Deriving Mappings for FrameNet Construction from a Parsed Corpus of Japanese

Stephen Wright Horn, Alastair Butler, Iku Nagasaki, Kei Yoshimoto
NINJAL, Hirosaki University, NINJAL, Tohoku University
horn.s.w@ninjal.ac.jp, ajb129@hotmail.com, inaga@ninjal.ac.jp, kei@compling.jp

Abstract

A novel method is introduced for employing a SUSANNE / Penn Historical style parsed corpus to produce FrameNet mapping slots. Target/dependent pairings that are specified for a full range of basic grammatical relations can be generated. These can serve as slots for further specification as frame elements. As outcomes, gains in the speed and accuracy of FrameNet annotation are expected. The information about argument realisation patterns that a fully parsed corpus provides could be used as a basis to choose between senses of a given instantiation of word use, allowing for automated assistance with identifying frames and frame elements, given a sufficiently specified FrameNet. In a parsed corpus, basic local and non-local dependencies (argument/predicate, modifier/head, antecedent/pronoun, etc.) are exhaustively described by associating grammatical relations (dependencies) to specific tree structures. By meshing such an annotation schema with a semantic calculation, a rich range of dependencies can be established without recourse to overt indexing in the source annotation. The semantic calculation derives dependencies from the structure and records them in predicate logic formulas. These logical expressions can then be used to generate derived indices. The end result is a grammar-driven, exhaustive analysis of text into sets of target words and grammatically related dependents. Each set can serve as the co-domain for mapping FrameNet roles onto grammatical structures. The technique yields robust and flexible target/dependent pairings, and can be adapted to many different languages, illustrated here with an example from Japanese.

Keywords: Parsed Corpus, FrameNet, Source Annotation, Derived Annotation, Lexical Semantics

1. Introduction

This paper presents a novel way in which a SUSANNE / Penn Historical style parsed corpus (Sampson 1995, Santorini 2010) can be processed into a form enabling fast and accurate FrameNet annotation. The proposal is instantiated using the NINJAL Parsed Corpus of Modern Japanese (NPCMJ; NINJAL 2016). Corpora of this type define basic grammatical dependencies (argument/predicate, modifier/head, antecedent/pronoun, etc.) as relations within tree structures. Tree structures are defined by labelled nodes, and the relations of precedence and dominance that obtain between those nodes. Within this framework, annotators make exhaustive descriptions of argument realisation. Null elements are employed to mark up both local and non-local dependencies. Grammatical processes such as control, coordination, relativisation, and displacement are also defined. The innovation consists of an automatically derived intermediate analysis that expresses these relations with predicate logic based formulas (relations between predicate/variable bindings), derived as output from a Tarskian style semantic calculation (Tarski and Vaught 1956, Dekker 2012, Butler 2015) that makes reference to the grammatical analysis assigned in the parse.

The approach contrasts with methods that rely on indexing in the source annotation to express abstract relations between nodes, where an index might be a shared mark (e.g., a numeral), or might exist as a value (name) that references the position of an annotation component. Indexing for specifying relationships is applicable to data in any form, and is used in many annotation formats to specify the semantic roles that a constituent holds with respect to some lexical head. Building annotation with indexing is typically costly, in the sense that it often requires a human in the annotation chain to make the critical decision regarding the relation established. Unfortunately, when the namings for indices are motivated by reference to position in a sequence or structure, the dependencies those indices are used to establish can be easily broken by changes in orthography or segmentation, by the introduction of null elements, or by changes in structural assignment. Post-processing texts with such indices is also complicated: There is the need to preserve the motivation of the name of such an index together with the dependency it is meant to encode across changes in sequence or structure.

Our solution to these problems lies in pairing an annotation schema that encodes dependencies as relations in tree structures with a semantic calculation that re-expresses those dependencies as predicate logic bindings (Butler and Horn 2017). Obtaining semantic dependencies relies on the tree structure providing sufficient conditions for identifying grammatical relations, but the definitions of grammatical relations allow for a certain amount of variation in the position and make up of syntactic constituents. In a word, the basis for deriving dependencies is flexible, but the dependencies thus derived (expressed in a structure-independent format) are reliably robust.

The advantages of such a system are many, but its application in annotating semantic roles is particularly noteworthy. Derived indices can be generated from the semantic expressions in a text. These can be shared between a target and a particular dependent (mediated by the appropriate grammatical role) in a post-processing phase. An annotator can associate the appropriate semantic role (e.g., a frame-element in FrameNet annotation) with the index on the target, without the need to establish the pairing by hand. In this way the system supplies an objective and exhaustive basis for assigning semantic roles, where the human labour involved consists of filling an empty slot with a role name. In cases where FrameNet is able to make distinctions between word senses by reference to argument structure,

the exhaustive description of an instantiation of word use in a parsed corpus could conceivably be used to automate the specification of a frame. More generally, embedding a FrameNet analysis in a fully developed description of discourse opens up new avenues of research, arguably multiplying the usefulness of both. For example, independently supported accounts of phenomena such as polysemy and construction meaning could be pursued in a corpus-based program of research.

The remainder of this paper is structured as follows. Section 2. outlines the principles by which basic grammatical relations are defined and shows their instantiation in an example. Section 3. shows how the index-less annotation is subsequently processed to enable a mechanism of semantic calculation that identifies dependencies by reference to structure, but re-expresses them in a structure-independent form. These dependencies can be subsequently assigned to structures through a derived indexing, but don't rely on indexing in order to be established. Section 4. outlines how the system can be harnessed for FrameNet annotation. Section 5. is a summary.

2. Parsed corpus annotation

We illustrate basic annotation principles using the following Japanese sentence:

(1) ehon o kat ta kodomo ga, sore
picture.book ACC buy PAST child NOM this
o oyatsu o tabe nagara yon de i ta.
ACC snacks ACC eat while read GER exist PAST
'The boy who bought the book was reading it while eating snacks.'

Local grammatical relations with respect to a predicate head are encoded through sisterhood under a clause node (IP) in conjuction with tag extensions for grammatical function ("-SBJ" for subject, "-OB1" for direct object, "-ADV" for adverbial, etc.). Consider these with respect to the verb *yon* 'eat' in the context of the matrix clause ("IP-MAT") of the annotation for (1):

(2)
```
( (IP-MAT (PP-SBJ (NP (IP-REL (NP-SBJ *T*)
                              (PP-OB1 (NP;{BOOK} (N ehon))
                                      (P-ROLE o))
                              (VB kat)
                              (AXD ta))
                          (N kodomo))
                    (P-ROLE ga))
          (PU ,)
          (PP-OB1 (NP;{BOOK} (PRO sore))
                  (P-ROLE o))
          (IP-ADV2-SCON (PP-OB1 (NP (N oyatsu))
                                (P-ROLE o))
                        (VB tabe)
                        (P-CONN nagara))
          (VB yon)
          (P-CONN de)
          (VB2 i)
          (AXD ta)
          (PU .))
  (ID example;JP))
```

Every basic grammatical function in the text in (1) is associated with a structural relation in the tree in (2). A relative clause is formed by a typed clause ("IP-REL") containing an index-less trace ("(NP-SBJ *T*)"). These are sufficient to associate the modified head *kodomo* 'child' with the subject argument for *kau* 'buy' in the relative clause by virtue of matching a generalized structural configuration on which the trace/relative head dependency is defined. The *nagara* 'while' clause is specified as subordinated ("-SCON") with a further specification ("-ADV2") requiring a subject-role argument as the antecedent for control (ruling out object *sore* 'this' as a potential antecedent). This is sufficient to establish the noun phrase headed by *kodomo* 'child' as controlling the index-less empty subject position for the verb *tabe* 'eat', again by virtue of matching a generalized structural configuration on which the subject control dependency is defined. Furthermore, sort information ("**;**{BOOK}") has been added to resolve the reference of the pronoun *sore* 'this' as co-valued with *ehon* 'picture.book'. In this way both local dependencies and non-local dependencies are established with a minimum of mark up language, and practically no recourse to overt indices in the annotation.

The above annotation schema is designed to be descriptively adequate for Japanese grammatical phenomena, but SUSANNE / Penn Historical style annotation can be adapted to describe many different languages. For each annotation schema a language-specific conversion can be applied to transform the data into a format that represents grammatical dependencies in a language-independent way.

3. Subsequent interpretation

This section sketches how structural relations are related to rules that interpret those relations as dependencies by a systematic conversion of the data. The conversion takes the form of a number of transformation steps, the first of which (tree normalisation) involves regularising tree structure and reducing the inventory of tag labels. Tag extensions are removed if redundant, or else can have their information contribution off-set. Also, particles marking core grammatical roles are substituted for the grammatical role they mark. Taking the Japanese tree in (2) as an example, the nominative case marker "(P-ROLE ga)" when under "PP-SBJ" is replaced by "(P-ROLE ARG0)", and "-SBJ" is removed. Other changes include collecting the verbal syntagm under one node with off-set information under an "ACT" node to preserve, e.g., tense information. In this way, the normalised tree in (3) is reached:

(3)
```
( (IP-MAT (ACT past)
          (PP (P-ROLE ARG0)
              (NP (CP-REL (IP-SUB (ACT past)
                                  (PP (P-ROLE ARG0)
                                      (NP *T*))
                                  (PP (P-ROLE ARG1)
                                      (NP (SORT *BOOK*)
                                          (N ehon)))
                                  (VB kat_ta)))
                  (N kodomo)))
          (PU ,)
          (PP (P-ROLE ARG1)
              (NP (SORT *BOOK*)
                  (PRO sore)))
          (PP (SORT *SITUATION*)
              (SCON *)
              (P nagara)
              (IP-ADV2 (PP (P-ROLE ARG1)
                           (NP (N oyatsu)))
                       (VB tabe)))
          (VB yon_de_i_ta)
          (PU .))
  (ID example;JP))
```

The second step is to convert the normalised tree into an expression that can serve as input to a semantic calculation system, specifically, the Scope Control Theory (SCT) system of Butler (2015). The normalized tree serves as input to a script that converts the data into an SCT expression, in which, for example, common nouns are treated as predicates taking "entity" variable bindings, verbs are treated as predicates taking "event" variable bindings, etc. Structure in SCT expressions is built exploiting normalized tree structure by locating any complement for the phrase head to scope over, adding modifiers as elements that scope above the head, and keeping track of the binding names (e.g., "`"ARG0"`" (logical subject)) for the resulting SCT expression. Conversion adds construction information from the constituent nodes (e.g, "`subord`" (subordinate clause), and "`Lam ("h", "T", ...)`" (an instruction to make the open "`"h"`" binding (the head binding internal to a noun phrase) into a "`"T"`" binding (the trace binding internal to a relative clause)), etc.). Conversion also adds instructions (e.g., "`gen "EVENT"`") to generate what will become bound variables of a resulting semantic calculation.

The overall output from conversion to an SCT expression for (1) is in (4) below:

(4)

```
val sent =
( fn fh =>
  ( fn lc =>
    ( ( fn lc =>
        ( some lc fh ".e" ( gen "ENTITY")
          ( scon fh "&"
            ( Lam ( "h", "T",
                subord lc nil
                ( ( fn lc =>
                    ( ( arg "T") "ARG0"
                      ( ( fn lc =>
                          ( some lc fh ".e" ( gen "BOOK")
                            ( nn lc "ehon")))
                        [ "ARG0", "ARG1", "h"] "ARG1"
                        ( past ".event"
                          ( verb lc ".event"
                            ["ARG1", "ARG0"] "kat_ta"
                            ( gen "EVENT"))))))
                  [ "ARG0", "ARG1"])))
            ( nn lc "kodomo"))))
      [ "ARG0", "ARG1", "h"] "ARG0"
      ( ( pro ["*"] [ "BOOK"] ".e" "sore" ( gen "BOOK")) "ARG1"
        ( scon fh "SCON_nagara"
          ( control2 lc
            ( ( fn lc =>
                ( ( fn lc =>
                    ( some lc fh ".e" ( gen "ENTITY")
                      ( nn lc "oyatsu")))
                  [ "ARG0", "ARG1", "h"] "ARG1"
                  ( verb lc ".event" ["ARG1"] "tabe"
                    ( gen "EVENT"))))
              [ "ARG0", "ARG1"]))
          ( past ".event"
            ( verb lc ".event" ["ARG1", "ARG0"] "yon_de_i_ta"
              ( gen "EVENT")))))))
  [ "ARG0", "ARG1"])
[ ".e", ".event"]
```

Following an evaluation of the above SCT expression, the predicate logic representation with sorted variables in (5) below is returned:

(5)

```
exists BOOK[4] EVENT[3] EVENT[6] EVENT[7] BOOK[2] ENTITY[5]
 ENTITY[1].(
    ehon(BOOK[2]) & kat_ta(EVENT[3],ENTITY[1],BOOK[2])
  & kodomo(ENTITY[1]) & BOOK[4] = BOOK[2]
  & oyatsu(ENTITY[5]) & past(EVENT[3])
  & past(EVENT[7]) & (tabe(EVENT[6],ENTITY[1],ENTITY[5])
    SCON_nagara yon_de_i_ta(EVENT[7],ENTITY[1],BOOK[4])))
```

Note how the predicate logic representation expresses the subjecthood of *kodomo* ("`ENTITY[1]`") and the objecthood of *ehon* ("`BOOK[2]`") and the action of buying ("`EVENT[3]`") as variables bound by the predicate `kat_ta`, even though *kodomo* does not appear locally with the verb. Identity between the book (*ehon*) that was bought and the pronoun (*sore*) standing in for the thing that was read is expressed as equality between two variables: "`BOOK[4] = BOOK[2]`". And control from the subject of the upstairs verb `yon_de_i_ta` (*was reading*) into the clause headed by `tabe` (*eating*) is captured by the way that "`ENTITY[1]`" is a bound argument of both predicates.

To illustrate the flexibility of the combination of structural definitions for grammatical relations and their rendering into predicate logic representations, consider a frame in which the verb *kau* is used in a sense including a beneficiary (e.g., *Boku wa otooto ni ehon o katta* "I bought a book for my little brother"). Recognising this sense, an annotator adds a null indirect object pronoun `(NP-OB2 *pro*)` as the first constituent in the relative clause in (2), thereby shifting the position of every other element in the tree. Notwithstanding, the definitions for subject, object, etc., still obtain, and these dependencies remain intact.

So far we have seen the automatic calculation of dependencies through structural assignments and their expression in structure-independent representations. All relations expressible in a well-formed parse are defined under the calculation. The accuracy of the calculation is directly related to the accuracy of the sourced parsed annotation. Using another automated process, a derived analysis such as that in (5) can be embedded back into the source phrase structure tree annotation to yield the tree in (6) below:

(6)

```
( (IP-MAT;@0:66
    (PP-SBJ;<0:22>;@0:22
      (NP;@0:19
        (IP-REL;<0:12>;@0:12
          (PP-OB1;<0:5>;@0:5 (NP;{BOOK};@0:3 (N;@0:3 ehon))
                             (P-ROLE;@5:5 o))
          (VB;<,0:5@ARG1,14:19@ARG0,EVENT[3]@EVENT,>;@7:9 kat)
          (AXD;@11:12 ta))
        (N;<,0:12@REL,0:22@h,>;<14:19>;@14:19 kodomo))
      (P-ROLE;@21:22 ga))
    (PU;@24:24 ,)
    (PP-OB1;<26:31>;@26:31
      (NP;{,0:5,};{BOOK};@26:29 (PRO;@26:29 sore))
                                (P-ROLE;@31:31 o))
    (IP-ADV2-SCON;@33:52
      (PP-OB1;<33:40>;@33:40 (NP;@33:38 (N;@33:38 oyatsu))
                             (P-ROLE;@40:40 o))
      (VB;<,33:40@ARG1,0:22@ARG0,EVENT[7]@EVENT,>;@42:45 tabe)
      (P-CONN;@47:52 nagara))
    (VB;<,SITUATION[5]@LINK,26:31@ARG1,0:22@ARG0,
         EVENT[8]@EVENT,>;@54:56 yon)
    (P-CONN;@58:59 de)
    (VB2;@61:61 i)
    (AXD;@63:64 ta)
    (PU;@66:66 .))
  (ID example;JP))
```

This "indexed" view gives a view of the tree structure with nodes dominating a constituent given a suffix "@n:m" where n is the n-th character of the overall tree yield (the collected terminal strings (words) retaining linear ordering and with word separations counting as single characters) marking the first character resulting from a yield of the constituent, while m is the m-th character of the tree yield marking the last character resulting from a yield of the constituent. In addition, indexing information is given to specify argument relationships and antecedence relationships having the form "m:n", with the same 'yield-span' use of n and m as just described. The indexing information gives explicit indexing of grammatical dependencies that the original annotation had left implicit. Specifically the indexing makes the following contributions:

- Indexing given the form "<n:m>" marks a yield-span that serves as an argument for a predicate, as well as providing an antecedent for anaphoric reference.
- The arguments that a predicate takes are marked on the pre-terminal node for the predicate with a "<,...,n:m@role,..,>" format, with "n:m" providing information to locate the argument and "role" stating the argument role.
- Pronominal information is presented with the format "{,...,n:m,...,}", that is, specifying potentially multiple antecedents.

Also note how the trace "(NP-SBJ *T*)", as a zero element that would merely duplicate information now captured by the indexing, has been removed from the tree, thereby removing it from the calculation of the tree yield. With the targets for dependencies spelled out in the predicate nodes, this is now a basis for deriving as output the kind of formatted annotation seen with FrameNet.

4. FrameNet annotation

The immediate utility of this approach is apparent when one considers that FrameNet generalizes over collocations in which dependents are normally related to their targets through grammatical relations. FrameNet (Ruppenhofer et al 2016) uses role labeling annotation to anchor semantic frames to instantiations in natural language. In FrameNet annotation, a frame-element relates to a frame that subsumes multiple predicates with various manifestations for frame-specific semantic roles. Predicates and their arguments are anchored to the character string of the source data through a FrameNet report. To demonstrate how target/dependent pairs identified by a semantic calculation on the grammar can potentially be transformed into frame/frame-element pairs in the FrameNet XML annotation format, consider (7) below, which is the output from the pipeline described here, given the data in (1) as input. (7) is an underspecified FrameNet report generated directly from the tree in (6) with yield-span index information. The FrameNet information that remains to be added is represented by attributes with numbered blanks: "attribute="_n_"".

(7)

```
<sentence>
  <text>ehon o kat ta kodomo ga , sore o oyatsu o tabe
        nagara yon de i ta .</text>
  <annotationSet luID="_1_" luName="kat.v" frameID="_2_"
                 frameName="_3_">
    <layer rank="1" name="Target">
      <label end="9" start="7" name="_4_"/>
    </layer>
    <layer rank="1" name="FE">
      <label end="19" start="14" name="_5_"/>
      <label end="5" start="0" name="_6_"/>
    </layer>
  </annotationSet>
  <annotationSet luID="_7_" luName="kodomo.n" frameID="_8_"
                 frameName="_9_">
    <layer rank="1" name="Target">
      <label end="19" start="14" name="_10_"/>
    </layer>
    <layer rank="1" name="FE">
      <label end="12" start="0" name="_11_"/>
    </layer>
  </annotationSet>
  <annotationSet luID="_12_" luName="tabe.v" frameID="_13_"
                 frameName="_14_">
    <layer rank="1" name="Target">
      <label end="45" start="42" name="_15_"/>
    </layer>
    <layer rank="1" name="FE">
      <label end="22" start="0" name="_16_"/>
      <label end="40" start="33" name="_17_"/>
    </layer>
  </annotationSet>
  <annotationSet luID="_18_" luName="yon.v" frameID="_19_"
                 frameName="_20_">
    <layer rank="1" name="Target">
      <label end="56" start="54" name="_21_"/>
    </layer>
    <layer rank="1" name="FE">
      <label end="22" start="0" name="_22_"/>
      <label end="31" start="26" name="_23_"/>
    </layer>
  </annotationSet>
</sentence>
```

To spell it out in more detail, the FrameNet format consists of source character data as the <text> content, followed by annotations for the predicates as <annotationSet> content. Predicates are picked out with start and end attributes for a Target. For example, the Target for the annotationSet with "luName="kat.v"" is the 8th ("start="7"") to 10th ("end="9"") characters of the text content, namely, "kat". Arguments are similarly established as spans of characters of the source string. For example, the element corresponding to the character span kodomo (identified by "start="14"" and "end="19"") is specified as having a role with respect to "luName="kat.v"". The role would fill the blank in "name="_5_"".

The annotation method we propose involves adding FrameNet information to target/dependent sets in the tree structures themselves. Such information can be added as the terminal strings of offset nodes which are adjacent to the target element and include information to pinpoint the ID of the relevant lexical unit—from which all frame details are recoverable—as well as the frame-elements applicable to target/dependent sets linked to the tree structure via the relevant grammatical function information. Note, for example, that a blank "*_5_*" appears with an "ARG0" node, which corresponds to a subject grammatical role in the tree (8) below.

(8)

```
( (IP-MAT (PP-SBJ (NP (IP-REL (NP-SBJ *T*)
                              (PP-OB1 (NP;{BOOK} (N ehon))
                                      (P-ROLE o))
                              (VB kat)
                              (FRAME (LU *_1_*)
                                     (ARG0 *_5_*)
                                     (ARG1 *_6_*))
                              (AXD ta))
                      (N kodomo))
                (P-ROLE ga))
          (PU ,)
          (PP-OB1 (NP;{BOOK} (PRO sore))
                  (P-ROLE o))
          (IP-ADV2-SCON (PP-OB1 (NP (N oyatsu))
                                (P-ROLE o))
                        (VB tabe)
                        (FRAME (LU *_12_*)
                               (ARG0 *_16_*)
                               (ARG1 *_17_*))
                        (P-CONN nagara))
          (VB yon)
          (FRAME (LU *_18_*)
                 (ARG0 *_22_*)
                 (ARG1 *_23_*))
          (P-CONN de)
          (VB2 i)
          (AXD ta)
          (PU .))
  (ID example;JP))
```

A frame-element value added to the blank "*_5_*" fills the appropriate place in an output transformed to FrameNet format. Given tree structures with frame information added *in situ*, completed FrameNet reports could be produced automatically. The immediate benefits include being able to refer to an exhaustive grammatical analysis in the process of assigning frame-elements, and being able to take advantage of robust dependent/target links that have been established in advance. The ability to make changes in structural assignment and segmentation (within the parameters of the definitions in the syntactic annotation) without the danger of interrupting dependencies is an added advantage.

Using a parsed corpus as source data for FrameNet annotation increases in value proportional to the richness and accuracy of the source parse. One precedent for such an undertaking is the SALSA project (Burchardt et al 2006). Embedding FrameNet information into a well-articulated description of discourse could open up new avenues for research. We propose that such an undertaking would also enjoy increased productivity and accuracy if integrated into systems such as those being developed in tandem with SCT. Extending the application, argument structure profiles for specific instantiations of words in the parsed data could be used to filter appropriate candidates for frame assignments, further reducing the burden on human annotators.

5. Summary

To sum up, the proposal is to take advantage of a technique of annotation that shifts the role of indexing onto the assignment of structural positions in a syntactic tree, and supplies an interpretive process that creates the specifications of dependencies. Language specific source annotation is expressed in a language independent form as the result of a semantic calculation. So far the system has been applied for obtaining valence frames in English[1], Contemporary Japanese[2], and Old Japanese[3]. The system can be used to generate sets of target/dependent pairings that directly correspond to sets of frame/frame-element pairings in FrameNet analyses. We show how assignments of values for frame/frame-element pairings can be added directly to tree structures, and how the resulting tree structures can be processed to give outputs as FrameNet reports.

[1] http://www.compling.jp/ajb129/tspc.html

[2] http://npcmj.ninjal.ac.jp/interfaces/index_en.html

[3] http://www.compling.jp/m97pc

Acknowledgements

This paper describes work funded as a component of the NINJAL Parsed Corpus of Modern Japanese. We thank all project members for their help and advice. This paper benefitted from the comments of three anonymous reviewers, who we gratefully acknowledge.

6. Bibliographical References

A. Burchardt, K. Erk, A. Frank, A. Kowalski, S. Padó, and M. Pinkal. 2006. The SALSA Corpus: a German Corpus Resource for Lexical Semantics. In *Proceedings of the 5th Language Resources and Evaluation Conference (LREC 2006)*. Genoa, Italy.

A. Butler. 2015. *Linguistic Expressions and Semantic Processing: A Practical Approach*. Heidelberg: Springer-Verlag.

A. Butler and S.W. Horn. 2017. Annotating syntax and lexical semantics with(out) indexing. In *Proceedings of Logic and Engineering of Natural Language Semantics 14 (LENLS 14)*, Paper 24, pp. 1–12. University of Tsukuba: JSAI-isAI 2017.

P. Dekker. 2012. *Dynamic Semantics*, vol. 91 of *Studies in Linguistics and Philosophy*. Dordrecht: Springer Verlag.

NINJAL. 2016. NINJAL Parsed Corpus of Modern Japanese. (Version 1.0). National Institute for Japanese Language and Linguistics. (http://npcmj.ninjal.ac.jp/interfaces/ accessed 2018/02/28).

J. Ruppenhofer, M. Ellsworth, M.R.L. Petruck, C.R. Johnson, C.F. Baker, and J. Scheffczyk. 2016. FrameNet II: Extended Theory and Practice. Tech. rep., Berkeley.

G.R. Sampson. 1995. *English for the Computer: The SUSANNE Corpus and Analytic Scheme*. Oxford: Clarendon Press (Oxford University Press).

B. Santorini. 2010. Annotation manual for the Penn Historical Corpora and the PCEEC (Release 2). Department of Computer and Information Science, University of Pennsylvania, Philadelphia. (http://www.ling.upenn.edu/histcorpora/annotation).

A. Tarski and R.L. Vaught. 1956. Arithmetical extensions of relational systems. *Compositio Mathematica* 13:81–102.