By making use of this consideration, we have the possibility to transform any document index file into an RDF file reflecting all the relationships defined by the underlying document index instance. The resulting RDF file can be used to provide a traditional index representation, i.e. an

Index (before insertion)

fieldmouse, 13, 15
- prairie vole, 16
- meadow vole, 16
- habitat, 15
- *see also* rodent

rodent, 1
- beaver, 10, 11
- dentision
  - incisor, 4
  - rotation of teeth, 5
- hamster, 6, 8 – 10
- *see also* field mouse

Index (after insertion)

mouse, 12
- **fieldmouse, 13, 15**
  - **prairie vole, 16**
  - **meadow vole, 16**
  - **habitat, 15**
  - ***see also* rodent**

rodent, 1
- beaver, 10, 11
- dentision
  - incisor, 4
  - rotation of teeth, 5
- hamster, 6, 8 – 10
- *see also* field mouse

Figure 3: Example of Index Entry Insertion of the new Index Entry *mouse*

alphabetically ordered list of index entries.
In addition, one can use the RDF data structure to display the index in different alternative ways that provide supplementary information. It is possible to display the document index as a topic map or as a graph, clarifying the relationships between the index entries by graphical visualizations that can be used for inner document navigation. Furthermore, the RDF index instance of the document index identifies the significant keywords of a document, thus providing information about what is important and what is not. According to this kind of interpretation, index entries with a large number of references in the document can be considered to be of higher significance than index entries with only a single reference.
Index entries also reflect how index keywords do interact, e.g. by giving information about hyperonymy, meronymy, homonymy, synonymy, and other kind of associations. This additional semantic information can be used to draw new links between different sections of the document. It enables the reader to break out of the linear text flow of the document by using cross connected index keywords (*see* and *see also* references) like the hypertext links.
The semantic relationships provided by the index can be utilized as a starting point for further semantic annotation of the document related to the index. Also corresponding domain ontologies, which match the concepts provided by the document can be identified more easily.

7. Conclusions and Outlook

Document index compilation is a sophisticated task. It requires smart knowledge processing. SMARTINDEXER supports the author during the process of index compilation. The compilation of a sound document index requires the identification of circles or blind references. This is accomplished by using a semantic index description (Index Ontology) in combination with the lexical resources provided by WordNet. The document and the index ontology together with WordNet's semantic relationships fosters the emergence of a new ontology from the document's index. This ontology can be used for visualization and navigation issues. Furthermore, it can as well supply additional semantic information for the underlying document. Therefore, SmartIndexer can be considered as being a first step towards semantic document annotation, which is mandatory for enabling the semantic web.

8. References

Grigoris Antoniou and Frank van Harmelen. 2004. *A Semantic Web Primer*. The MIT Press, Cambridge, Massachusetts.

Hilary Calvert and Drusilla Calvert. 1997. MACREX Manual for Version 6.5.

John Didion. 2004. The Java WordNet Library.

Li Ding, Timothy W. Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. 2004. Swoogle: A Search and Metadata Engine for the Semantic Web. In Grossman et al. (Grossman et al., 2004), pages 652–659.

C. Fellbaum. 1998. *WordNet – An Electronic Lexical Database*. MIT Press.

David Grossman, Luis Gravano, ChengXiang Zhai, Otthein Herzog, and David A. Evans, editors. 2004. *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004*. ACM.

Leslie Lamport. 1987. MakeIndex: An Index Processor for LaTeX.

Brian McBride. 2002. Jena: A Semantic Web Toolkit. *IEEE Internet Computing*, 6(6):55–59.

Nancy C. Mulvany. 1994. *Indexing Books*. The University of Chicago Press, Chicago.

Martin F. Porter. 1980. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137.

Laurent Prévot, Stefano Borgo, and Alessandro Oltramari. 2005. Interfacing Ontologies and Lexical Resources. In *Proceedings of OntoLex 2005 - Ontologies and Lexical Resources*, Jeju Island, Republic of Korea, 15 October.

Frank van Harmelen, Jeen Broekstra, Christiaan Fluit, Herko ter Horst, Arjohn Kampman, Jos van der Meer,

and Marta Sabou. 2001. Ontology-Based Information Visualisation. In *International Conference on Information Visualisation, IV 2001*, pages 555–562, London, England, 25-27 July.

How Linguistic Resources May Help to Recommend TV Programmes

Bernd Ludwig, Stefan Mandl, and Sebastian Schmidt
Chair for Artificial Intelligence
University of Erlangen-Nürnberg
Am Weichselgarten 9, D-91058 Erlangen
firstname.lastname@informatik.uni-erlangen.de

Abstract

This paper presents an approach to exploit free text descriptions of TV programmes as available from EPG data sets for a TV recommendation system that takes the content of programmes into account[1]. The paper focusses on the natural language understanding problem underlying the analysis of free text descriptions and on methods of classifying free text descriptions in relation to a natural language user query.

A evaluation of user acceptance is presented. The paper closes with a discussion of future work and proposals of how to integrate our search algorithm into recommendation systems that are described in the literature (see (Ardissono et al., 2001; Gena, 2001; Buczak et al., 2002)).

1 Introduction

The Electronic Programme Guide (EPG) provides an enormous amount of information about TV programmes. Viewers are overwhelmed by the huge number of programmes when they select a programme to watch. User models in current recommendation systems allow to search for certain features of programmes, like genre, starting time, and other information – these features are quite easy to retrieve and compare by database queries.

However, viewers would prefer to know more about the content of a programme when deciding whether to watch it or not. This requires a recommender to have available information about the content and to be capable of performing a semantic analysis to meet the viewer's interests.

[1]The research presented in this paper is sponsored by *Software-Offensive Bayern* (http://www.software-offensive-bayern.de/english.xml). Ferdinand Herrmann, Heike Ott, Kristina Makedonska, and Sebastian von Mammen provided valuable help implementing major parts of the presented system.

The paper gives an overview of our recommendation system. This comprises a natural language query in which a viewer expresses his/her current preferences, long-term information about the viewer stored in a user model, and long-term information about viewer groups stored in particular user group models. At the beginning of the paper, we compare our work with previous research efforts. In Section 3 we report on a user study that shows how viewers select programmes. In Section 4 we explain our approach to analyse free text descriptions. We conclude the paper with a report on a first evaluation of the system in Section 5 and some remarks on future work.

2 Related Work

The design and implementation of TV recommendation systems has attracted great interest in different research groups already. They use sophisticated user models such as (Ardissono et al., 2004). In order to allow for default reasoning, stereotypes for users are applied which are based on the analysis of the average user's lifestyle (see (Gena, 2001)). Much attention is paid on the issue on how to design an attractive, functional, and easy-to-use graphical interface between users and the recommendation system (see (van Barneveld and van Setten, 2004)). In order to increase the user's confidence in the system proposals, the generation of trust-worthy suggestions that take programmes watched ealier into account has been studied in detail (see (Buczak et al., 2002)). All these research directions deliver valuable contributions to building recommendation systems. But they do not cope with the question of how a system could take the contents of TV programmes into account.

The work presented in this paper addresses this issue by developing a technique of shallow semantic analysis of free text descriptions available by state-of-the-art EPG systems. The results of the work are seen as another contribution to the com-

plex task of building informative and attractive TV recommendation systems.

Next generation TV sets will be equipped with embedded systems that provide a huge amount of computational ressources and will allow the development of software that runs on the TV set in the background offering extra features for the user's convenience, e.g. a recommendation system. No additional PC will be necessary.

3 How Viewers Select Programmes ...

A user study (Nitschke and Hellenschmidt, 2003) conducted as part of the research project EMBASSI (see (Herfet et al., 2001; Kirste and Heider, 2002)) revealed a number of interesting facts about how users like to select TV programmes. Candidates were situated in front of a computer display that suggested an automatic recommendation system to be at work. Actually, in a room nearby, a human person monitored the candidate and responded according to the information available from a TV magazine. In the experiments, the users were allowed to ask arbitrary questions about the currently available TV programmes. On the display a list of proposals was presented and users could ask more specific questions on certain proposals or start a new search if they wanted to.

The Wizard-of-Oz experiments showed that almost all test persons ask questions about the *content* of a TV programme:

> *Ich will was Lustiges oder Informatives sehen. (I want to see something funny or informative.)*
>
> *Spannung, Fantasie, Fabelwesen (Tension, fantasy, mythical creatures)*
>
> *Liebe, Romantik (Love, romance)*
>
> *Eine politische Sendung (A political programme)*
>
> *Dokumentationsreihe über den zweiten Weltkrieg und die Landung in der Normandie (Documentary about World War II and Operation Overlord)*

Users often expressed emotional attitudes they desired the programme to have, or even their own emotions hoping the system would come up with proposals that match their mood:

> *Entspannen (To relax)*
>
> *Show, Witz (Show, fun)*

Kate's boss feels that a married person is the best bet for a promotion because they tend to stay put and enhance the firm. In order to advance her career, Kate must find a way to pose as "attached." She also has a crush on co-worker Sam who is only interested in girls who are spoken for. She fulfills both requirements by hiring Nick, a young man she just met at a friend's wedding, to pose as her beloved. Nick agrees to play the part of the "picture perfect fiance" but soon falls in love with Kate for real. When Nick decides to head home, Kate soon realizes what true love can be.

Figure 1: Free text description of the movie *Picture Perfect* (English version from `http://www.hollywood.com`

> *Ich möchte gern etwas Spannendes sehen. Humor sollte auch dabei sein. (I'd like to watch something thrilling. It should also be humorous.)*
>
> *Ich bin gerade müde und erschöpft und wünsche mir ein bißchen Harmonie. (I am tired and exhausted and I wish for some harmony.)*

With the data available in standard EPG, it is hard to retrieve the necessary information to answer such complex queries. Current recommendation systems use some sort of formal typology for genres. As explained e.g. in (Ardissono et al., 2001), the typology of such recommendation systems is based on a sort of specialised ontology that provides a sort of standard for classifying the content of TV programmes[2]. With such a categorial system, many user requests can be satisfied with proposals that match well. However, the examples above require inference capabilities that are beyond the limits of a typology, as more specific information on the content is required to give satisfying answers.

4 Recommendations on the Basis of Free Text Descriptions

Recommendations for TV programmes could be improved if they relied not on genre types only, but if it was possible to know more about the content of programmes and connect this information appropriately with its knowledge about the user and

[2]ETSI EN 300707 by the European Telecommunications Standards Institute (available via `http://www.etsi.org`) specifies such a standard

his/her inquiry in natural language. The remainder of this section explains our solution to this issue.

4.1 EPG Data

For generating user-tailored recommendations that take the content of programmes into account, we rely on natural language information available in EPG data (which is provided by the TV stations).

In Figure 1 you can see an example in German with a paraphrase in English. In a few sentences the content of the movie is described. In order to compare this description with the criteria in a user query (such as *entertainment*, *action*, *funny* and so on), we try to get an overview of the topics the description talks about. In the example, such a list of topics could consist of:

- professional career
- missing success
- nice, intelligent, young, single woman
- problems in the job
- love story

Each of these topics is more or less related to one or more of the user's criteria. So, if one could somehow extract such a topics list from each EPG data item available (representing each available TV programme), one could try to find those items whose contents are closest to the user's criteria and suggest them as the best recommendations.

Given the current state of the art in natural language understanding, it is impossible for a recommendation system to understand the free text description in the way humans do. Therefore, a method of shallow semantic analysis is required that extracts topics out of a given description.

4.2 The DORNSEIFF Lexicon

Our approach to shallow analysis is based on the DORNSEIFF lexicon for German. Like a thesaurus, it groups words according to certain topics, i.e. in each group there are words (even of different word categories) that describe a particular aspect of a certain topic. The DORNSEIFF lexicon[3] is not a synonym lexicon, but a "topic" lexicon. In a two-level hierarchy, the lexicon organizes topics in chapters (e.g. chapter 15 contains subtopics *social life*) and subchapters (e.g. subchapter 15.39 is the topic *reward*). If the meaning of a word is ambiguous, it is listed in more than one subchapter. For the German word *Beförderung* in the description of the movie *Picture Perfect* there are four topics the word is related to:

group id	description
8.5	*Beförderung* (*transport*)
9.33	*Vollenden* (*completion*)
15.39	*Belohnung* (*reward*)
15.62	*Ehre, Ruhm* (*fame, glory*)

As an example for a lexicon entry, we give the translations of the nouns in group 15.39.

entry in German	English paraphrase
Auszeichnung	*decoration*
Beförderung	*promotion*
Belohnung	*gratification*
Ehrenbürgerschaft	*honorary citizenship*
Ehrensold	*gratuity*
Gehaltserhöhung	*raise of salary*
Prämie	*bonus*
Prämierung	*giving a bonus to sb.*

As English words have connotations different from those of German words, a perfect translation is almost impossible without context. Therefore, the example here is mainly to show that words in a group tend to be related to a common topic.

4.3 Technical Overview

Before discussing how the DORNSEIFF lexicon is applied to classify TV programmes according to a user-defined topic field, we briefly describe the architecture of our system and how real EPG data are accessed. As far as hardware is concerned, the system is based on a Linux machine with a special hardware component for receiving and decoding satellite signals EPG data are retrieved via satellite and stored in a data base that is available for reading from any process running on the Linux machine. The user interface is multimodal, consisting of a graphical user interface and a natural language dialogue system. The user can communicate with the system with a remote control or in German, depending on his/her personal preferences. When viewers look for programmes e.g. of a certain genre, channel, or start time, standard data base filtering is used to retrieve matching proposals. Implicit criteria, such as those presented in the previous section, are processed by the recommendation system in a special way that will be discussed in the remainder of this paper.

[3] Interested readers can test the online version of the DORNSEIFF lexicon on `http://wortschatz.uni-leipzig.de`.

group	#	group	#	group	#	group	#	group	#	group	#
intensity	1	creation	2	maintenance	1	duration	2	visible	1	stopping	1
steering	1	pull	1	plan	1	random	1	work	2	preparation	1
custom	1	easy	1	high quality	1	improve	1	cooperate	1	help	1
prohibit	1	success	2	wit	1	hope	1	wish	1	love	2
think	1	reason	1	creativity	1	illusion	1	learn	1	insane	1
secret	1	reveal	4	notify	1	advice	1	affirm	1	proof	1
truth	2	pop music	1	family	1	marriage	1	single	2	applause	1
harmony	1	friendship	2	reward	2	unsociable	1	resistance	1	fight	1
victory	1	glory	1	hot, salty	1	sports	1	game	1	reign	2
authority	2	command	1	obligation	1	subserviency	1	imprisonment	1	acquisition	5
grant	3	sell	1	dishonest	1						

Figure 2: DORNSEIFF characterisation (group numbers omitted) of the free text description in Fig. 1

4.4 How Free Text Descriptions Are Used

How do user inputs typically look like? Syntactically, they vary between the extremes of keywords only and complex sentences. As people formulate arbitrarily complex sentences that go beyond the capabilities of any natural language understanding algorithm, a chunk parser (see (Bücher et al., 2002)) is used to process user utterances. This method is quite robust for processing keywords and extracting the main information out of more complex queries. For each chunk, the DORNSEIFF groups are computed. This analysis of the user input results in a characterisation of the topics addressed by the user query (see Fig. 2).

In principle, generating recommendations is then reduced to finding TV programmes whose free text descriptions are as close as possible to the characterisation of the user query.

For determining this distance between a programme and the user's interest, free text descriptions of TV programmes are processed in the same way as the user query: For each chunk the DORNSEIFF groups are computed. All different readings are taken into account, since it is impossible for the recommendation system to decide which reading the user had in mind without asking him/her.

Our naive baseline approach to calculate distances between descriptions and queries is to compute the EUCLIDean distance between the characterisation of the user query and the free text descriptions. The distance is then used to sort all analysed programmes in ascending order.

4.5 A Psychological Approach: Valence and Arousal

In order to reduce the dimensionality of the feature space, we were looking for an exhaustive list of emotions and a mapping from words to basic emotions. COWIE ET AL. (see (Cowie et al., 2001)) provide such a list of 107 emotional attitudes. They are related to German words as follows (we use *adventurous* as an example):

In German, *adventurous* means *abenteuerlich*. Its DORNSEIFF groups are *9.72 Gefahr (danger)*, *10.23 Lächerlich (ridiculous)*, *10.38 Tollkühn (daredevil)*, and *11.26 Einbildung, Wahn (illusion)*. Each of these groups comprises words and sometimes even phrases that are used now to indicate the attitude *adventurous* when they appear in a free text description. Some examples for group 9.72 are: *gefährlich (dangerous)*, *tollkühn (daredevil)*, *Hinterhalt (ambush)*.

To each attitude, COWIE ET AL. assign a position in a two-dimensional diagram (see Fig. 3, for *adventurous* the coordinates are $(4.2, 5.9)$). In his view, any emotional state can be expressed by two values: *valence*, which addresses the quality of an emotion (ranging from very negative over neutral to very positive) and *arousal*, which refers to the (quantitative) activation level of the feeling (ranging from very low to very high).

For the 107 emotional terms and their valence-arousal coordinates (further called VA-coordinates) provided in (Cowie et al., 2001), we searched for the corresponding DORNSEIFF groups as described above.

For each free text, a set of DORNSEIFF groups is computed (see Fig. 2). This set is mapped onto a corresponding set of VA-coordinates as described above. Thus, in addition to the analysis of addressed topics, we get an analysis of emotional attitudes implied by the description.

The transformation of a free text description leads to a geometrical interpretation in terms of VA-coordinates. In this two-dimensional space, it is obvious to apply geometrical distance measures, such as the EUCLIDean distance of two points as a measure of the similiarity of two descriptions.

The orthogonal geometry is distorted by the

pairwise semantic distances of the DORNSEIFF group names: The first step is the identification of those components of the feature vector that cover the "main theme" of a programme description. A triangular semantic-distance matrix is computed for the DORNSEIFF groups occurring at least once. Each group name (a German noun, adjective, or verb) is looked up in GERMANET. We get a set of trees of the group name's synsets as in the example in Fig. 4 for *reign* and *authority*. The distance $\text{dist}(s,t)$ between group s and t is measured by the number of steps from the leaf to the first node in the left tree in Fig. 4 that is also found in the right tree (S: (n) **abstraction**) and from there to the leaf. If there are multiple readings in GERMANET, the maximum number of steps is taken. If two group names don't have a common hyperonym at all, the pair is omitted further on.

For computing the distance of two descriptions, words that occur very frequently in German are ignored. All other words contribute to the position in the VA-space as described above.

The result of this progress is shown of an example description in Fig. 5. The third dimension is the frequency how often a certain VA-coordinate could be found in a description. This data is interpreted as a kind of density distribution (higher frequencies weighting more than lower ones), and the center of gravity is computed as a VA-coordinate. In this way, two descriptions can be compared by computing the EUCLIDean distance of their centers of gravity. Fig. 3 shows an example for the comparison of two descriptions.

Finally, the best proposal for a user query is the description whose center of gravity is closest to that of the query. The onomasiological DORNSEIFF lexicon provides a list of topics or word

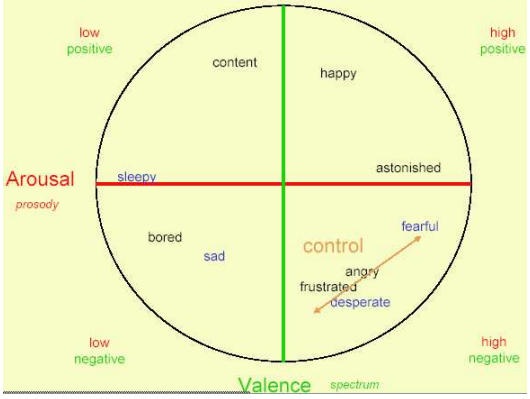


Figure 3: Valence-Arousal diagram

S: (n) reign
S: (n) time period, period of time, period
S: (n) fundamental quantity, fundamental measure par S: (n) measure, quantity, amount
S: (n) **abstraction**
S: (n) abstract entity
S: (n) entity

S: (n) authority
S: (n) authority, authorization, authorisation, potency, dominance, sa y-so
S: (n) control (power to direct or determine)
S: (n) power, powerfulness
S: (n) quality
S: (n) attribute
S: (n) **abstraction**
S: (n) abstract entity
S: (n) entity

Figure 4: Synset trees in WordNet for *reign* and *authority*

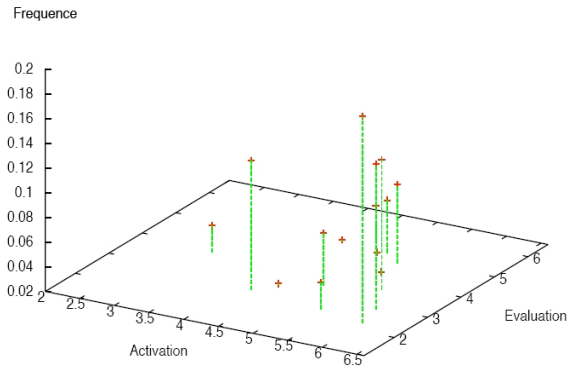


Figure 5: VA-representation of a EPG description

fields and lists words and phrases for them. GERMANET however is semasiological and defines the meaning of words in terms of concept hierarchies. Therefore, we have two independent organizational approaches whose definitions are uncorrelated and allow the integration of two knowledge sources for the retrieval task to be solved.

5 A User-Centered Evaluation

The most important figure to evaluate is how good do proposals match the user query in the user's view. Users rate the quality of the system high, if the system generates suggestions that seem plausible to the user on the basis of how he/she understands the query.

As the system computes distances of feature vectors, but does not solve a classification task, precision/recall figures cannot be computed.

For a first evaluation of the performance from the user's point of view, in a public presentation of the demonstrator system people of different sex, age, education and interest could test the system as long as they wanted to. Then they filled in a questionaire in order to rate how good the programme

descriptions met their expectations and how helpful the recommender was for choosing a TV programme.

60 questionaires are being evaluated at the time of this writing. More than 90 percent of the users said, the proposals were very good, good, or fair, the rest rated them as not matching well or unappropriate. For about 75 percent the system was helpful, and an equal number of persons would buy such a system for a price comparable to a state-of-the-art TV set and recommend the system to a friend. These results are very promising and provide good motivation for further work.

Currently, we are building a simulator system which actually is a Wizard-of-Oz system and a system that computes suggestions on the basis of genre preferences only. In this way, we get two types of baseline systems that help judging the performance of our approach.

6 Conclusions and Future Work

The implemented system operates in real time on an embedded Linux system comparable to a Pentium III (500 MHz). We are currently applying the developed method on an online-help application where relevant paragraphs of a user manual have to be identified. This scenario is better suited for an evaluation of the precision of the search algorithm because it is easier to define what system response constitutes a correct and sound answer.

For textual context a more elaborate model of context is desirable. The DORNSEIFF lexicon provides co-occurrence information for each word in the lexicon. Information about case frames is available as well. Both knowledge sources can be combined to construct a tagger for DORNSEIFF groups that assigns the most probable sequence of DORNSEIFF groups to each sentence in a free text.

In conclusion: Applying the DORNSEIFF lexicon appears to be a successful approach to abstract from a given text. It offers a way to generalization that does not exclusively rely on statistical methods which are the key work horse for shallow processing of unrestricted text.

The evaluation of the user feed back indicates that such an approach is also accepted in typical situations of using text retrieval systems.

References

Liliana Ardissono, F. Portis, P. Torasso, A. Chiarotto, and Angelo Difino. 2001. Architecture of a system for the generation of personalized electronic program guides. In *Proc. UM2001 Workshop on Personalization in Future TV (TV01)*, Sonthofen, July.

Liliana Ardissono, Christina Gena, Pietro Torasso, Fabio Bellifemmine, Angelo Difino, and Barbara Negro. 2004. User modelling and recommendation techniques for personalized electronic program guides. In Liliana Ardissono, Alfred Kobsa, and Mark T. Maybury, editors, *Personalized Digital Television – Targeting Programs to Individual Viewers*, volume 6 of *Human-Computer Interaction Series*, chapter 1, pages 3–26. Springer.

Anna L. Buczak, John Zimmerman, and Kaushal Kurapati. 2002. Personalization: Improving ease-of-use, trust, and accuracy of a tv show recommender. In *Proceedings of the TV'02 workshop on Personalization in TV*, Malaga (Spain).

Kerstin Bücher, Michael Knorr, and Bernd Ludwig. 2002. Anything to clarify? report your parsing ambiguities! In Frank van Harmelen, editor, *Proceedings of the 15th European Conference on Artifical Intelligence*, pages 465–469, Lyon (France), July. IOS Press.

R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J.G. Taylor. 2001. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18:32 – 80, January.

Cristina Gena. 2001. Designing tv viewer stereotypes for an electronic program guide. In *Proc. UM2001 Workshop on Personalization in Future TV (TV01)*, Sonthofen, July.

Thorsten Herfet, Thomas Kirste, and Michael Schnaider. 2001. Embassi – multimodal assistance for infotainment and service infrastructures. *Computers and Graphics*, 25(4):581–592, August.

Thomas Kirste and Thomas Heider. 2002. Supporting goal-based interaction with dynamic intelligent environments. In Frank van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 22–26, Lyon (France), July. IOS Press.

Julia Nitschke and Michael Hellenschmidt. 2003. Design and evaluation of adaptive assistance for the selection of movies. In *Proceedings of IMC 2003 "Assistance, Mobility, Applications"*, Rostock, June.

Jeroen van Barneveld and Mark van Setten. 2004. Designing usable interfaces for tv recommender systems. In Liliana Ardissono, Alfred Kobsa, and Mark T. Maybury, editors, *Personalized Digital Television – Targeting Programs to Individual Viewers*, volume 6 of *Human-Computer Interaction Series*, chapter 1. Springer.

Exploiting Linguistic Resources for building linguistically motivated ontologies in the Semantic Web

Maria Teresa Pazienza, Armando Stellato

AI Research Group, DISP, University of Rome, Tor Vergata
Via del Politecnico 1 00133 ROMA (ITALY)
{pazienza,stellato}@info.uniroma2.it

Abstract

Ontologies provide formal models for representing domain knowledge, which reveal to be useful in several contexts where efficient organization of available data and an shared understanding of its content reveals to be crucial. The Semantic Web offers the most appropriate scenario for exploiting ontologies' potentialities, due to the large amount of information which is to be exposed and accessed. The Semantic Web is however not a controllable and easy to manage knowledge base, and is instead characterized by huge quantities of documents accessed by thousands of users. Though machine readability is a primary demand for automatic exchange of data, several SW services (Intelligent Q&A, Semantic Search Engines etc..) still need to access knowledge expressed in the primary way humans can easily understand it: natural language. Moreover, the role of different cultures and languages is fundamental in a real World aWare Web, so that multilingualism becomes of great interest in this boiling cultural cauldron. These premises suggest that ontologies as we know them now, should be enriched to cover formally expressed conceptual knowledge as well as to expose its content in a linguistically motivated fashion. This paper presents our approach in establishing a framework for semi-automatic linguistic enrichment of ontologies, which led to the development of OntoLing, a plug-in for the popular ontology development tool Protégé. We describe here its features and design aspects which characterize its current release.

1. Introduction

The scenario offered by the SW (and by the Web in general) is however characterized by huge quantities of documents and by users willing to access them. Though machine readability is a primary aim for allowing automatic exchange of data, several SW services like Intelligent Q&A, Semantic Search Engines etc.. still need to understand and expose knowledge expressed in the sole way humans can easily understand it: natural language. Moreover, the role of different cultures and languages is fundamental in a real World aWare Web and, though English is recognized de facto as a "lingua franca" all over the world, much effort must be spent to preserve other idioms expressing different cultures. As a consequence, multilinguality has been cited as one of the six challenges for the Semantic Web (Benjamins et al., 2004). These premises suggest that ontologies as we know them now, should be enriched to cover formally expressed conceptual knowledge as well as to expose its content in a linguistically motivated fashion.

In this paper we introduce our work in establishing a framework for semi-automatic linguistic enrichment of ontologies, which has run through the identification of different categories of linguistic resources and planning their exploitation to augment the linguistic expressivity of ontologies. This effort has lead to the development of OntoLing, a plug-in for the popular ontology editing tool Protégé (Gennari et al., 2003) which allows for linguistic enrichment of ontologies. We describe here the features characterizing its current release and discuss some of the innovations we are planning for the near future. In particular, Section 2 describes the motivations for a linguistically-aware approach to ontology development, and lists the main objectives which guided the development of OntoLing. Section 3 provides some background on linguistic resources, their availability and how they are characterized. Section 4 describes a general interface for accessing the content of these resources, introducing the concept of Linguistic Watermark. In Section 5 we describe the architecture of OntoLing, its functionalities and its adaptive behavior towards different lexical resources. Section 6 describes how linguistic enrichment has been modeled in Protégé and Protégé OWL. Section 7 concludes this document with considerations on the work done so far, adding some hints on future research directions.

2. Ontologies meet language

Ontology Development is a task requiring considerable human involvement and effort, at a large extent with the objective of providing a shareable perspective over domain related knowledge. What "shareable" means, depends on the nature of the task(s) the ontology is thought for. The scenario offered by the Semantic Web is in fact characterized by distributed services which must both realize and rely on a proper connection of machine-accessible formal semantics and more traditional Web content.

For this connection to be true, a complete Ontology Development process should consider the formal aspects of conceptual knowledge representation, as well as guarantee that the same knowledge be recognizable amongst its multiple expressions which are available on real data: that means language.

To achieve such a deeper expressivity, we should reconsider the process of Ontology Development to include the enrichment of semantic content with proper lexical expressions in natural language. Ontology Development tools should reflect this need, supporting users with dedicated interfaces for browsing linguistic resources: these are to be integrated with classic views over knowledge data such as class trees, slot and instance lists, offering a set of functionalities for linguistically enriching concepts and, possibly, for building new ontological knowledge starting from linguistic one.

By considering some of our past experiences (Atzeni et al, 2004, Pazienza et al. 2003, 2005) with knowledge

based applications dealing with concepts and their lexicalizations, a few basic functionalities for browsing linguistic resources (from now on, LRs) emerged to be mandatory:

- Search term definitions (glosses)
- Ask for synonyms
- Separate different sense of the same term
- Explore genus and differentia
- Explore resource-specific semantic relations as well as some others for ontology editing:
- Add synonyms (or translations, for bilingual resources) as additional labels for identifying concepts
- Add glosses to concepts description (documentation)
- Use notions from linguistic resources to create new concepts

While ontologies have undergone a process of standardization which culminated, in 2004, with the promotion of OWL (Dean et al, 2002) as the official ontology language for the semantic web, linguistic resources still maintain heterogeneous formats and follow different models, which make tricky the development of such an interface. The next sections address this problem and discuss our approach in defining the model of OntoLing, the Plug-in for Protégé dedicated to linguistic enrichment of ontologies.

3. Linguistic Resources, an overview

"The term linguistic resources refers to (usually large) sets of language data and descriptions in machine readable form, to be used in building, improving, or evaluating natural language (NL) and speech algorithms or systems" (Cole et al, 1997). Examples of linguistic resources are written and spoken corpora, lexical databases, grammars, treebanks and field notes. In particular, this definition includes lexical databases, bilingual dictionaries and terminologies (which can all be indicated as lexical resources), which may reveal to be necessary in the context of a more linguistic-aware approach to KR. In past years several lexical resources were developed and made accessible (a few for free), and a wide range of resources is now available, ranging from simple word lists to complex MRDs and thesauruses. These resources largely differentiate upon the explicit linguistic information they expose, which may vary in format, content granularity and motivation (linguistic theories, task or system-oriented scope etc…).

Multiple efforts have been spent in the past towards the achievement of a consensus among different theoretical perspectives and systems design approaches. The Text Encoding Initiative [OR5] and the LRE-EAGLES (Expert Advisory Group on Linguistic Engineering Standards) project (Calzolari et al., 1996) are just a few, bearing the objective of making possible the reuse of existing (partial) linguistic resources, promoting the development of new linguistic resources for those languages and domains where they are still not available, and creating cooperative infrastructure to collect, maintain, and disseminate linguistic resources on behalf of the research and development community.

However, at present time, with lack of a standard on existing LRs, it appears evident that desiderata for functionalities which we described in section 2, would depend upon the way these resources had been organized. Often, even a local agreement on the model adopted to describe a given (a series of) resource does not prevent from an incorrect formulation of its content. This is due to the fact that many resources have been initially conceived for humans and not for machines, As an example, on existing available dictionaries words' definitions and synonyms are not always managed the same way: in some cases synonyms are clustered upon the senses which are related to the particular term being examined (among others, Babylon [OR1] and Dict [OR2] dictionaries, where the senses are separated by a ";" symbol), other simply report flat lists of terms without even identifying their different meanings (as for Freelang dictionaries [OR3]). In several dictionaries, synonyms are mixed with extended definitions (glosses) in a unpredictable way and it is not possible to automatically distinguish them. Terms reported as synonyms may sometimes not be truly synonyms of the selected term, but may represent more specific or general concepts (this is the case of Microsoft Word synonymy prompter). Of course, the ones mentioned above represent mere dictionaries not adhering to any particular linguistic model, though they may represent valuable resources on their own.

A much stronger model is offered by Wordnet (Fellbaum, 1998), which, being a structured lexical database, presents a neat distinction between words, senses and glosses, and is characterized by diverse semantic relations like hypernymy/hyponymy, antonymy etc… Though not being originally realized for computational uses, and being built upon a model for the mental lexicon, WordNet has become a valuable resource in the human language technology and artificial intelligence. Due to its vast coverage of English words, WordNet provides with general lexico-semantic information on which open-domain text processing is based. Furthermore, the development of WordNets in several other languages (Vossen, 1998) extends this capability to trans-lingual applications, enabling text mining across languages.

It is impossible to foresee all the features which could be exposed by different resources, from simple word lists to complex multilingual Wordnets: a trade-off must be found, to outline the shape of an interface with sufficient level of generality to be exploited automatically, while leaving space for introducing custom functionalities, to be considered as resource specific services and thus exploited upon discovery.

4. A General Interface for Lexical Resources: The Linguistic Watermark

Along with the analysis of a general interface for linguistic resources, it emerged the logical independence which it could maintain with respect to its possible embedding applications. Our experience pointed out usefulness in diverse natural language related applications like Ontology Mapping, Question&Answering and Information Extraction, where support for multilinguality and a wider linguistic awareness could be, if not necessary, at least useful for improving performances. Moreover, the interface could also act as a sort of unique fingerprint for describing the underlying resource for which access is provided, its information being exploitable in many application-dependant contexts.

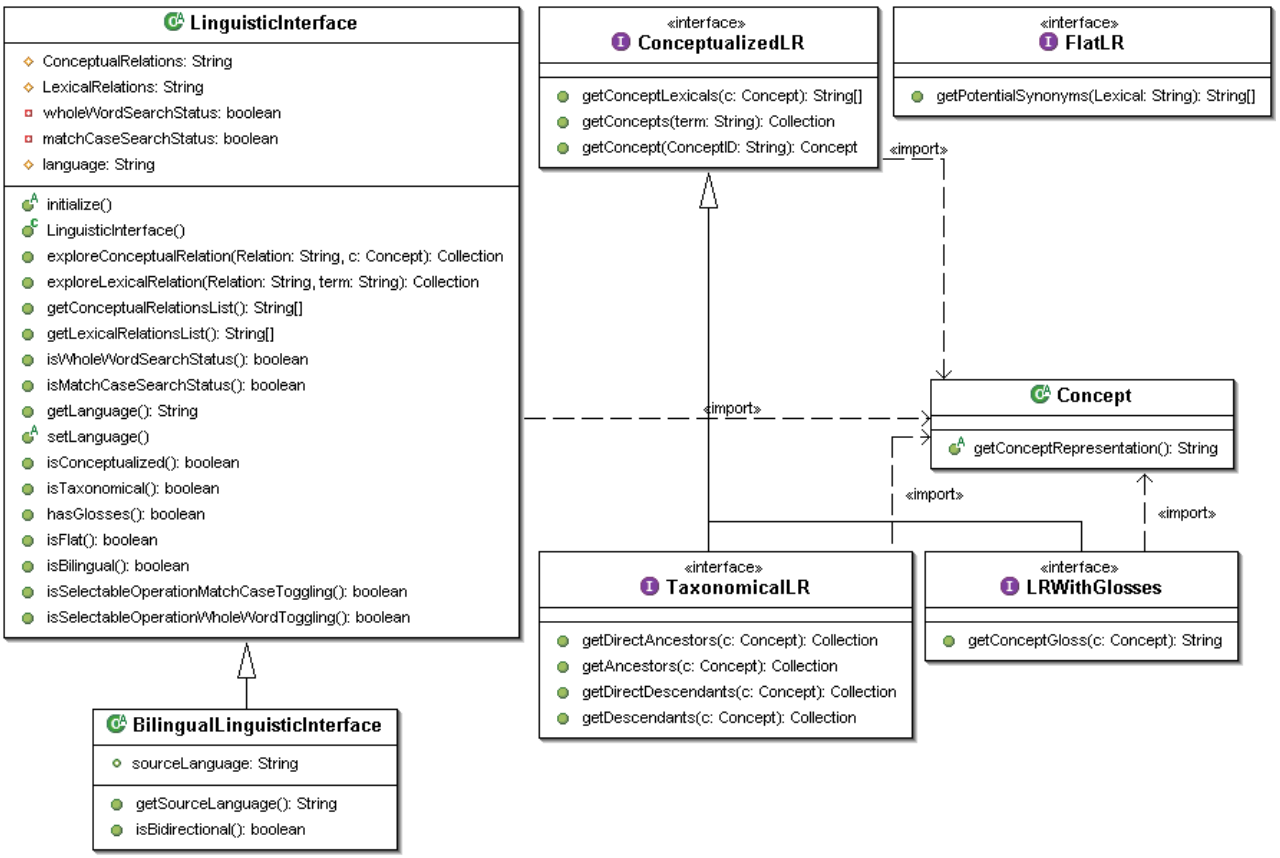


Figure 1: A Class diagram depicting part of Linguistic Watermark classes and interfaces

For this reason, we introduced the notion of Linguistic Watermark, as the series of characteristics and functionalities which distinguish a particular resource inside our framework. As we can observe from the Class Diagram in Fig. 1, we sketched a sort of ontology of linguistic resources, with the addition of operational aspects. Linguistic resources are in fact structured and described in terms of their features and how their lexical information is organized; the ontology has then been completed with query methods for accessing resource's content. We thus implemented this operational ontology as a java package on its own, which can externally be imported by any application willing to exploit natural language resources like lexicons and terminologies. The core of the package is composed of an Abstract Class, named `LinguisticInterface`, which is both the locus for a formal description of a given linguistic resource and a service-provider for exposing the resource specific methods. The other abstract classes and interfaces in the package, which can be implemented or not, depending on the profile of the resource being wrapped, provide instead the signatures for known interface methods.

We have currently developed several implementations of the Linguistic Watermark. Two of them, the WordNet Interface and the last DICT Interface, being related to freely available resources, have been made publicly available on the OntoLing site.

The first one is an almost totally complete implementation of the Linguistic Watermark. The WordNet Interface is in fact a `ConceptualizedLR`, because its linguistic expressions are clustered upon the different senses related to the each term. These senses – "synsets", in WordNet terminology – have been implemented through the Concept interface, which we see bounded by the import statement in the class diagram. WordNet is a `LRWithGlosses`, as glosses are neatly separated from synonyms and organized in a one-to-one relation with synsets. Finally, WordNet Interface implements `TaxonomicalLR`, as its indexed word senses are organized in a taxonomy of more specific/more generic objects.

The other one, DICT Interface, is based on the Dictionary Server Protocol (DICT) [OR2], a TCP transaction based query/response protocol that allows a client to access dictionary definitions from a set of natural language dictionary databases. The DICT interface is conceptualized too, though its word senses are not indexed as in WordNet (that is, it is not possible to correlate senses of two different terms upon the same meaning). DICT Interface is also a `BilingualLinguisticInterface`, as its available word-lists provide translations for several idioms.

Other available interface classes denote Flat resources (as opposed to Conceptualized ones), which contain flat lists of linguistic expressions for each defined term, and `BidirectionalTranslators`, which represent a further specialization of Bilingual Linguistic Interfaces providing bidirectional translation services. Other interfaces (`ApproximateSearchToggling`) are not directly related to the characteristics of the wrapped LR, but to search functionalities which have been provided for it.

As previously mentioned, we defined two classes of methods for browsing LRs: those defined in advance in the interfaces, which can thus be exploited inside automatic processes, and other very specific resource-

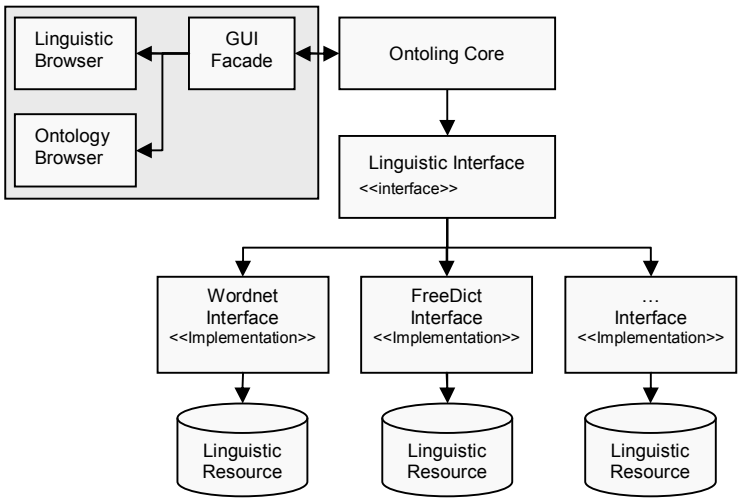


Figure 2: OntoLing Architecture

dependent methods, which are loaded at run-time when the LR is interfaced to some browsing application (e.g. OntoLing). Two methods available in LinguisticInterface: `getLexicalRelationList` and `getConceptualRelationList` act thus as service publishers, the former providing different methods for exploring lexical relations among terms or relating terms to concepts, the latter reporting semantic relations among concepts. Through these methods, the WordNet Interface makes available to the user all the semantic relations contained in WordNet.

5. OntoLing Architecture

The architecture of the Ontoling plugin (see Fig. 2) is based on three main components:

1. the GUI, characterized by the Linguistic Resource browser and the Ontology Enrichment panel
2. the external library Linguistic Watermark, which has been presented in the previous section, providing a model for describing linguistic resources
3. the core system

and an additional external component for accessing a given linguistic resource. This component, which can be loaded at runtime, must implement the classes and interfaces contained in the Linguistic Watermark library, according to the characteristics of the resource which is to be plugged. In the following sections we provide details on the above components.

5.1. OntoLing Core Application

The core component of the architecture is responsible for interpreting the Watermark of linguistic resources and for exposing those functionalities which suit to their profile. Moreover, the behavior of the whole application is dependant on the nature of the loaded resource and is thus defined at run-time. Several methods for querying LRs and for exposing results have been encapsulated into objects inside a dedicated library of behaviors: when a given LR is loaded, the core module parses its Linguistic Watermark and assigns specific method-objects to each GUI event.

With such an approach, the user is provided with a uniform view over diverse and heterogeneous linguistic resources, as they are described in the Linguistic Watermark ontology, and easily learns how to interact with them (thus familiarizing with their peculiarities) by following a policy which is managed by the system.

For example, with a flat resource, a search on a given term will immediately result in a list of (potential) synonyms inside a dedicated box in the GUI; instead, with a conceptualized resource, a list of word senses will appear in a results table at first, then it will be browsed to access synonymical expressions related to the selected sense. Analogous adaptive approaches have been followed for many other aspects of the Linguistic Watermark (mono or bidirectional Bilingual Translators, presence of glosses, Taxonomical structures and so on…) sometimes exploding with combinatorial growth.

Future development of Ontoling will go in the direction of considering supervised techniques for automatic ontology enrichment; selecting and modeling the right strategies for the adopted LRs is another task the core module is in charge for.

5.2. OntoLing User Interface

Once activated, the plug-in displays two main panels, the Linguistic Browser on the left side, and the Ontology Panel on the right side (see Fig. 3).

The Linguistic Browser is responsible for letting the user explore the loaded linguistic resource. Fields and tables for searching the LR and for viewing the results, according to the modalities decided by the core component, are made available. The menu boxes on the left of the Linguistic Browser are filled at run time with the methods for exploring LR specific Lexical and Conceptual relations.

The Ontology Panel, on the right, offers a perspective over ontological data in the classic Protégé style. By right-clicking on a frame (class, slot or instance), the typical editing menu appears, with some further options provided by OntoLing to:

1. search the LR by using the frame name as a key
2. change then name of the selected frame to a term selected from the Linguistic Browser
3. add terms selected from the Linguistic Browser as additional labels for the selected frame
4. add glosses as a description for the selected frame
5. add IDs of senses selected from the linguistic browser as additional labels for the frames

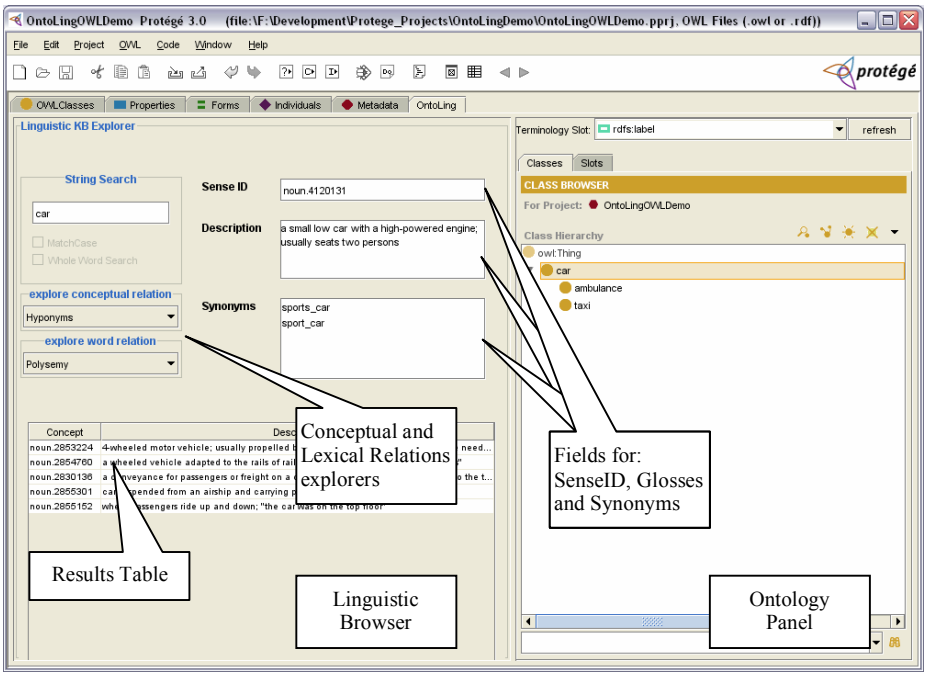


Figure 3: A screenshot of the OntoLing Plug-in

6. create a new frame with a term selected from the Linguistic Browser as frame name (identifier)
7. only in class and slot browser: if the LR is a `TaxonomicalLR`, explore hyponyms (up to a chosen level) of the concept selected on the Linguistic Browser and reproduce the tree on the frame browser, starting from the selected frame, if available

These functionalities allow not only for linguistic enrichment of ontologies, but can be helpful for Ontologists and Knowledge Engineers in creating new ontologies or in improving/modifying existing ones.

How terms and glosses are added to the description of ontologies concepts, depends on the ontology model which is being adopted and is explained in detail in the following section.

6. Using OntoLing with Protégé and Protégé OWL

When a frame-based approach was first adopted in Protégé as a knowledge model for representing ontologies and knowledge bases, no explicit effort was dedicated to the representation of possible alternate labels (synonyms) for concepts neither to support the idea of multilingualism in Ontologies. Frame names were almost as equivalent as IDs, and people were only encouraged, as it is common practice in computer programming when addressing variable names, to adopt "meaningful and expressive names" to denote these IDs. The Protégé model was indeed quite strong and expressive, so that every ontology developer could deal with his linguistic needs at a meta-ontological level and find the right place for them, though no official agreement was yet established.

Later on, with the advent of OWL as a KR standard for the Semantic Web, and with the official release of the Protégé OWL plug-in (Knublauch et al., 2004), things started to converge towards a minimal agreement for the use of language inside ontologies. When we first started working on OntoLing, the OWL plug-in had just been released, and the majority of users continued to use Protégé in the usual way, so we had to find a solution that was quite easy (for the user) to make do with this lack in the standard Protégé model.

To this end, we defined the notion of terminological slot, as a slot which is elected by the user to contain different linguistic expressions for concepts. Any string-typed slot with cardinality set to multiple, can potentially be selected as a terminological slot, and, for easiness of use, OntoLing prompts the user only with this class of slots. This way, to use Ontoling with standard Protégé, a user only needs to define a proper metaclass and metaslot, containing the elected terminological slot; naturally, the same slot can be dedicated to instances at class level. Multilingual ontologies can also be supported by creating different slots and selecting each of them as terminological slots during separate sessions of Linguistic Enrichment, with diverse LRs dedicated to the different chosen languages. Concerning glosses, these can be added to the common "documentation" slot which is part of every frame by default.

Conversely, Linguistic Enrichment of OWL Ontologies follows a more predictable path, thanks to OWL's language dedicated Annotation Properties, such as *rdfs:label* and *owl:comment*. When Ontoling recognizes a loaded ontology as expressed in the OWL language, the terminological slot is set by default (though modifiable) to *rdfs:label*. In this case the *xml:lang* attribute of the label property is automatically filled with the language declared by the Linguistic Interface.

7. Conclusions and future work

As it has been widely described and discussed in the literature on Ontology Development (Noy & McGuinness, 2001, Fernandez et al, 1997), the role of language must not be underestimated. In this work we contributed to the linguistic aspects of ontology development, by identifying functionalities for augmenting the linguistic expressivity of existing ontologies and by implementing these functionalities in the OntoLing Protégé plug-in.

OntoLing, with WordNet as its first exploitable resource, has been adopted by a community of users coming from diverse research areas, from pure linguists approaching ontologies, to ontology developers exploiting specific parts of WorldNet's taxonomical structure as a basis for creating their own domain ontology, up to users needing its main functionalities to enrich ontological concepts of existing ontologies with greater linguistic emphasis. With the recent release of the DICT Interface we added a little step in assisting multilingual ontology development and we now look forward other freely available resources to be added to Ontoling plug-in library: two extensions for MultiWordNet (Pianta et al., 2002) and EuroWordNet (Vossen, 1998) are being developed and will be released in the next months. Moreover, we are currently examining the possibility of extending the interface beyond traditional lexical resources, embracing other type of linguistic resources, such as FrameNet (Baker et al., 1998) and VerbNet (Kipper et al., 2000).

Another explored research direction (see Pazienza & Stellato, 2006) is related to automatization of the process, in order to reduce human effort to a fully supervised methodology for linguistic enrichment of ontologies. We are improving our conceived techniques and testing their quality against real available ontological data, as the results of this specific research will contribute to extend the possibilities offered by the whole framework.

Finally, an important aspect we will address in the future is to better express the relations between ontology and language. Adherence to nowadays standards for ontology representation has been in fact a limit for our research on linguistic enrichment, where a more structured and close bridging between conceptual and linguistic knowledge, with respect to the one we have provided, would be expected. The link we establish in this work between conceptual knowledge and its associated linguistic representation is characterized by simple references between concepts and labels (as offered by the standard *owl:comment* and *rdfs:label* properties), while more sophisticated relationships between lexical entries and ontological objects are required to address the complex conceptualizations which characterize a significant fraction of every ontology.

8. Online Resources

[OR1] Babylon: www.babylon.com
[OR2] DICT: www.dict.org/bin/Dict
[OR3] Freelang: www.freelang.com
[OR4] WordNet: http://www.cogsci.princeton.edu/~wn/
[OR5] Text Encoding Iniziative: www.tei-c.org

9. References

Atzeni, P., Basili, R., Hansen, D. H., Missier, P., Paggio, P., Pazienza, M. T. and Zanzotto, F. M.: Ontology-based question answering in a federation of university sites: the MOSES case study. *9th International Conference on Applications of Natural Language to Information Systems (NLDB'04)* Manchester (United Kingdom), June 2004

Baker, C.F., Fillmore, C.J and Lowe., J.B.: The Berkeley FrameNet project. *In Proceedings of the COLING-ACL*, Montreal, Canada, 1998

Benjamins V. R., Contreras, J., Corcho O. and Gómez-Pérez, A. *Six Challenges for the Semantic Web*. SIGSEMIS Bulletin, April 2004

Calzolari, N., McNaught, J. and Zampolli, A.: *EAGLES Final Report: EAGLES Editors Introduction*. EAG-EB-EI, Pisa, Italy 1996

Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A. and Zue, V. Eds. *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1997

Dean M. and Schreiber, G., Editors. *OWL Web Ontology Language Reference*. W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/. Latest version available at http://www.w3.org/TR/owl-ref/

Fellbaum, C.: *WordNet - An electronic lexical database*. MIT Press, (1998).

Fernandez, M., Gómez-Pérez, A. and Juristo, N. (1997) METHONTOLOGY: From Ontological Art Towards Ontological Engineering, *AAAI-97 Spring Symposium on Ontological Engineering*, Stanford University, March 24-26th

Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubézy, M., Eriksson, H., Noy, N., and Tu, S.: The evolution of Protégé-2000: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.

Kipper, K., Trang Dang, H. and Palmer, M.: Class-Based Construction of a Verb Lexicon. *AAAI-2000 Seventeenth National Conference on Artificial Intelligence*, Austin, TX, July 30 - August 3, 2000

Knublauch, H., Fergerson, R. W., Noy, N. F. and Musen M.A.: The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. *Third International Semantic Web Conference - ISWC 2004*, Hiroshima, Japan. 2004

Noy, N. F., McGuinness, D. L: *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05. March 2001.

Pazienza, M.T. and Stellato, A.: Linguistic Enrichment of Ontologies: a methodological framework. *Second Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex2006)*, held jointly with LREC2006, Magazzini del Cotone Conference Center, Genoa, Italy, 24-26 May 2006

Pazienza, M.T., Stellato, A., Vindigni, M., Valarakos, A. and Karkaletsis, V.: Ontology integration in a multilingual e-retail system. *HCI International 2003*, Crete, Greece, 2003

Pazienza, M. T., Stellato, A., Henriksen, L., Paggio, P., Zanzotto, F. M.: Ontology Mapping to support ontology-based question answering. *Proceedings of the second MEANING workshop*. Trento, Italy, February 2005

Pianta, E., Bentivogli L., & Girardi, C.: MultiWordNet: Developing an aligned multilingual database. *In Proceedings of the 1st International Global WordNet Conference* (pp. 293--302), Mysore, India, January 21-25, 2002

Vossen, P.: EuroWordNet: *A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, Dordrecht, 1998

Metadata Cards for Describing Project Gutenberg Texts

Ronald P. Reck

RRecktek LLC
Chantilly, VA. USA
rreck@rrecktek.com

Abstract

The quantity of data available for linguistic analysis is ever increasing as the Internet expands. However, this is of questionable utility to automated processing when the format of the data is unpredictable. Significant variations can occur, even within a single source. Both data producers and consumers should be able to construct, interpret, and expect a consistently delineated set of metadata for depicting a text-based lexical resource. Standards exist for describing resources but they should be extended in order to support the type and range of information needed for accurate automated processing. Metacards describing the resources would be most beneficial if they extended the existing metadata standards to cover the variation a researcher is likely to encounter. Data producers should be expected to supply enough descriptive information so that a researcher can create the quality of work that others can build upon. This paper describes an effort for creating metacards of Project Gutenberg texts, examples of the variations that occur, and a sample metacard in RDF format.

1. Unpredictable Data is Less Useful

The automated or programmatic processing of corpus data are limited when there are significant or unpredictable variations in the source data. The more data there is, the greater the likelihood of variation, as well as the increased likelihood that the range of variation will cause problems. On the surface it appears that the best solution to the problem would be to have data producers create a consistent and documented data presentation. While consumers would benefit from consistency and documented formats they are not in a position to compel either consistency or documentation. Therefore, until expectations change, researchers need to be prepared to create their own metadata or operate without it.

1.1. Project Gutenberg

A good example of unexpected variation can be seen in the large text archive of freely available text called Project Gutenberg (PG, 2006). Project Gutenberg claims to be "the oldest producer of free ebooks on the Internet" (PG, 2006). PG's ebook collection is an effort produced by hundreds of volunteers. As of January 2006 Project Gutenberg purports to have more than 17,000 books in electronic format most of which are unencumbered by copyright (PG, 2006).

Two distinctly different kinds of metadata are relevant here. *First order metadata* describes information about the archive file itself. First order metadata is structural and might express the number of files in the archive, the archive format, the archive compression ratio, the archive checksum, and similar information describing qualities of the archive file. The second type of metadata, referred to as *second order metadata*, describes specific characteristics of the content such as the author, title, copyright, or editor.

For the purpose of this discussion, both first order metadata and second order metadata are collapsed into a single metadata presentation called a metacard. A more accurate presentation would have created a metacard for the archive that would contain first order metadata and a subsequent metacards would contain second order metadata describing the files contained in the archive. Relating the two metacards could be done with a *contains* relationship in the metacard describing the archive. This level of complexity would increase the precision of the following presentation, but would not add true value to the point of the discussion.

1.1.1. First Order Metadata Problems

Tens of thousands of PG ebooks sound like a treasure trove to a computational linguist until one tries to process them. The first problem a researcher can encounter involves the lack of a mechanism to verify the data integrity of an archive. Without a checksum it is unclear whether the downloaded file is complete. A checksum is a value that is calculated to check data integrity. In situations where the zip file itself is invalid or corrupted it is not clear whether the problem is local to the researcher or if the problem lies in the repository or mirror where the file was retrieved from. If the problem was determined to be a local problem the researcher could merely retrieve the file again. In the latter case the researcher might wonder if the mirror itself is the problem and that other file repositories might contain a valid version of the file. All these questions would be moot if the archive checksum was available in a manner consistent with the tens of thousands of software projects that regularly and adequately deal with this type of problem.

The next challenge a researcher is likely to encounter is the inconsistent directory structure. Some PG archives contain directories whereas others do not. Researchers programmatically dealing with PG archives need to compensate for any possible situation. Perhaps there is a directory in the archive, maybe there is not, maybe there

are multiple directories. Perhaps the directory is named in a manner consistent with the naming of the archive; maybe the directory name is arbitrary. To complicate matters, some archives contain multiple files while other archives contain only a single file. In the end, the usefulness of the archives as a lexical resource would be increased if metadata describing each archive was made available by the producer of the data. As it currently stands, each consumer of the data needs to be apprised of the variations and needs a heuristic to recognize and deal with them when they occur.

In essence, first order metadata problems can be compared to errors of omission in that the difficulty is based on information not being provided.

1.1.2. Second Order Metadata Problems

Once the first order PG problems are successfully navigated, an entirely new type of challenges makes themselves painfully evident. From an analysis standpoint it is often desirable to know characteristics of a file like the author, title or most importantly, the licensing or copyright restrictions of the material. While the more recent PG texts present at least some of this information, thousands of files neglect to include this information or do so in such a variety of ways that the researcher needs to employ another complex heuristic in order to glean even the most basic of information.

For example, among the 434 files released in 1999 by PG, numerous variations (Table 1) were observed for depicting the title of an ebook (PG, 2006).

Title Variations in PG eBooks
Title: A Midsummer Night's Dream
Project Gutenberg's The Three Musketeers
Project Gutenberg's Etext of Tom Swift And His Wizard Camera
Project Gutenberg's Etext of Tom Swift And His Giant Cannon
<p>*Project Gutenberg's Etext of Tom Swift And His Aerial
<pre>Project Gutenberg's Etext of Tom Swift Among The Fire Fighters
The Project Gutenberg Etext of History of England.

Table 1: Title variations in PG ebooks

As these examples illustrate neither the presence of the asterisk at the start of the line, nor the possessive marker, nor the word *title* would be sufficient to consistently determine the text's title. In other words, the inconsistencies create unnecessary challenges for the programmer.

Examples of other variations can include the free variation of words like *Ebook* versus *Etext*, or *preparers* versus *transcribers* versus *producers*. Although, it is possible that these variations actually indicate a distinction, it is also likely that they do not. While it is understandable that variations would occur, consistent metacards created by data producers would help consumers considerably in terms of automatic processing.

2. Metadata Increases Usefulness

If producers made metadata available for lexical resource researchers, then researchers could select to work with data samples that meet their requirements. Thus, metadata would enable the researcher to retrieve data without the added effort to filter it locally with heuristics. Secondly, if unexpected variations occur researchers could verify whether they were the product of an error or intentionally introduced. These are just a couple of the benefits that would be gained by consumers.

2.1. Generating Metadata Programmatically

While consumers would undoubtedly benefit from as much metadata as possible, the ability to determine second order metadata programmatically is quite limited and is often impossible. Second order metadata, as defined here, often requires specific knowledge about the origin or history of a work. Much of the metadata generated in this effort describes only the archive file, or physical qualities about the ebook. Metadata that describes the contents of the file generally came from inside the file itself or is the product of programmatic analysis. Structural similarities between files made it possible to determine many elements of second order metadata once the variability was decomposed. The use of regular expressions made compensating for variations much more manageable than it would otherwise have been.

2.2. Metadata Presentation in Metacards

An ideal solution for presenting metadata is in the form of a metacard. A metacard could be created by data producers and consumers alike. In this analysis a single metacard was created for each of the text based lexical resources using Resource Description Framework (RDF, 2004). Clearly, it would be most beneficial if the data producers created metacards rather than each consumer having to create them on their own.

In the worst case scenario, consumers could download data and then create metacards themselves to facilitate their lexical analysis. Once the metacards were created consumers could then share them with other interested parties throughout the community.

2.3. Existing Metadata Standards

Many metadata standards exist today, but the information they express is not equally important to all consumers. The most important kind of metadata is the type that facilitates programmatic analysis. Once a researcher can determine if the file is intact it is less important to provide metadata that they could create themselves. The second most important type of metadata is one that depicts license restrictions on the content. Of tertiary importance is metadata expressing attributes of

the archive that would otherwise require domain specific knowledge such as the title, author, or genre. This is not as significant of a problem when working with a file at a time since research can often supplement deficient or inaccurate information. Automating a task to analyze a repository such as Project Gutenberg had better employ a robust strategy or it is basically futile.

2.3.1. Dublin Core Metadata Element Set

Dublin Core Metadata Element Set (DCMES) – The Dublin Core Metadata Element Set is comprised of 15 optional elements and entails only a subset of the more encompassing Dublin Core Metadata Initiative (DCMI Usage Board, 2003). Currently, there are two formally endorsed versions of the Dublin Core Metadata Element Set 1.1 (DCMI Usage Board, 2003). They are the ISO Standard 15836-2003 and the NISO Standard Z39.85-2001. These 15 elements are so well considered that the entire set is applicable to the metacard creation effort described by this paper. It is clear this is the product of careful forethought as the creators state, "there are no fundamental restrictions to the types of resources to which Dublin Core metadata can be assigned" (DCMI, 2003).

In this study, four Dublin Core elements were used.

- *dc:title* – used to express the title of an ebook
- *dc:language*[1] – used to express the language an ebook was presented in
- *dc:creator* – used to express the author of an ebook
- *dc:available* – used to express the date an ebook became available in Project Gutenberg in the format YYYY-MM

2.3.2. ISLE Metadata Initiative

The International Standard for Language Engineering (ISLE) Metadata Initiative (IMDI) is a metadata standard proposal for describing multi-media and multi-modal language resources (ISLE, 2006). This proposal recognizes a distinction between top level *catalogue* metadata elements for describing *published corpora* and *session* level metadata elements targeted at describing multi-modal multimedia and written language corpora. Broader in scope than the metacards proposed here, IMDI creators intended to provide metadata for automatic resource discovery as well as *human readable* descriptions.

The IMDI initiative involves a set of proprietary tools that support its use such as the IMDI Editor and IMDI BCBrowser (ISLE, 2006). The ISLE metadata is very expressive and can represent much more detail than the metadata cards described in this paper. IMDI covers some of the same areas as Dublin Core, and where there is overlap between the two standards the Dublin Core version was used.

An interesting element described by IMDI is *CoreMediaFile Type* (ISLE, 2006). This element is encoded as a top-level media type from Multipurpose Internet Mail Extension (MIME) as described in RFC2046 (1996). This element would make a well needed addition to the metacards described here since PG contains several MP3 audio files. Programmatic analysis of audio files was outside of the computationally based scope of the current effort. It was not determined whether this was possible with automated processing but it likely it is.

Special editing tools are intended to support the re-use of existing ISLE metadata transcriptions to create new ones. This time and energy saving feature is clearly intended to support manual creation of metadata. The only elements that are regarded as mandatory are the ones needed for the correct functioning of the tools for working with the metadata descriptions.

This flexibility is likely to encourage adoption and use of ISLE metadata. However, its usefulness is somewhat limited by the lack of clearly articulated requirements.

2.3.3. Open Language Archives Community

The Open Language Archives Community (OLAC) is an international partnership of institutions and individuals who are creating a virtual library of language resources by developing consensus of best current practices while developing a network of interoperable repositories and services for housing and accessing resources (2006). OLAC's strategy tackles two problems at once. It both prescribes a data format for metadata and a repository for storage of that metadata.

The metadata described by OLAC fits into three distinct categories (2006). The first category of OLAC metadata follows the guidelines for embedding Dublin Core in XML. The second category of OLAC metadata uses the *xsi:type* mechanism to access to the full power of XML Schema. This permits the narrowing and restricting of element content. Lastly, OLAC metadata records may use extensions from other namespaces. The process for creating these extensions is well documented and quite accessible to interested parties wishing to create and express metadata in purely XML format.

The effort described here expressed metadata in RDF format for use in RDF aware data stores such as Siderean Software and Oracle 10g. For this reason the metacards are in RDF instead of XML, hence OLAC's metadata approach was not adopted.

2.3.4. Friend of a Friend

The Friend of a Friend (FOAF) project is based around creation of machine readable information about people, groups and companies (2006). The FOAF vocabulary is based on RDF/OWL and is quite straight forward and easy to understand even for humans. This contrasts some other uses of RDF. Three elements from the FOAF project were used in the PG metacards.

- *foaf:name* – used to express the name of the PG text editor/translator/producer
- *foaf:mbox* – used if a producer's email address was provided and if determinable

[1]Our use of *dc:language* uses the three letter country codes of ISO639-2 instead of the two letter country codes of ISO639-1.

- *foaf:sha1*[2] – adopted to express the checksum of the archive file

2.4. Extending Metadata Elements

While clearly, there are a range of useful elements in existing standards, the analysis here has created eleven metadata elements that are not part of any of the cited existing standards applicable to PG lexical resources. They include:

- *charactercount* – A count of the characters in the uncompressed archive. This value is determined through the use of the *wc* command.
- *characterset* – This is a product of the *file* command.
- *cratio* – This element expresses the ratio of the compressed archive to the uncompressed archive and therefore is derivable from two other elements. It comes from the default output of the *zipinfo* command (version 2.40).
- *csize* – The element expresses the number of bytes the compressed archive takes up on disk. It comes from the default output of the *zipinfo* command (version 2.40.
- *etext* – This is the PG number for the text. Each PG text has a unique number.
- *fcount* – This element is the number of files that are contained in the archive as determined by the *zipinfo* command (version 2.40).
- *ftype* – The file type specified by this metadata element comes from the output of the *file* command (version 3.39). This program is believed to exceed the System V Interface Definition of FILE(CMD)2.
- *linecount* – A count of the lines in the uncompressed archive. This value is determined through the use of the *wc* command.
- Producer – This is the PG producer (a.k.a. transcriber, translator, or editor) for the text.
- *ucsize* – The element expresses the number of bytes the uncompressed archive takes up on disk. It comes from the default output of the *zipinfo* command (version 2.40).
- *wordcount* – A count of the words in the uncompressed archive. This value is determined through the use of the *wc* command.

3. Metacard

The metacard format and elements described in this paper were created for 15,511 books of PG. Of those texts there were significant problems with approximately 3 percent of the texts. The problems were caused by incomplete, inconsistent, or incorrect internal metadata, characters outside the range of the current operating system supported character sets corrupted files or archives, non- textual formats such as pictures or audio.

[2]The checksum created in this situation was done using Secure Hash Algorithm 1 (SHA1). The sha1sum 160-bit checksum (as described in FIPS-180-1) was calculated using the 'sha1sum' command that is included with coreutils 4.5.3.

The program for creating the metacards was written using the Perl programming language running under RedHat AS 3.0.

The following table (Table 2) details the size of the PG data set that was analyzed for this project.

Data	Count	Data in MB Compressed	RDF Assertions	Words in Billions
ebooks	15,511	16155	N/A	8.3
metacards	15,022	63	912,806	N/A

Table 2: Data set

To help put the data analyzed into proper perspective, the full metacards required around two hundred megabytes of disk space. The RDF assertions made by the metacards numbered 912 thousand.

3.1. Sample Metacard

In this section, the creation of the sample metacard found in Figure 1 is dissected element by element. In this metacard example as well as the rest of the metacards created for this effort, the first seven lines are referred to as the prologue and establish the namespaces for the tags that follow. Following the prologue section, the *book:Book* tag is a container element that holds the rest of the metacard values. The book:Book element was chosen to facilitate ease in integrating the model in Siderean's Seamark server. In the metacard example provided, the container element name does not have a large significance.

While gathering the information required for the creation of the sample metacard, most of the second order metadata was easily discovered in the first thirty-eight lines of the ebook file. Such elements as *dc:title*, *dc:language*, *dc:creator*, *pg:characterSet*, and *dcterms:available* were found in the ebook with lines starting with the words Title, Language, Author, Character set encoding, and Release Date respectively. In addition, the etext number was discovered on the Release Date line in the PG ebook.

Although the content for the elements mentioned above were fairly straightforward, the content for the content of the *dc:producer* element was somewhat inaccurately placed within the layout of the file. The content for this element was detected on a line following the words *'Transcribed by'*. It would have been more accurate if '*Transcribed by*' preceded the line stating "*** START OF THE PROJECT GUTENBERG EBOOK, A HORSE'S TALE ***" (PG, 2006) since presumably the author, Mark Twain, did not have the assistance of transcriber, David Price. David Price's email address followed his name.

The remaining elements in the *pg* namespace were determined using the commands as explained in section 2.4 titled 'extending metadata elements'.

```
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:book="http://www.siderean.com/ia/ns/bookdemo/"
        xmlns:pg="http://iama.rrecktek.com/daml/ont/pg#"
        xmlns:foaf="http://xmlns.com/foaf/0.1/" >

  <book:Book about="ftp.archive.org/pub/etext/etext97/hrstl10.zip">
       <dc:title>A Horse's Tale</dc:title>
       <dc:language rdf:resource="http://skosaurus.rrecktek.com/ont/language#eng"/>
       <dc:creator rdf:resource="http://skosaurus.rrecktek.com/ont/author#mark_twain"/>
       <pg:characterset rdf:resource="http://skosaurus.rrecktek.com/ont/character_set#US-ASCII"/>
       <dcterms:available rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#date">1997-10</dcterms:available>
       <pg:etext>1086</pg:etext>
       <pg:producer>David Price</pg:producer>
       <foaf:person rdf:parseType="Resource">
          <foaf:name>David Price</foaf:name>
          <foaf:mbox rdf:resource="mailto:ccx074@coventry.ac.uk"/>
       </foaf:person>
       <pg:linecount rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#int">2365</pg:linecount>
       <pg:wordcount rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#int">19257</pg:wordcount>
       <pg:charactercount rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#int">107174</pg:charactercount>
       <foaf:sha1>386126b01230dd062894742701cb208c525471db</foaf:sha1>
       <pg:ftype>ASCII English text, with CRLF line terminators</pg:ftype>
       <pg:fcount rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#int">1</pg:fcount>
       <pg:csize rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#int">44497</pg:csize>
       <pg:ucsize rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#int">109539</pg:ucsize>
       <pg:cratio rdf:datatype="http://www.w3c.org/2000/10/XMLSchema#int">59.4</pg:cratio>
  </book:Book>
</rdf:RDF>
```

Figure 1: Sample Metacard for ebook '*A Horse's Tale*'

3.2. The Range of Metacard Values

In this section, the dataset encompassing the majority of the PG ebooks is characterized by the four characteristics of language, authors, character set, and compression ratio.

3.2.1. Language

Language could only be determined conclusively in 75 percent (11,288) of the 15,022 texts in our data sample. The texts contained content in 25 different languages. The languages translate as follows: 91 percent (10,379) of texts were in English, 4 percent (468) in French, 2 percent (324) in German and the remaining languages were represented in less than 20 files each.

3.2.2. Authors

The 15,022 texts analyzed came from 5,225 different authors. Mark Twain is credited with 132 works, and the second most prolific author was Honore de Balzac with 119 publications. The generic label 'Various authors' was on 7 percent (1,133) of texts, and less than one percent (120) of texts was labeled anonymous. The top 100 authors accounted for 28 percent or 4,184 of the texts.

3.2.3. Character Set

The 15,022 texts had 155 different file type labels. The largest category of character sets was "ASCII English text, with CRLF line terminators" which accounted for 82 percent (12,369) of the texts examined.

3.2.4. Compression Ratio

Out of the 15,022 texts, 93 percent (13,999) of the archive files were compressed between 56 to 66 percent.

4. Conclusion

As explained in the preceding discussion of this project, metadata is invaluable to researchers. Creation of metadata is worthwhile effort for data producers and consumers alike.

When metadata cannot be determined through a programmatic means, discovery or correction of poor, inaccurate or absent metadata can involve significant labour. Lexical researchers can save considerable time and effort if community expectations change to reflect the need for accurate machine readable information depicting lexical resources.

5. Future Direction

The metadata cards described here depict only an initial set of useful metadata that can be generated programmatically. Metadata for text based lexical resources can be extended further to include other information such as the string frequencies for each of the terms in an ebook. Other useful types of metadata could include measurements that indicate the complexity of a written work. Complexity measurements might include a SMOG Index, a Flesch-Kincaid score, a Gunning-Fog Index, or a Coleman-Liau Index. These measurements characterize the understandability for a piece of writing. Measurements of this type lend themselves to a programmatic analysis which could provide a richer understanding of language.

6. Acknowledgements

I would express my gratitude to Olga Lorincz-Reck, Ruth A. Reck, and Kenneth Sall who were all kind enough to lend their support in the writing of this paper. Mike DiLascio of Siderean Software was instrumental in permitting the use of Seamark Server version 4.0 for faceted navigation of the RDF metacards.

7. References

Beckett, D., Miller, E., & Brickley, D. (2002). *Expressing Simple Dublin Core in RDF/XML*. Retrieved from: http://dublincore.org/documents/dcmes-xml/

Darwin, Ian F. (1999). *Manpage for file*. Retrieved from: http://man.he.net/?topic=file&section=all

DCMI Usage Board (2003). *Metadata terms.* Retrieved from: http://dublincore.org/documents/2003/03/04/dcmi-terms/

FIPS 180-1 (1995). *Secure Hash Standard*. Retrieved from: http://www.itl.nist.gov/fipspubs/fip180-1.htm

FOAF (2006). *The friend of a friend project*. Retrieved from: http://www.foaf-project.org/

Free Software foundation (2002). Manpage for wc. Retrieved from: http://man.he.net/?topic=wc&section=all

ISLE (2003). *International Standard for Language Engineering.* Retrieved from: http://www.mpi.nl/ISLE/index.html

ISO 15836 (2003). Available at: http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=37629&ICS1=35&ICS2=240&ICS3=30

ISO 639-1 (2002). Available at: http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=22109&ICS1=1&ICS2=140&ICS3=20

ISO 639-2 (1998). Available at: http://www.loc.gov/standards/iso639-2/langhome.html

OLAC (2006). *Open Language Archives Community.* Retrieved from: http://www.language-archives.org/

Oracle 10*g* (2006). *Oracle Database 10*g *Downloads.* Retrieved from: http://www.oracle.com/technology/software/products/database/oracle10g/index.html

NISO Z39.85 (2001). Available at: http://www.niso.org/standards/resources/Z39-85.pdf

Perl (2006). *Practical Extraction and Report Language.* Retrieved from: http://www.perl.com

PG - Project Gutenberg (2006). *Free eBooks.* Retrieved from: http://www.gutenberg.org/

PG (2006). *A horse's tail.* Retrieved from: ftp.archive.org/pub/etext/etext97/hrstl10.zip

RDF (2004). Resource Description Framework. Retrieved from: http://www.w3.org/RDF/

Redhat (2006). *Red Hat Enterprise Linux AS.* Retrieved from: https://www.redhat.com/rhel/details/servers/

RFC 1766 (1995). *Tags for the Identification of Languages* Retrieved from: http://www.faqs.org/rfcs/rfc2046.html

RFC 2046 (1996). *Multipurpose Internet Mail Extensions Part Two: Media Types* Retrieved from: http://www.faqs.org/rfcs/rfc2046.html

Roelofs, Greg (2002). *Manpage for zipinfo*. Retrieved from: http://man.he.net/?topic=zipinfo&section=all

Siderean Software (2005). Retrieved from: http://www.siderean.com/

Open-class Named Entity Classification in Multiple Domains

Arndt Faulhaber *, **Berenike Loos** †, **Robert Porzel** †, **Rainer Malaka** *

* University of Heidelberg, Germany
arndt.faulhaber@diagnosdata.com
† European Media Laboratory, Heidelberg, Germany
{berenike.loos, robert.porzel}@eml-d.villa-bosch.de
* University of Bremen, Germany
malaka@informatik.uni-bremen.de

Abstract

The exploration of lexical resources has become an important approach to meet the challenge of learning ontologies for the Semantic Web. This paper will introduce the construction and evaluation of a classification system that aims at finding hypernyms for open-class named entities. The final goal will be to integrate these named entities and their corresponding hypernyms as instances and concepts into an ontology. The presented approach employs the web as a source of information. It sends queries to a search engine for a list of named entities and creates a corpus by retrieving the top-ten ranked pages for each named entity. From this corpus a set of sequences of words is extracted which contain the named entity in the middle. Following, we construct vectors that are based on structural information conveyed by the words in the sequences and data extracted from the titles of the web pages. These vectors build the foundation for learning patterns that indicate a correct hypernym of the named entity at hand. To create a supervised learning scenario, all nouns identified in the sequences are annotated as either representing a possible hypernym to a given named entity or not. The evaluation of the presented method shows that its results exceed those of our baseline method.

1. The Challenge

One of the major challenges in creating open-domain spoken dialogue systems is to deal with the Out-Of-Vocabulary (OOV) problem. For instance an automatic speech recognition system (ASR) has to process words, which are not in the lexicon of the speech recognizer (Klakow et al., 1999). In much the same way also natural language understanding systems may encounter unknown words. One reason for this problem is the limited capacity of the speech recognition lexicon (or the decline in performance in Large Vocabulary Speech Recognition (LVSR)).

And another major reason is the dynamic character of natural language that makes complete lexicons impossible. Especially, the continuously changing domain of named entities (NEs), e.g. names of upcoming Hollywood movies or stars, makes a solution to the OOV problem pertinent especially for dialogue and question answering systems. Hence, there is a strong emphasis emerging which seeks to find solutions for phoneme-based recognition (Gallwitz, 2002) as well as for extracting knowledge about these unknown words in order to make them available to class-based lexica or, ultimately, to populate knowledge stores such as ontologies. The latter constitutes the focused application of the work presented herein.

This paper, therefore, considers one central aspect of the solution to the OOV problem, namely, the classification of out-of-vocabulary named entities. In other words our goal is to extract hypernyms[1] for given NEs. Due to the frequent emergence of new NEs the search of hypernyms in standard corpora does not work for many NEs. A simple example would be a NE that did not exist before the creation of the corpus, e.g. the name of a new movie star.

[1] A hypernym is a word that has a more general meaning than a given word, for example "vehicle" is a hypernym of "car". Hypernyms also exist for NEs, for instance "country" is a hypernym of "Germany".

2. Related Work

Approaches to extract hyponym-hypernym relations range from the usage of handcrafted grammatical patterns to extract such knowledge from natural language texts (Hearst, 1992) to modern applications of such patterns, which circumvent their inherent sparse data problem by using the Web as a source of information, demonstrated by Kilgarriff (2001), Evans (2003) and Cimiano (2004). Consequently, machine learning techniques were put to work to either learn patterns or to extract named entities directly from textual sources given an *a priori* training set of patterns or annotated data. To further enhance the precision of such approaches, a variety of methods have been applied including Latent Semantic Analysis, see Cederberg (2003), integrating data repositories like WordNet (Fellbaum, 1998) and gazetteers. Further error reduction was accomplished by combining multiple extraction approaches as shown by Florian et al. (2003). Additionally online-verification of extracted facts and bootstrapping of lexico-semantic patterns was realized in the KnowItAll system (Etzioni, 2005). Our approach allows for hypernyms from an open class. I.e. in contrast to MUC and a lot of other work in the field, we do not have an *a priori* set of classes to which NEs have to be assigned. We permit all words near the NE (that is within a certain distance, a couple of words in front or behind) to be possible hypernyms, which can be virtually any word, and thus gets us an open class. Neither do we build upon an existing Ontology as described in Cimiano (2004).

3. Our Approach

The presented work extends the application of machine learning (ML) methods to the problem of finding hypernyms for a given NE. By evaluating the gain in precision

in comparison to a baseline method, our experiments show that machine learning is a viable approach to solve the introduced task.

Our approach extracts a set of features from the surrounding of a NE (*surrounding* means using a window size of n words to the left and right around the NE) and from the document it occurs in. Applying machine learning methods to this information leads to the recognition of similarities or patterns that indicate the existence of a hypernymic relation between the NE and a word from its surroundings.

Features can be structural information of the words surrounding a NE, e.g., the type of a word, the phrase it belongs to or the distance to the NE. Another feature related to the document as a whole would be the information whether the NE appears in the document's title (based on the assumption that a text whose title contains the NE tends to convey more (important) information about it).

The developed approach works with a corpus that has been created from web pages returned by a query to a search engine. A couple of reasons were decisive for this choice. First of all, the World Wide Web is a vast resource of information. It not only contains encyclopedias like the well-known Encyclopedia Britannica or the fast growing free encyclopedia Wikipedia, it also contains a myriad of glossaries, scientific publications, essays, news articles - uncountable pieces of information, many of which are at the verge of being published at real-time (for example prices for shares or news). Following these arguments, one can assume that many systems will use the Web as their source of information. Consequently, creating a corpus from a set of web pages for this evaluation seems to be an obvious choice and provides a realistic setup.

The method we present aims at performing a task that can be performed by humans quite reliably. I.e. to infer the hypernym of a word, previously unknown, by looking at the surface structures surrounding the word in question, if it is embedded into a context that suggests a meaning (see Section 3.1.). For example, looking at the sentence "We will fly to Paschlumbia" one can rather safely infer that "Paschlumbia" is a location, maybe a country or a town. By gathering multiple examples of how "Paschlumbia" is used in natural language, we can normally solidify one of our hypotheses about its correct meaning. Exactly this circumstance is one of the central aspects of the method presented.

As in most other work in the field of named entity recognition we also employ ML techniques to learn the patterns that indicate hypernymy relationships. However, our approach differs from others: On the one hand we added contextual information to the query to the search engine. Such information can be obtained and used in various ways for natural language processing (Porzel et al., 2005). For example, when searching for a locality, the user's current location can be added as contextual information. This leads to queries such as "Albatros AND Berkeley" where "Albatros" is the NE and "Berkeley" its location. On the other hand we allow for results from any class. Open class means that we do not specify a specific set of target classes.

3.1. Acquiring and Annotating a Corpus

The corpora chosen for a specific task are of fundamental importance. Thus we took special care to the creation process. After assembling a list of 117 NEs and assigning each one a "context word", we decided to create a corpus by retrieving the top-ten web page results of a search engine query for any given NE, i.e. the 117 NEs selected for evaluation (see Section 3.2. for the evaluation and Table 1 for example NEs). The result is a set of 1170 webpages. These HTML-texts are then processed in a series of extraction tasks: Separating natural language from HTML-codes and tokenizing it[2]. The set of Tokens furthermore is split into sentences, which then are tagged by a Part-of-Speech-Tagger (POS-tagger) and furthermore the tokens also get assigned Chunk-Tags[3]. Finally the different sequences containing the examined NE are single out. These sequences build the base for the lexico-semantic analysis of the local surrounding of each NE.

Named Entity	Context
Olympia	Heidelberg
Sayang	Heidelberg
Santa Lucia	Heidelberg
Alraune	Natur (nature)
Michelin	Motorsport (motor sports)
Bianchi	Radsport (cycling)
Bloody Mary	Geschichte (history)
Alpha Centauri	Astronomie (astronomy)
Mount Kailash	Geographie (geography)
Nike	Mythologie (mythology)

Table 1: Examples of Named Entities

Since our goal is to employ machine learning to automate the task of deciding about hyponymic relations, we need to compile and evaluate the feature vectors to be presented to the classifiers. Data to fill the feature vectors originates from the aforementioned series of extraction tasks.
For the evaluation a list is created containing all tokens that are part of a sequence and have been identified to be a noun. That list then constitutes the base for the annotation together with the corresponding NE. To create a gold standard (i.e. the best available answers to the given keys, in this case NEs) for the annotation of the list of hypernym candidates, three annotators reviewed the NE-hypernym candidate pairs.
The corpus includes 7162 markables. A markable in our case is a pair of key and answer candidate (i.e. NE and hypernym candidate), which then could be attributed a true or a false according to the existence of a hyponym-hypernym relation. In case the NE is "Paschlumbia" and three nouns, "country", "region" and "day", are assigned to it, the annotators would be presented three mark-

[2]Here a token is generally a string and should be either a word or a punctuation mark - though if some parsing is inadequate for the input it can be multiple words or other constructs like an e-mail address or similar.

[3]A Chunk, in this case, is a phrase of a sentence, like *noun phrase* or *verbal phrase*.

ables: “Paschlumbia,country”, “Paschlumbia,region” and “Paschlumbia,day”. Afterwards, the annotators discussed the decisions that had not been made unanimously to fix the final assessment and ensuing gold standard.
The user manual defined for the annotation includes not only the intruction that nominative forms are correct hypernyms, but also how to deal with plural, genitive, accusative and dative forms, as well as words with obvious spelling errors. This paper presents in detail the results for a more liberal way of annotation that allows all above mentioned word forms. Similar results have been achieved for a stricter set of rules that only allowed for exact matches, e.g. if the hypernym was “Hotel” then an answer such as “Hotels” was marked as incorrect.[4]
The Kappa coefficient (Carletta, 1996), a measure of agreement, resulted in $\kappa = 0,93$, which states an excellent agreement about the evaluation of the markables.

3.2. Evaluation Setup

The presented results are based on the evaluation of 117 NEs. These can be distinguished into two different sets. One set contains 70 localities, like hotels or restaurants and the other set includes NEs belonging to miscellaneous other domains (some of which are mythology, names of movies or people’s names, see Table 1). This set of NEs has been chosen by a neutral person who is not involved in any part of the development of the presented method (except for the task of assembling the list of NEs).
For the assessment of the ML-based method, the set of NEs has been divided into a training set of 67 NEs and a test set of 50 NEs with a homogeneous distribution of the two classes (localities and miscellaneous). Those sets have been carefully crafted to deliver a similar performance when classified by the baseline. This means that the percentage of hypernyms correctly extracted by the baseline method for the NEs of the training set, resembles the result for the test set.
In the following, a hypernym is considered to be correct if an exact match exists to one of the terms marked as a correct hypernym to a given NE in the annotation data. If for example only “Hotel” is listed as correct hypernym for the named entity “Auerstein”, any hypernym candidate returned but “Hotel” will result in a failure of the test.
Normally the evaluation of named entity recognition tasks calculates precision and recall values and combines them into the commonly employed f-measures as defined by van Rijsbergen (1979). In our case our classifiers assign exactly one hypernym to each NE of our corpus, thus recall is not useful. We will therefore not compute f-measures but concern ourselves with the accuracy of our approach, i.e. precision.

3.3. Baseline

Due to the lack of existing baseline methods to evaluate the performance of our hypernym extraction method, we devised a simple approach that will serve as our baseline throughout this paper. Our baseline method consists of a couple of steps that will be described in the subsequent subsections. For a quick overview:

- regard sequences containing the NE in the middle,
- create a list of nouns (previously classified as such by a Part-of-Speech Tagger) and count their frequency of occurrence in the set of sequences, these nouns are then considered to be *hypernym candidates*,
- group words, where one word begins or ends with the other, and synonyms (as taken from openthesaurus.de[5]) (Section 3.4.),
- calculate sum of occurrences of each word in a group and assign a value to the respective group (Section 3.5.),
- choose a noun from one of the groups as defined by a selection function (Section 3.5.),
- finally verify the selection to the annotation.

3.4. Grouping Hypernym Candidates with Similar Meaning

We propose a grouping of compound words, which is a simple algorithm that verifies whether two words start or end with the same substring and if they do, those words are considered to bear a similar meaning and thus are put into the same group. The idea behind this grouping is derived from the way new words can be created in the German language. If the hypernym in question is “Göttin” (goddess) a possible compound word would be “Siegesgöttin” (goddess of victory). Since Nike is the goddess of victory, she is a goddess in particular. Hence we do not want to distinguish between these two hypernyms but rather consider them together with a higher weight. This heuristic does have some drawbacks, though. Looking at the words “approach” and “roach”, these two words are obviously unrelated in the above described way, and therefore would be grouped erroneously. Our experiment showed, however, that this tends to enhance overall precision rather than degrade it. This is actually the only part of our method, that is actually German specific and does not work for example with English, since the way to build compound words in English differs in such a way that the explained approach would not work. Though with slight modification (considering parts of a compound word, including spaces, as one word) the grouping of compound words would even work for English. An analogous argumentation can be used for another strategy to unite words into groups or groups into bigger groups: The examination of synonyms. Here again if two words are found to convey synonymous meaning (according to a thesaurus) they - or rather the groups they belong to - are joined together. A verification of the feasibility of these heuristics can be found in Section 4.

3.5. Selection Functions

By introducing the kind of grouping presented above, two different kinds of values have been assigned to each hyper-

[4] This evaluation naturally yielded correspondingly lower precision rates for both the baseline and the classification methods, but the relative gain achieved over the baseline was the same as with the liberal annotation. Because of this similarity these results are not included in this paper.

[5] www.openthesaurus.de (last access November 2005) - OpenThesaurus is an Open Source thesaurus for the German language.

nym candidate. One is associated with the group the hypernym candidate belongs to and one to itself. Thus there is more than one strategy to select the "best" candidate and this shall be the purpose of a selection function.
For the baseline we assume two ratings, the rating for each group, σ, and the rating o for each single hypernym candidate itself, which is its frequency of occurrence in the sequences. Let G be a group with elements g, then

$$\sigma(G) = \sum_{g \in G} o(g)$$

is G's *group-rating* and also the *group-rating* of all elements of G.
In the case our approach uses machine learning techniques, the rating of the hypernym candidates changes, because every single occurrence of a hypernym candidate is assigned a separate value by a classifier. So for this situation, we define an additional rating ϕ for a hypernym candidate. $\phi(g)$[6] is the sum of all ratings assigned to the different occurrences of g, g's *candidate-rating*.
Further we define, a normalized value f for each hypernym candidate. To compute f, the ratings assigned by a classifier of each single occurrence of a hypernym candidate are summed up and the resulting sum is then normalized by dividing it through the highest of all group values, $\sigma_{G,max}$. This finally leads to a normalized value f of:

$$f(g) = \frac{\phi(g)}{\sigma_{G,max}} \in [0, 1]$$

Following we present the selection function for the baseline, *Base*, and two different ones for the ML driven approach, *Max* and *Best*.

Base: To choose the "best" hypernym candidate, *Base* selects the hypernym candidate with the highest frequency of occurrence in the set of sequences that belongs to the group with the highest assigned value, i.e. $argmax_{g \in G}(o(g))$, with $G = argmax_{G \in H}(\sigma(G))$ and H being the set of all groups.

Max: The *Max* function ignores grouping. It simply chooses the hypernym candidate that has been assigned the best absolute rating by the classifier.

Best: In contrast to *Max*, *Best* does regard the grouping strategies. Furthermore, the different ratings for each occurrence of the same candidate - assigned by a classifier - will be taken into account. *Best* selects $argmax_g(f(g))$.

These functions select the hypernym candidate that the system returns. The result can either be correct or wrong (it is a hypernym of the NE at hand, or not). The system verifies the result against the gold standard to assess its correctness. Accordingly the precision P can be defined as the ratio of correctly selected hypernyms to the amount of NEs to be evaluated. Let L be the set of NEs for which a hypernym has to be assigned and C $\subseteq$ L be the subset containing all NEs to which a selection function assigned a correct hypernym, then

$$P(L) = \frac{|C|}{|L|}$$

3.6. Features for the Machine Learning Task

A set of features encoding the patterns has to be defined for the ML classifiers so they can actually learn patterns. The presented system considers the following features:

- POS-Tags[7] of all tokens in a sequence surrounding the NE
- Chunk-Tags[8] of all tokens in a sequence including the NE's Chunk-Tag
- The distance of a hypernym candidate from the NE
- Boolean: NE appears in the document's title
- Boolean: hypernym candidate (HC) appears in the document's title
- Boolean: NE is part of a HEARST-Pattern
- Boolean: A word that hints towards co-reference[9] can be found between the NE and a hypernym candidate in the sequence

The first 3 feature types will be referenced as *structural* features in the following.

3.7. Selecting the Learning Algorithm(s)

In an initial test we tested more than a dozen algorithms and variants and selected the most promising ones for further analyzes. The more suitable ones were the following taken from the WEKA (Witten and Frank, 2005) ML library:

- Averaged One-Dependence Estimators (AODE), which averages over various Bayesian learners
- Alternating Decision Tree (ADTree)
- J48, a C4.5 based Decision Tree
- Naïve Bayes tree (NBTree)

The employed ML framework allows for altering the behavior of some of the different classifiers. In those cases we included (some of) the different variations into our evaluation.

4. Results

In the following we present the results of our evaluation. After presenting the increase in performance of the baseline by application of the grouping strategies, we delve into the effects of varying window sizes and compare the resulting precision for the overall best-performing classifier. To give

[6] If g := "pub" occurs twice in the set of sequences and the first occurrence is rated 0.2, the second 0.9 then $\phi(g) = 0.2 + 0.9 = 1.1$.

[7] From the STTS Tag-set with 54 different tags as described in Brants et al. (1999).

[8] From a set of the 8 most common Chunk-Tags.

[9] Words like "this" or "that" often hint towards co-reference, as in "...it is worth to visit the Ritz. That hotel..." - here "That hotel" is co-referential with the previous "Ritz".

a better picture, we also include a graph (Figure 2) showing the precision of a moderately performing Algorithm. Then we take a glance at the influence of the proposed features and finally, we compare the efficiency of different ML methods for the notional "optimal" window size (i.e. the window size for which the best result has been achieved in this evaluation).

4.1. Grouping Feasibility

An evaluation using our corpus shows that the two grouping strategies, grouping compound words and synonyms, increase the performance of the baseline from 33% to 37% and 38% respectively. The combination of the two yields an overall increase to *42% in precision for the baseline method.*[10] Moreover, the grouping of compound words has been assessed by marking out unrelated words from groups, in the same way the annotation (verify Section 3.1.) has been done. The result showed an average error rate of 6.8%.

4.2. Classifiers' Performances

Figure 1 shows the results achieved with AODE for different window sizes, while Figure 2 shows the same setting for the ADTree classifier using 50 boosting iterations. It can be noted that a window size of n=4 returns the best results for the AODE classifier, while the result is significantly different when applying the Alternating Decision Tree, with maxima at n=3 and n=5.

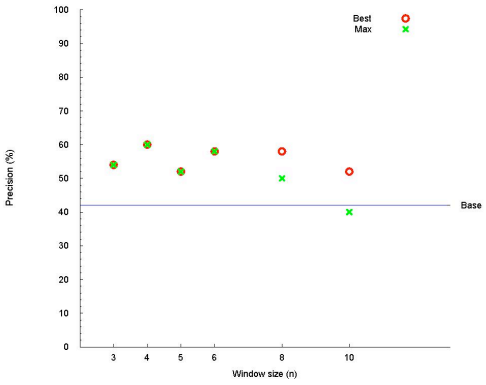


Figure 1: ADOE performance with varying window sizes, all features enabled

A look at the different non-structural features reveals, that only adding the Hearst feature has an influence on the precision of the classification task (verify Figures 3 and 4). Yet, including all 4 features in combination slightly increases the maximum performance, shown in Figure 5.
Our experiments also showed that the AODE ML approach is the best performing machine learning technique we assessed; as depicted in Figure 5, thus leading to the result shown in Table 2.

[10]From here on we calculate precision over the tagged hypernyms, i.e. how many of the selected hypernyms matched those of the gold standard. To include all correctly discarded hypernyms would artificially increase our precision to way over 90%, due to the vast number of nouns, i.e. hypernym candidates, that were in fact not hypernyms.

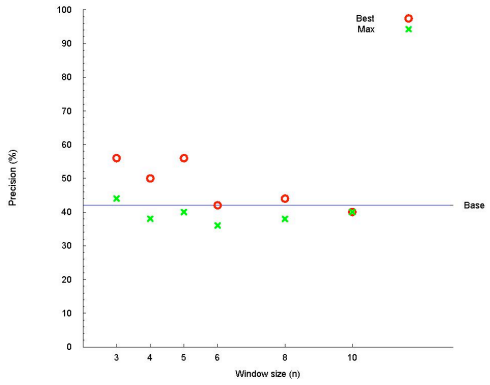


Figure 2: ADTree50 performace with varying window sizes, all features enabled

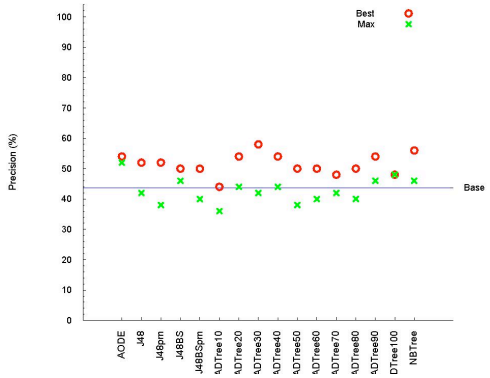


Figure 3: Machine leaning techniques with window size n=4, only structural features

5. Conclusion

In this work we have shown that, the overall performance of open-domain systems (whether spoken dialogue or question answering) can be enhanced by the open-class named entity classification approach described herein. This is especially true if one considers the nature of named entities from an ASR perspective. Here Zipf's law (Zipf, 1949) comes into play which applied for the realm of ASR, means that:

a) short words are tough cookies for phoneme recognition and for longer words phoneme recognition (and phoneme to grapheme mapping) is more accurate,

b) frequent words are usually short and infrequent ones long,

c) rare and unknown (and unknowable) words such as most named entities tend to be long words and stand a good chance to be recognized phonemically in an accurate manner.

We consider a more fine-grained approach for named entity classification, aiming to be more specific than the standard ACE categories - for example, to distinguish between persons such as rock stars and soccer players - necessary for

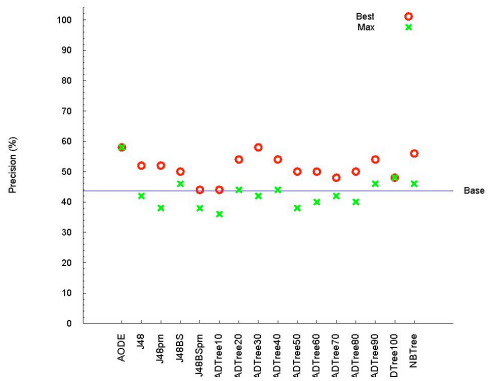


Figure 4: Machine leaning techniques with window size n=4, with structural features and Hearst feature

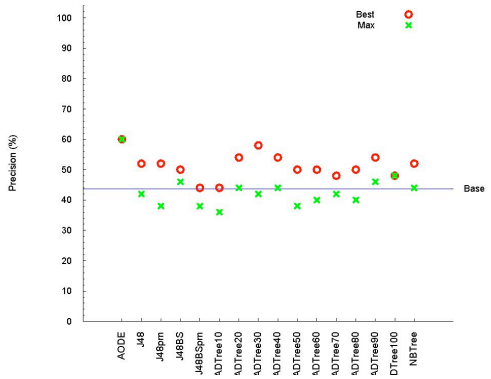


Figure 5: Machine leaning techniques with window size n=4, full set of features

later processing stages which incorporate the tagged entities into ASR lexica or ontologies. We, therefore, see open-domain and open-class NER as a challenging task, which is viable because we stand a realistic chance of obtaining adequate graphemic representations of the OOV words from ASR systems (Zipf) to use as input into our system.

Future work will, of course, be put upon improving precision of our approach, e.g. by means of including valence information of the verbs involved as well as enhancing the inclusion of contextual and domain-specific information, e.g. via automatic domain recognition (Rüggenmann and Gurevych, 2004).

6. References

T. Brants, W. Skut, H. Uszkoreit. 1999. Syntactic annotation of a German newspaper corpus. *In Proceedings of the ATALA Treebank Workshop*.

J. Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22:2, 249-254.

S. Cederberg, D. Widdows. 2003. Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hypernymy Extraction. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL.*

Approach	Precision
Baseline	42%
AODE (n=4)	60%
Gain	18%

Table 2: Performance comparison, our approach used AODE with a window size of n=4

P. Cimiano, S. Staab. 2004. Learning by Googling. *SIGKDD Explorations*.

O. Etzioni et al. 2005. Unsupervised Named-Entity Extraction from the Web: An Experimental Study *Artificial Intelligence 2005*.

R. Evans. 2003. A Framework for Named Entity Recognition in the Open Domain. *Proceedings of Recent Advances in Natural Language Processing*, Borovetz, Bulgaria.

C. Fellbaum (ed.). 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge MA.

R. Florian, A. Ittycheriah, H. Jing, T. Zhang. 2003. Named Entity Recognition through Classifier Combination. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL.*

F. Gallwitz. 2002. *Integrated Stochastic Models for Spontaneous Speech Recognition*. Logos, Berlin.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France.

A. Kilgarriff. 2001. Web as corpus. *Proceedings of Corpus Linguistics*.

D. Klakow, G. Rose and X. Aubert. 1999. OOV-Detection in a Large Vocabulary System Using Automatically Defined Word-Fragments as Filler. In: *Proceedings of EUROSPEECH'99*, Budapest, Hungary, 49-53.

R. Porzel, I. Gurevych and R. Malaka. 2005. In Context: Integrating Domain- and Situation-specific Knowledge. In: Wahlster (ed.) *SmartKom: Foundations of Multimodal Dialogue Systems*, Springer, Heidelberg.

C.J. van Rijsbergen. 1979. *Information Retrieval*. University of Glasgow.

K. Rüggenmann and I. Gurevych. 2004. Assigning Domains to Speech Recognition Hypotheses. *HLT-NAACL 2004 Workshop: Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing*, Boston, MA, 70-77.

I. Witten and E. Frank 2005. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

G. K. Zipf. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, MA, Addison-Wesley.

Interfacing Lexical and Ontological Information in a Multilingual Soccer FrameNet

Thomas Schmidt

International Computer Science Institute
1947 Center Street, Berkeley, CA 94704
Schmidt@icsi.berkeley.edu

Abstract

This paper presents ongoing work on a multilingual (English, French, German) lexical resource of soccer language. The first part describes how lexicographic descriptions based on frame-semantic principles are derived from a partially aligned multilingual corpus of soccer match reports. The remainder of the paper then discusses how different types of ontological knowledge are linked to this resource in order to provide an access structure to the resulting dictionary. It is argued that linking lexical resources and ontologies in such a way provides novel ways to a dictionary user of navigating a domain vocabulary.

1. Introduction

This paper presents ongoing work on a multilingual lexical resource of soccer language – the Soccer FrameNet (SFN). At present, three languages – French, (British) English and German – are taken into account, but the design is potentially open to include additional languages (see also section 1.2). The overall goal is to organize verbs, nouns, adjectives and idiomatic expressions that are used to describe actors, objects and events in and around a soccer match into a lexical network. This network should then serve as an electronic (mono- or bilingual) dictionary to a human user and – potentially – be exploitable by a machine for purposes of natural language processing (e.g. semantic web technology).

The methodological starting point for the development of the lexical resource is frame semantics (Fillmore 1982) and the methodology employed in the construction of the FrameNet lexicon (Ruppenhofer et al. 2005). This basically means that the notion of the semantic frame – "a script-like conceptual structure that describes a particular type of situation, object or event and the participants and props involved in it" (Ruppenhofer et al. 2005) – is used as a fundamental organization principle of the lexicon above the individual linguistic unit. As has been argued, for instance in (Boas 2005), semantic frames can also act as a kind of interlingua for multilingual resources.

The paper is structured as follows: Section 3 gives an overview of the project and illustrates how the basic lexical descriptions of the resource are organised. The following sections then discuss ways of providing additional structure to these lexical descriptions on the basis of ontological-driven principles. Section 5 discusses the assignment of lexical units to a poly-hierarchy of concepts, section 6 shows how arguments of lexical units can be linked to an ontology in the same way, and section 7 introduces the notion of a scenario as an additional ontological structure that can help to organize the lexicon.

2. Related Work

Soccer has been chosen as an exemplary domain in a number of studies related to ontologies as well as in lexicographic research.

Regarding the latter, several contrastive (mostly French-German) analyses of soccer vocabulary have been carried out in the framework of lexicon grammar, most notably by Seelbach (2001, 2002 and 2003). The project presented here differs from that work not only in the choice of languages (English, German and French) and in the basic theoretical approach (frame semantics), but also in its effort to go beyond an exemplary analysis of a small number of examples and instead provide a comprehensive electronic lexical resource which covers a substantial part of the entire soccer vocabulary.

Regarding ontologies, the MUMIS project has constructed a soccer ontology for the purpose of multi-media retrieval of soccer data (Nijholt et al. 2003, Reidsma et al. 2003). Currently, the SMARTWEB project is developing a sports event ontology as a component of a cross lingual, cross media semantic web application for the soccer world cup 2006 in Germany (Buitelaar et al. 2005, Buitelaar et al. 2006). In these projects, the focus is clearly on machine processing of natural language, and the ontologies of these systems consequently play a much more central role than in this project, where the focus is on lexicographic description and ontologies are simply seen as *one* means of organizing such descriptions (see below).

3. Project overview

3.1. Design principles

Although, in constructing the SFN, frame semantics provides the basic methodology for the analysis and the representation of lexical descriptions, there are two reasons not to follow the guidelines for the development of the General Language FrameNet (GLFN) by the book: firstly, the GLFN methodology has been developed with a monolingual lexicon in mind and some requirements that arise only in the construction of a multilingual resource may consequently not have been taken into account.

Secondly, in contrast to the GLFN, the SFN is a domain specific resource. This restriction holds the potential for some methodological alterations. Most importantly, this regards the fact that the number of relevant lexical units will be limited to a comparatively low, finite number (not greater than 1,500 for each language, as a first careful estimate, see also section 3.4) making it possible for the lexicographer to maintain a much more complete and detailed overview of the resource than would be feasible in the general language case. A bottom-up approach to the organization of the lexicon – starting with a "flat" list of LUs and then adding structure to this list – as described in more detail below is greatly facilitated by this fact.

3.2. Some general characteristics of soccer language

All lexical units investigated so far fall into one of the following categories:

- soccer terms: words specifically coined for concepts in soccer, e.g. the noun 'free-kick' or the verb 'to wrong-foot' in English, the noun 'Strafstoß' or the verb 'dribbeln' in German, the noun 'coup de pied arrêté' or the verb 'tacler' in French;
- soccer jargon: words used also in general language, but taking on a distinctively specified meaning when used for talking about soccer, e.g. the noun 'wall' or the verb 'to save' in English, the noun 'Fahrkarte' or the verb 'tunneln' in German, the noun 'petit pont' or the verb 'expulser' in French;
- general language: words frequently used in soccer reports and not having a distinctively different meaning when used outside soccer, e.g. the noun 'victory' or the verb 'to lose' in English.

An obvious characteristic of soccer language, and one that makes it especially interesting for lexicographic purposes, is that it abounds with synonyms. More often than not, one and the same concept can be expressed by more than one lexical item. Consider for instance, the following collection of German verbs each of which can be used to describe that a player overcomes his opponent in a one-on-one challenge:

(1) ausdribbeln, ausspielen, austanzen, austricksen, düpieren, tunneln, umdribbeln, umspielen, verladen, vernaschen, versetzen

Likewise, it is very common in soccer reports to alternate between synonymous nominal and verbal predicates:

(2) Substitute Nilmar **was fouled** by Frank Fahrenhorst just inside the area.[1]
(3) Frank Fahrenhorst **commited a foul** on substitute Nilmar just inside the area.

1 All examples are authentic corpus examples but have been shortened for the purpose of this paper.

3.3. Corpus data

A partially aligned corpus of soccer match reports is used to carry out the lexicographic analysis. The core corpus consists of approximately 500 texts (coming up to around 300,000 words) in each of the languages English, German and French. Around half of these texts are parallel – i.e. they are direct translations of one another –, while the other half consists of comparable texts – i.e. they report the same match but have been written independently of one another. All of the texts have been retrieved from the official website of the UEFA (www.uefa.com). This core corpus is supplemented by additional material from other sources. For German, this comprises match reports from a German soccer journal (www.kicker.de) amounting to roughly 1,000,000 words. For English and French, there are altogether 200,000 more words from other sources. The UEFA website also contains soccer reports in Spanish, Italian, Portuguese, Russian and Japanese. These have also been retrieved in the acquisition process and could potentially be used to supplement the resource for other languages in the future.
All texts have been preprocessed: for the core corpus, this involved tokenizing and sentencizing the text, identifying hyphenated compounds and other automatically detectable multi-word expressions, as well as aligning the parallel portions of the corpus on the paragraph level. All texts are stored in TEI compliant XML.

3.4. Lexicographic data

On the most basic level, the development of the lexical resource consists in finding usages of soccer specific lexical units (like "header", "offside", "to nutmeg", "to defeat") in the corpus, to analyze their argument structure following frame semantic principles, to write a definition that incorporates this argument structure analysis and to annotate a number of example sentences for each unit according to this analysis. The following are examples of resulting LU descriptions for the English noun "cross" the English verb "to dispossess" and the German verb "tunneln":

cross.n

Using a part of his body (ARG4), a player (ARG1) transfers the ball from a source location (ARG2) to a target location (ARG5) on the field in the intention of putting a team-mate (ARG3) in a position to shoot at goal. Typically, the source location of a cross is somewhere near the byline, and its target location is somewhere near the opponent's goal.

Examples:
(1) [Ronaldo]ARG1 delivered a **cross** [from the by-line]ARG2 [for Milan Baros]ARG3
(2) [Jørgensen]ARG1 put over a **cross** [with the outside of his right foot]ARG4 [for Jon Dahl Tomasson]ARG3

(3) [Quaresma]ARG1 swung an inviting **cross** [into the box]ARG5 which was deflected on to Maniche.

Figure 1: Lexical description of the LU 'cross"

dispossess.v

In a one-on-one challenge at a certain location on the field (ARG3), the attacking player (ARG1) manages to take the ball from the player in possession (ARG2).

Examples:

(1) [Benayoun]ARG1 was tripped after **dispossessing** [Costas Kaiafas]ARG2 [on the edge of his own area]ARG3.

(2) On 16 minutes Hungary went close when [Robert Waltner]ARG2 **was dispossessed** [by Maltese goalkeeper Justin Haber]ARG1 at the last gasp.

(3) [Ronaldo]arg1 **dispossessed** [Wisla goalkeeper Radoslaw Majdan]ARG2 [on the edge of the box]ARG3 only for Arkadiusz Glowacki to produce a last-ditch tackle.

(4) PSV's energy and endeavour was enthralling, with [Park]ARG1 typifying their approach by **dispossessing** [Andrea Pirlo]ARG2 [on the centre spot]ARG3 in the 28th minute and releasing countryman Lee Young-Pyo on the left.

Figure 2: Lexical description of the LU 'dispossess'

tunneln.v

In a one-on-one challenge at a certain location on the field (ARG3), the player in possession (ARG1) manages to overcome the attacking player (ARG2) by playing the ball between the latter's legs.

Examples:

(1) [Diogo Rincón]ARG1 **tunnelte** [Paul Freier]ARG2 [im Strafraum]ARG3 und sein Schuss trudelte an Jörg Butt vorbei und landete in Netz.

(2) [Ailton]ARG1 **tunnelte** [Chris]ARG2 [an der Strafraumgrenze]ARG3 und spielte so Klasnic frei.

(3) [Auf der linken Seite]ARG3 geht [der Angreifer]ARG1 auf und davon, **tunnelt** [Lucio]ARG2 und schnibbelt das Leder gekonnt ins rechte untere Eck (44.).

(4) In der 10. Minute **tunnelte** [Arvidsson]ARG1 [den Ex-Bochumer Fahrenhorst]ARG2, verzog aber aus kurzer Distanz.

Figure 3: Lexical description of the LU 'tunneln'

As the following table illustrates, so far (March 2006) more than 1,200 lexical units have been described in this way[2]:

	DE	EN	FR	Total
LUs	554	383	286	1223
Nouns	277	172	142	591
Verbs	263	196	135	594
Examples	2292	1627	1300	5220

Table 1: Lexical units and examples in the soccer frame net

The part of the vocabulary that has been most extensively analyzed so far are words describing individual events during a match (shots, passes, goals etc.). Whereas the resource seems to be relatively complete in this area in so far as the corpus only infrequently uncovers LUs that have not yet been accounted for, other areas of the vocabulary have not yet been analyzed with the same amount of detail. Most importantly, this regards words that speak about a match as a whole (and its place in a competition) and words that denote actors and objects of a match (e.g. goalpost, penalty area, etc.). It is expected that a complete analysis of these areas of vocabulary will at least double the existing number of LUs.

4. Ontologies for lexicographic purposes

Prévot et al. (2005) distinguish three different options for linking ontologies and lexical resources: (1) *restructuring* a computational lexicon on the basis of ontological-driven principles; (2) *populating* an ontology with lexical information and (3) *aligning* an ontology and a lexical resource. In the SFN, the first of these options is explored – my interest in ontologies is mainly concerned with their ability to provide additional layers of structure to a dictionary. From the dictionary user's point of view, these additional layers of structure should provide a means of navigating the vocabulary that goes beyond traditional lexicographic access structures (the two most important of which are alphabetical lists of head words and thesaurus-like groupings of sense related words).

The most straightforward way of linking lexicographic data to an ontology for lexicographic purposes is to assign individual lexical units to specific members of a well-defined system of (possible interrelated) language-neutral concepts. In this way, various types of semantic equivalence between two different lexical units can be expressed.

1) grouping synonymous words: the fact that two lexical units are synonymous can be expressed by assigning them to the same concept in the ontology. For instance, the English nouns "penalty" and "spot-kick" will be mapped to one and the same concept PENALTY_KICK in the ontology.

2) grouping semantically equivalent predicates of different part-of-speech types: the same principle can be applied

2 The fact that German LUs are significantly more numerous than English and French LUs is partly due to the different corpus sizes (see above), but partly also to the tendency of German to form complex compounds that enter as individual LUs into the resource.

also across different part-of-speech categories. For instance, the noun "through-ball" and the verb "to release" both carry the core meaning of "(playing) a long pass such that its recipient can get through on goal". Linking both these lexical units to a concept THROUGH-BALL in the ontology captures this.

3) distinguishing polysemous words: conversely, the polysemy of a given lemma can be captured by assigning the different uses to different concepts in the ontology. Thus, for instance, one use of the French verb "marquer" would be assigned to a concept MARK_PLAYER, while another use would be assigned to a concept SCORE_GOAL.

4) cross-linguistic linking: just like an ontology can be used to capture synonymy within a language, it can also be used as an interlingua for representing translation equivalence across languages. For instance, the fact that the English lexical unit "hat-trick" translates as "Hattrick" into German and as "coup du chapeau" into French can be represented by assigning all three units to a concept HAT_TRICK in the ontology.

Clearly, this way of interfacing lexical data with an ontology covers a substantial part of the information one would expect of a traditional mono- or bilingual dictionary. The ontology, in this case, is simply a language-neutral meta-structure that is used to indirectly capture those relationships that a traditional dictionary would express by direct links between synonymous or otherwise semantically equivalent lexical units. As Alexa et al. (2002) point out, such a manner of proceeding can have great practical value in dictionary creation and maintenance. For instance, with a language-neutral ontology as a backbone to one or several monolingual lexicographic resources, it may become easier for a lexicographer to construct the same resource for an additional language. From the user's point of view, however, these types of links alone do not yet constitute a substantially novel way of working with a dictionary. The next three sections will illustrate ways of interfacing lexical resources with ontologies that may be more innovative in that respect.

5. Poly-hierarchy of concepts

Mapping the lexicographic descriptions exemplified in section 1.3. to a set of ontology concepts as described in section 2 results in a list-like organization of the lexicon as in table 2.

Additional structure is established by organizing concepts into a poly-hierarchy, i.e. by adding links between them that are to be interpreted as an "is_a" relation. For the set of concepts in this example, the most obvious such link is that between SET-PIECE as a superordinate concept of all other concepts - a SET-PIECE is, by definition, the general term for bringing the ball back into play after some kind of interruption. Depending on the type of interruption, this will be a CORNER, a FREE-KICK, a PENALTY etc. Introducing the types of interruption as intermediate concepts yields the hierarchy depicted in figure 4.

Concept	***EN***	***DE***	***FR***
CORNER	*corner*	*Eckball, Ecke, Eckstoß*	*corner c. d. p. de coin*
FREE-KICK	*free-kick*	*Freistoß*	*coup franc*
GOAL-KICK	*goal-kick*	*Abstoß*	*c. d. p. de but*
PENALTY	*penalty spot-kick*	*Elfmeter, Elfer Strafstoß*	*penalty, c. d. p. de réparation*
PUNT	*punt (n), punt(v)*	*Abschlag, abschlagen*	*dégagament*
SET-PIECE	*set-piece dead ball position*	*Standard, Standardsituation ruhender Ball*	*coup de pied arrêté*
THROW_OUT	*throw out*	*Abwurf, abwerfen*	*renvoi de la main*
THROW-IN	*throw-in, throw*	*Einwurf, einwerfen*	*touche*

Table 2: A list of concepts with corresponding lexical units

SET-PIECE
- AFTER FOUL
  - **PENALTY**
  - **FREE-KICK**
- AFTER BALL OFF FIELD
  - AFTER BALL OVER GOAL LINE
    - **CORNER**
    - **GOAL-KICK**
  - AFTER BALL OVER TOUCH LINE
    - **THROW-IN**
- AFTER GOALKEEPER CONTROLS BALL
  - **PUNT**
  - **THROW-OUT**

Figure 4: The SET-PIECE concept and subordinated concepts

With this kind of hierarchy, the dictionary user is given a means of discovering semantically closely related lexical units. For instance, by navigating the hierarchy, he is able to learn that "dead ball position" is a hyperonym of "corner" and that "corner" and "throw-in" are co-hyponyms.

However, this is not the only possible way of organizing the given concepts. Other useful distinctions are:

1) Set-pieces that are carried out by shooting the ball (CORNER, FREE-KICK, GOAL-KICK, PENALTY, PUNT) vs. set-pieces that are carried out by throwing the ball (THROW_OUT, THROW-IN)

2) Set-pieces that are awarded by the referee (CORNER, FREE-KICK, GOAL-KICK, PENALTY, THROW-IN) vs. set-pieces that are not (THROW_OUT, PUNT)

3) Set-pieces that can be conceived as a pass (i.e. that may have a team-mate as a potential recipient: CORNER, FREE-KICK, GOAL-KICK, PUNT, THROW_OUT, THROW-IN) and set-pieces that can be conceived as a shot (i.e. that can be directed directly at goal: PENALTY and, again, FREE-KICK)

Representing these distinctions as additional concepts and adding hierarchical links accordingly should also be helpful to the dictionary user to understand semantic differences and commonalities between the LUs directly associated with the superordinate term "set-piece".

6. Semantically typing arguments

The basic lexicographic building block of the SFN does not only consider the lexical unit itself, but also its arguments (see section 3.4). Consequently, the linking of the lexical resource to an ontology can also be done for the arguments of a predicate. For instance, the three arguments of the LU "to flick on" can be assigned to the concepts PASS and PLAYER in the ontology, as in the following annotated example:

(4) [A diagonal ball from Ioannis Christou]PASS **was flicked on** [by Thomas Makris]PLAYER [to Chloros]PLAYER

Likewise, the arguments of the LU "to award" are assigned to the concepts TEAM, COMPENSATION and OFFENSE in the following annotated example:

(5) On 71 minutes [Terek]TEAM **were awarded** [a penalty]COMPENSATION [after Mariusz Mowlik's handball]OFFENCE, but Khomukha's spot-kick was weak and Piatek easily parried.

As a rule, the concepts suitable to be assigned to an argument of LUs will be more general than the concepts assigned to the LUs themselves, i.e. they will usually be nodes that are relatively high up in the concept hierarchy. In fact, it has been found that the large majority of all arguments can be covered by no more than 25 different concepts, the most common of which are concepts such as PLAYER, BALL, LOCATION, PART_OF_BODY etc.

In terms of dictionary use, these types of links from the lexicographic resource into the ontology offer an important new way of navigating the vocabulary. Consider again the LU "free-kick" which has been assigned to the concept FREE-KICK as described in section 3. The poly-hierarchy of concepts will provide the information that a) a free-kick can be conceived as a kind of PASS and that b) a free-kick is a COMPENSATION awarded by the referee after a foul. Since the arguments of the LUs "to flick on" and "to award" are assigned to the same concepts, a user looking up "free-kick" thus has a simple means of discovering not only the meaning of the term itself, but also of learning about other predicates with which it is used as an argument.

Following the same principle, a lookup of an LU like "goalkeeper" will not only reveal that this is a word used to describe one of the actors of a soccer team, that it is synonymous to the LU "keeper", that "player" is one of its hyperonyms and "defender", "playmaker" etc. its co-hyponyms and that it translates as "Torhüter" into German and "gardien" into French, but also that "to punt", "to punch", "to spill", "to fist", "parade" and "save" are LUs that take an argument of the type GOALKEEPER.

7. Scenarios

The poly-hierarchy of concepts described so far is exclusively concerned with static semantic relations between lexical units. However, a soccer match being a dynamic event unfolding over time, temporal relationships between concepts also play an important role for organizing soccer vocabulary. To describe such temporal relationships, the FN methodology offers the concept of a scenario, i.e. a background description for a sequence of events and transitions. Reidsma et al. (2003), in their ontology-based approach to multimedia information extraction from soccer data, use a similar notion which they call "scene".

In the SFN, a number of prototypical sequences of events in a soccer match have been described as scenarios. These scenarios are all centered around a core event (e.g. a shot) which has a number of participants, and which may be composed of smaller substages. In addition to that, a scenario describes background prerequisites that are necessary for the core event to happen, as well as possible outcomes or following actions. As an example, consider the description of the pass scenario. The main participants in a pass are the passer, the recipient, the ball, a source and a target location on the field, as well as a potentially intervening player (the interceptor) and a potential second recipient. The following diagram illustrates how they take part in a passing event.

The core event of this scenario is lexicalized by LUs such as "to pass", "to center", "through-ball" and "cross", and the arguments of these LUs are linked to the corresponding participants of the scene:

(6) With three minutes remaining [substitute Marcelo Zalayeta]PASSER passed [the ball]BALL [into the middle]TARGET where the unmarked Trezeguet made it 4-1.

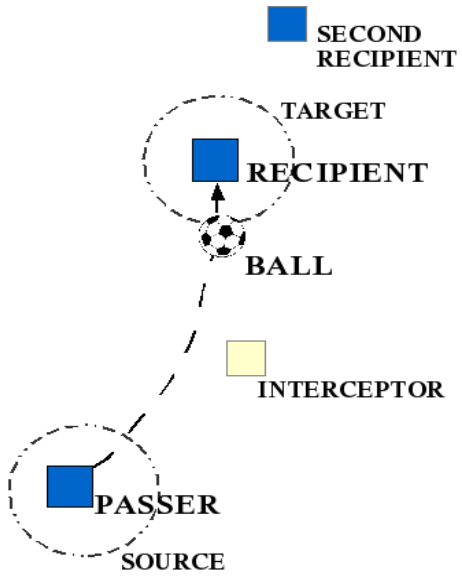


Figure 5: The pass scenario

Note that this assignment is different from the semantic typing of arguments described in the previous section, the difference being basically one between types and roles: whereas the argument "substitute Marcelo Zalayeta" in sentence (6) would be assigned the semantic type PLAYER, a property which holds independently of a specific scene, the assignment of the role PASSER is only valid within this particular passing event.
The same assignment is applied to LUs that do not describe the core event, but a substage or an outcome of it. For instance, one possible outcome of a pass event is that the recipient controls a pass. This is lexicalized by LUs such as "to chest down", "to control", "to fasten on" etc. Linking the arguments of these LUs to the concepts describing the participants in the pass scenario yields annotations of the following type:

(7) [Jeff Whitley]RECIPIENT chested down [a free-kick from Mark Clyde]PASS [at the edge of the box]TARGET
(8) [He]RECIPIENT fastened on [to Shearer's lay-off]PASS [20 metres out]TARGET

Besides controlling a pass, other substages or possible outcomes in the pass scenario include connecting with a pass (LUs: to connect, to meet etc.), missing a pass (LUs: to miss, to miscontrol etc.) flicking on a pass (LUs: to flick on, flick-on) and intercepting a pass (LUs: to intercept, interception etc.). The descriptions of these LUs are linked to the pass scenario in the same way:

(9) [García]RECIPIENT flicked on [Steven Gerrard's set-piece]PASS [for centre-back Hyypiä]SECOND_RECIPIENT
(10) Then [Gert Verheyen]INTERCEPTOR intercepted [a Shakhtar pass]PASS and fed Balaban.

Systematically applying this kind of link between the lexical data and a language neutral description of scenarios provides one further way for the dictionary user to discover semantic relations between lexical items. For instance, starting with a look-up of the LU "pass", the network of links belonging to the pass scenario will take the user to other lexical units describing events that are temporally related to this LU. Since talking about soccer prototypically means lexicalizing sequences of events, this should be of great practical value especially when a dictionary is used actively, i.e. to produce rather than merely to understand a linguistic expression.
Moreover, if the dictionary's task is to help the user to *translate* from one language into another, this kind of information can be crucial in dealing with lexical gaps. Consider, for instance, the following sentence which contains two LUs that would be assigned to different parts of the pass scenario – the verb "to connect" and the noun "cross":

(11) [Bresciano]RECIPIENT missed the target after **connecting** [with [Fabio Simplicio's]PASSER **cross** [from the left]SOURCE]PASS

German does not offer a straightforward translation equivalent for the verb "to connect". However, knowing that the subject of this verb describes the participant RECIPIENT of the pass scenario allows the user to reformulate the sentence by integrating the recipient role as an argument of the LU "Flanke" (which, in turn, is marked as a translation equivalent of "cross" via the concept mapping described in section 2). In that way a translation like the following one might be derived:

(12) [Fabio Simplicio]PASSER schlug eine **Flanke** [von links]SOURCE [auf Bresciano]RECIPIENT. Dieser verfehlte jedoch das Ziel.

8. Summary and Outlook

This paper has presented on-going work on a multilingual lexical resource of soccer language based on frame semantic principles. It has been sketched how different links from the description of lexical units and their arguments into different systems (a poly-hierarchy and a set of scenarios) of language-neutral concepts can act as an access structure to the resulting dictionary, and it has been argued that this kind of access structure provides the user with novel ways of discovering and exploiting semantic relationships between words that traditional dictionaries do usually not cover.
The work as presented here is far from being complete. The next step in the development of the SFN will therefore be to increase the number of lexical units and to supplement the concept hierarchies and scenario descriptions accordingly. Following that, a very important objective will be to develop user interfaces that allow the dictionary user to actually exploit in practice the type of links between lexicographic and ontological data described here.
Concerning the lexicographic side of the work, a more long-term goal is to supplement the corpus data, which at the moment consists entirely of written match reports, with *spoken* data. It is expected that this will not only lead to a substantial number of new lexical units (because spoken soccer language, even more than its written counterpart, is known to be very rich in idiomatic expressions), but also that it will reveal new argument patterns for existing LUs. Furthermore, adding audio data to the lexicographic description of LUs has an obvious didactic value especially for a foreign language user of a dictionary. A number of audio recordings of German radio soccer commentaries have been collected as a first step towards this goal.
Concerning research into ontologies, no concrete steps beyond the ones sketched here are planned for the near future. However, there are some obvious ways in which this work could be related to other studies whose focus is more on natural language processing than on lexicography for human users: Firstly, just like ontologies are constructed here to structure a given set of lexical units, these lexical units could conversely be used to populate existing ontologies with lexical material. This would correspond to the second type of interface between lexical resources and ontologies described by Prévot et al. (2005).

It could be interesting to investigate how well the bottom-up method of collecting LUs and then using an ontology to structure them fits with a top-down method of devising an ontology for a given domain and then "filling" it with lanuage-specific information. Secondly, a more formalized approach to ontology modelling than the one presented here might be a future line of research. The ontologies in the SFN are formulated as simple XML files with pointers into the lexical data, containing no more information than what is directly needed for the lexicographic task at hand. Expressing the same ontologies in a standardized framework, adding rules about concepts and linking concepts to upper ontologies like SUMO could constitute a way of making the knowledge contained in the SFN usable for machine processing purposes.

9. Acknowledgments

The work presented in this paper is being funded by a post-doc grant from the German Academic Exchange Service (DAAD). I am grateful to the Berkeley FrameNet project (Charles Fillmore, Collin Baker, Michael Ellsworth, Josef Ruppenhofer, Miriam Petruck) and its current visitors (Kyoko Ohara, Carlos Subirats, Jan Scheffczyk) for their support.

10. References

Melina Alexa, Bernd Kreissig, Matrina Liepert, Klaus Reichenberger, Lothar Rostek, Karin Rautmann, Werner Scholze-Stubenrecht, and Sabine Stoye. 2002. The Duden Ontology: An Integrated Representation of Lexical and Ontological Information. In *Proceedings of The Ontologies and Lexical Knowledge Bases Workshop (OntoLex02), Las Palmas*.

Hans C. Boas. 2006. Semantic Frames as Interlingual Representations for Multilingual Lexical Databases. In *International Journal of Lexicography 18(4), 445-478.*

Paul Buitelaar, Michael Sintek, and Malte Kiesel. 2005. Feature Representation for Cross-Lingual, Cross-Media Semantic Web Applications. In *Proceedings of the Workshop on Knowledge Markup and Semantic Annotation (SemAnnot2005) at the International Semantic Web Conference, Galway, Ireland, Nov. 2005.*

Paul Buitelaar, Thomas Eigner, Greg Gulrajani, Alexander Schutz, Melanie Siegel, Nicolas Weber, Philipp Cimiano, Günter Ladwig, Matthias Mantel, Honggang Zhu. 2006. Generating and Visualizing a Soccer Knowledge Base. In Frank Keller, Gabor Proszeky (eds.): *Proceedings of the EACL06 Demo Session, 4/2006*

Charles Fillmore. 1982. Frame semantics. In *Linguistics in the Morning Calm*, Seoul, Hanshin Publishing Co., 111-137.

A. Nijholt, R. op den Akker, and F. de Jong. 2003. Language Interpretation and Generation for Football Commentary. University of Twente.

Laurent Prévot, Stefano Borgo, and Alessandro Oltramari. 2005. Interfacing Ontologies and Lexical Resources. In *Proceedings of OntoLex2005, Jeju Island, South Korea.*

Dennis Reidsma, Jan Kuper, Thierry Declerck, Horaccio Saggion, and Hamish Cunningham. 2003. Cross document ontology based information extraction for multimedia retrieval. In *Supplementary proceedings of the ICCS03, July 012003, Dresden, 2003.*

Josef Ruppenhofer, Michael Ellsworth, Miriam Petruck, and Chris Johnson. 2005. FrameNet: Theory and Practice. http://framenet.icsi.berkeley.edu/book/book.html

Horacio Saggion, Hamish Cunningham, Kalina Boncheva, Diana Maynard, Cris Ursu, Oana Hamza, and Yorick Wilks. 2002 Access to Multimedia Information through Multisource and Multilanguage Information Extraction. In B. Andersson, M. Bergholtz, P. Johannesson (eds.): *Natural Language Processing and Information Systems: 6th International Conference on Applications of Natural Language to Information Systems, NLDB 2002, Stockholm, Sweden, June 27-28, 2002*.

Dieter Seelbach. 2001. Das kleine multilinguale Fußball-Lexikon. In Bisang, W. & Schmidt, G. (ed.): *Philologica et Linguistica. Historia, Pluralitas, Universitas. Trier.*

Dieter Seelbach. 2002. La traduction des verbes avec adverbes appropriés et des verbes à particule allemands. In *Traduire au XXIème siècle: Tendances de perspectives, Facultés des lettres UATH Athens.*

Dieter Seelbach. 2003. Separable Partikelverben und Verben mit typischen Adverbialen. Systematische Kontraste Deutsch-Französisch / Französisch-Deutsch. In Seewald-Heeg, Uta (ed..): *Sprachtechnologie für die multilinguale Kommunikation. Beiträge der GLDV-Frühjahrstagung.*