

DynamicPower at SemEval-2016 Task 8: Processing syntactic parse trees with a Dynamic Semantics core

Alastair Butler

National Institute for Japanese Language and Linguistics

ajb129@hotmail.com

Abstract

This is a system description paper for a submission to Task 8 of SemEval-2016: Meaning Representation Parsing. No use was made of the training data provided by the task. Instead existing components were combined to form a pipeline able to take raw sentences as input and output meaning representations. Components are a part-of-speech tagger and parser trained on the Penn Parsed Corpus of Modern British English to produce syntactic parse trees, a semantic role labeller and a named entity recogniser to supplement obtained parse trees with word sense, functional and named entity information, followed by an adapted Tarskian satisfaction relation for a Dynamic Semantics that is used to transform a syntactic parse into a predicate logic based meaning representation, followed by conversion to penman/AMR notation required for the task appraisal.

1 Introduction

This is a system description paper for a submission to Task 8 of SemEval-2016: Meaning Representation Parsing. Syntactic structures are first obtained by parsing raw language input, from which meaning representations are derived by printing off information accumulated with an adapted Tarskian satisfaction relation for a Dynamic Semantics (Dekker, 2012). This is akin to compositional approaches of formal semantics that view the task of reaching a semantic value as being rooted in first obtaining a syntactic parse. Key advantages are modularity and domain independence.

This paper is structured as follows. Section 2 sketches the method used to obtain a syntactic parse. Section 3 covers reaching a semantic representation. Section 4 outlines conversion to penman/AMR notation. Section 5 reports experiment results. Section 6 is a conclusion. An appendix details how to run the available implementation.

2 Obtaining a syntactic parse

The approach first needs a way to obtain syntactic parse trees. Major components used were the Stanford Log-linear Part-Of-Speech Tagger¹ (Toutanova et al., 2003) and the Berkeley Parser² (Petrov and Klein, 2003), both trained on data from the years of 1840–1908 of the Penn Parsed Corpus of Modern British English³ (Kroch et al., 2010). The particular setup followed suggestions of pre-processing and post-processing made by Kulick et al. (2014) and used tools, notably `create_stripped`, from the system of Fang, Butler and Yoshimoto (2014).⁴

Dating from over one hundred years ago, training data for the syntactic parser was not chosen for suitability to a potential task domain, but instead for the practical benefit that parse results would conform to the scheme proposed in the *Annotation manual for the Penn Historical Corpora and the Parsed Corpus of Early English Correspondence (PCEEC)* (Santorini, 2010). This scheme is exceptionally consistent, especially with regards to facilitating identi-

¹<http://nlp.stanford.edu/software/tagger.html>

²<https://github.com/slavpetrov/berkeleyparser>

³<http://www.ling.upenn.edu/hist-corpora/PPCMBE-RELEASE-1/index.html>

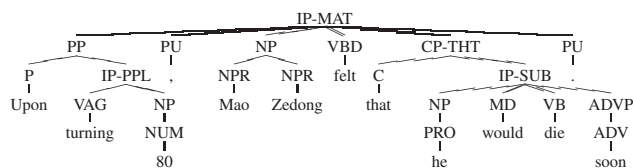
⁴<http://www.compling.jp/haruniwa/index.html>

fication of construction types (small clause, comparative, cleft, etc.) and for its handling of coordination, offering the least obstacle for a robust conversion to the structures fed to the semantic component (seen in the next section).

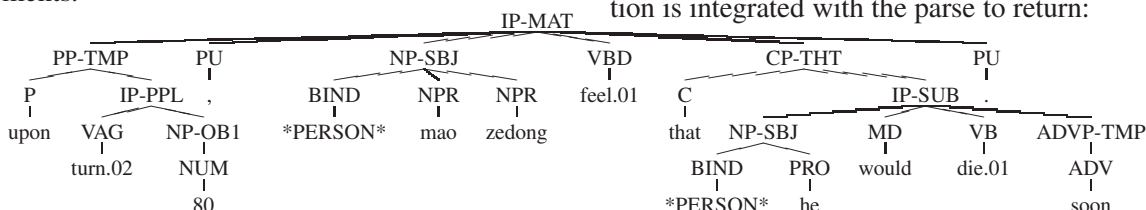
As an example, consider:

- (1) Upon turning 80, Mao Zedong felt that he would die soon.

A parse in tree form returned looks like:



All words are part-of-speech labelled. Clause structure is generally flat with multiply branching nodes and no specified VP level, except with VP coordination. Phrasal nodes (NP, PP, ADVP, etc.) immediately dominate the phrase head (N, P, ADV, etc.) which has as sisters both modifiers and complements.



Despite such flat clause and phrase structure, extended phrase labels marking function enable distinguishing the presence of modifiers (participial clauses (IP-PPL), adverbial clauses (CP-ADV), relative clauses (CP-REL), etc.) from complements (infinitive complements (IP-INF), that-complements (CP-THT), embedded questions (CP-QUE), etc.).

To arrive at a more complete parse for the task, word sense and functional information was obtained with *mateplus*⁵ (Roth and Woodsend, 2014), which is an extended version of the *mate-tools* semantic role labeller (Björkelund et al., 2009). In addition, named entity information was gathered with the Stanford Named Entity Recognizer⁶ (Finkel et al., 2005) using the MUC model that labels e.g., PERSON, ORGANIZATION, and LOCATION. Furthermore, pronouns, e.g., *he*, *she*, *they*, are default

marked PERSON. From (1) as input, the combination of these tools collects the following information:

upon	TMP	—	O
turn	PMOD	turn.02	O
80	OBJ	—	O
,	P	—	O
mao	NAME	—	PERSON
zedong	SBJ	—	PERSON
felt	ROOT	feel.01	O
that	OBJ	—	O
he	SBJ	—	PERSON
would	SUB	—	O
die	VC	die.01	O
soon	TMP	—	O
.	P	—	O

The first column gives word lemmas, the second contains functional information to identify syntactic subjects (SBJ) and objects (OBJ) as well as adjunct roles such as LOC, MSR and TMP, the third column provides word sense information related to PropBank (Bonial et al., 2010) semantic frames ('turn.02', 'feel.02', 'die.01'), and the fourth column provides entity information. This column information is integrated with the parse to return:

The tree now includes extended phrase labels marking function (e.g., NP-SBJ=subject, NP-OB1=direct object, ADVP-TMP=temporal modifier). Terminal nodes are either word lemmas or, whenever available, PropBank word senses. Furthermore entity information is integrated with a BIND tag.

3 Obtaining semantic analysis

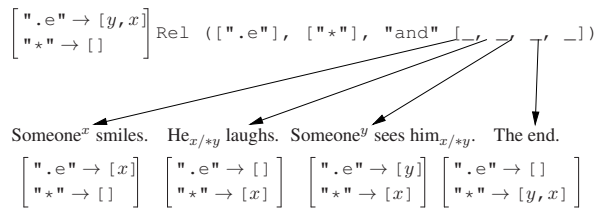
Having syntactic structures, the next step is to reach a level of semantic analysis. This is derived by printing off information accumulated with an adapted Tarskian satisfaction relation for a Dynamic Semantics (Dekker, 2012). Specifically, use is made of the Treebank Semantics implementation,⁷ with syntactic structures converted into expressions of a formal language (Scope Control Theory or SCT) with a number of primitive operations (notably, among

⁵<https://github.com/microth/mateplus>

⁶<http://nlp.stanford.edu/software/CRF-NER.html>

⁷<http://www.compling.jp/ts>

others: `Namely` to make available fresh bindings, `T` to create bound arguments, `At` to allocate semantic roles, `Close` to bring about quantificational closures, `Rel` to establish predicate or connective relations, and `If` to conditionalise how calculation of a semantic value proceeds based on assignment state). The full list of operations and details are given in Butler (2015). Operations access or possibly alter a sequence based information state (Vermeulen, 2000) that retains binding information by assigning (possibly empty) sequences of values to binding names. This can be demonstrated with `Rel` creating an "and" relation with four arguments, each of which are processed against a different assignment state determined by instances of `Namely` embedded in occurrences of *Someone*:



Pronouns are able to link to "*" bindings, which are accessible bindings that have reached the discourse context because of prior indefinites, while indefinites take bindings from ".e", which is a source for fresh bindings. This approach gives a handle on discourse, and more generally governs the interaction of quantification to capture the empirical results of accessibility from Discourse Representation Theory (Kamp and Reyle, 1993), as well as intra-sentential binding conditions (Butler, 2010).

Regarding the running example, the tree for (1) is converted to the following:

```
val ex1 =
  ( fn fh =>
    ( fn lc =>
      ( npr "PERSON" "mao_zedong" "ARG0"
        ( ( someFact fh ( gensym "TIME" ) ".e" "FACT"
          ( control lc
            ( ( fn lc =>
              ( some lc fh ( gensym "ENTITY" ) ".e"
                ( someClassic lc fh ( gensym "ATTRIB" ) ".e" nil
                  ( adj lc "80" ) "CARDINAL"
                    ( nn lc "" ) ) "ARG1"
                      ( verb lc ( gensym "EVENT" ) ".event" ["ARG1" "turn.02"] )
                        [ "CARDINAL", "ARG1", "ARG0", "h" ] ) ) "TMP"
                          ( past ".event"
                            ( embVerb lc ( gensym "EVENT" ) ".event" ["TMP", "ARG0"] "feel.01"
                              ( THAT lc
                                ( ( fn lc =>
                                  ( pro ["*"] fh ["PERSON"] ( gensym "PERSON" ) ".e" "he" "ARG0"
                                    ( md fh "would" free
                                      ( advp lc fh ".event"
                                        ( adv lc "TMP_soon"
                                          ( verb lc ( gensym "EVENT" ) ".event" ["ARG0"] "die.01" )
                                            [ "FACT", "ARG0", "ARG1", "h" ] ) ) ) )
                                                [ "TMP", "ARG0", "ARG1", "h" ] )
                                                  [ ".e", ".e", ".event" ]
```

Such an expression is built exploiting the input syntactic structure by locating any complement for the phrase head to scope over, and adding modifiers with scope over the head. During construction information about binding names is gathered and integrated at clause levels with `fn fh =>` and `fn lc =>` acting as lambda abstractions.

With conversion to `ex1`, part of speech tags transform to operations (`some` (indefinite), `nn` (noun predicate), `verb` (predicate with an event argument), `pro` (pronoun), `gensym` (trigger to create a fresh binding), `free` (ensures no quantificational closure), etc.). Also, constructions can bring about the inclusion of operations (e.g., `someFact` with the participial clause, and `THAT` with the that-complement). Conversion also adds (i) information about local binding names (e.g., "ARG0" (logical subject role) and "ARG1" (logical object role)) and (ii) information about sources for fresh bindings from quantificational closures ("`@e`", "`.e`" and "`.event`"). Once built `ex1` reduces to primitives of the SCT language, the start of which is as follows:

```
Sct.Close ("E",["@e", ".e", ".event"],
  Sct.Clean (0, ["ARG0"], "*"),
  Sct.Namely (Lang.C ("mao_zedong", "PERSON"),"@e",
    Sct.Lam "@e", "ARG0",
    Sct.Clean (0, ["TMP"], "*"),
    Sct.QuantThrow (Lang.X (1, "TIME"),".e",
      Sct.Lam ".e", "TMP",
      Sct.Rel (["@e", ".e", ".event"], ["*", "*", "*"], "", [
        Sct.Throw ".e",
        Sct.Rel ([], [], "FACT", [Sct.At (Sct.T ("TMP", 0), "FACT"),
          Sct.At (
            Sct.Lam "TMP", "",
            Sct.Clean (0, ["ARG1", "LGS", "ARG2"], "ARG0",
              Sct.Clean (0, ["TMP", "ARG1", "h"], "*"),
              Sct.If (fn,
                Sct.Clean (0, ["ARG1", "LGS", "ARG2"], "ARG0",
                  Sct.Clean (0, ["TMP", "ARG1", "h"], "*", ...
```

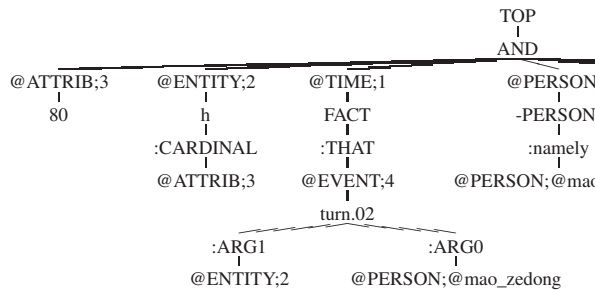
Such an expression is given to the adapted Tarskian satisfaction procedure which, instead of returning a semantic value (e.g., `true` or `false` with respect to a model), is used to produce a meaning representation by printing accumulated information, thus:

```
∃ PERSON[6] ATTRIB[3] TIME[1] EVENT[4] EVENT[7]
  EVENT[5] ENTITY[2] (
    80 (ATTRIB[3]) ∧ is_CARDINAL(ENTITY[2], ATTRIB[3]) ∧
    is_FACT_THAT(TIME[1],
      turn.02(EVENT[4], PERSON[mao_zedong], ENTITY[2])) ∧
    PERSON[6] = he{PERSON[mao_zedong]} ∧
    TMP_soon(EVENT[7]) ∧ past(EVENT[5]) ∧
    feel.01(EVENT[5], PERSON[mao_zedong],
      MD_would(die.01(EVENT[7], PERSON[6])) ∧
    is_contained_in(TMP(EVENT[5]), TIME[1]))
```

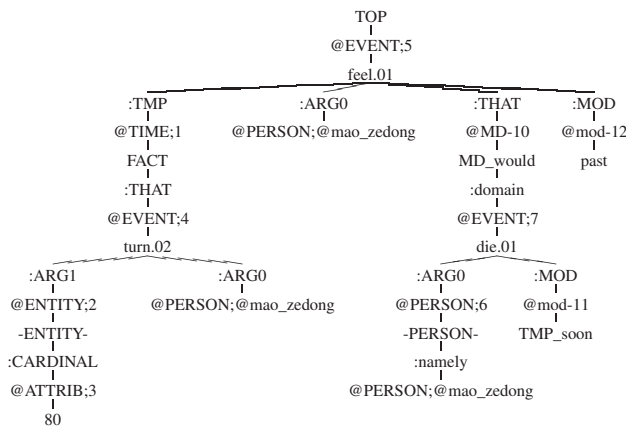
This gives a Davidsonian representation (Davidson, 1967) in which verbs are encoded with minimally an event argument. All bindings are existentially quantified over at the highest level, which is a convenience for reaching penman/AMR notation and not an approach limitation. Also note the pronoun is resolved to the only accessible PERSON antecedent.

4 Conversion to penman/AMR notation

Conversion to penman/AMR notation (Matthiessen and Bateman, 1991; Banarescu et al., 2015) involves transforming obtained semantic structures into trees with explicit argument role information. An argument of each predicate (e.g., ‘@EVENT’ if present, or the sole argument of a one-place predicate) is made the parent of the predicate. Also, binding is made implicit with the removal of quantification levels. Thus, the running example becomes:

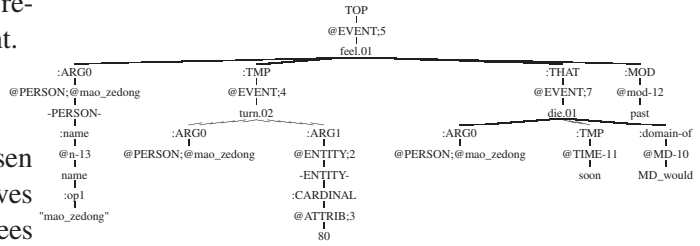


Content is further re-packaged: a daughter D of an AND level is moved inside a sister S when the argument name at the root of D is contained as an argument within S. Movement is to only one location.



In addition, flatter structures are arrived at by excising redundant linking information, e.g., @TIME; 1 FACT :THAT and folding tree material around inverse roles (signalled by ending the role name with

‘-oF’). The latter is seen with the modal (*would*), but also serves to compact long distance dependencies that arise with relative clauses, comparatives, clefts, etc. There is also expansion of name information and some reordering of role placement.



The final step involves pretty printing the assembled tree into the penman/AMR format, as well as the removal of tense information and remapping role names, e.g., :THAT changes to :ARG1, :TMP to :time, :ON to :prep-on, :CARDINAL to :quant, :ATTRIBUTE to :mod, and :POS to :poss.

```
( EVENT-5 / feel-02
  :ARG0 ( PERSON-mao_zedong / PERSON
    :name ( n-13 / name :op1 "mao" :op2 "zedong" ) )
  :ARG1 ( EVENT-7 / die-01
    :ARG0 PERSON-mao_zedong
    :time ( TIME-11 / soon ) )
  :time ( EVENT-4 / turn-02
    :ARG0 PERSON-mao_zedong
    :ARG1 ( ENTITY-2 / ENTITY
      :quant ( ATTRIB-3 / 80 ) ) ) )
```

For comparison, the gold analysis from the task training data for the running example is as follows:

```
( f / feel-02
  :ARG0 ( p / person
    :wiki "Mao_Zedong"
    :name ( n / name :op1 "Mao" :op2 "Zedong" ) )
  :ARG1 ( d / die-01
    :ARG1 p
    :time ( s / soon ) )
  :time ( t / turn-02
    :ARG1 p
    :ARG2 ( t2 / temporal-quantity
      :quant 80
      :unit ( y / year ) ) ) )
```

Table 1: Parsing results on the evaluation data of SemEval-2016 Task 8 and test data of LDC2015E86

	Precision	Recall	F-score
Task 8 eval	-	-	0.47
Test	0.37	0.39	0.38

It can be seen that gaps remain. In particular, there is no wikification to ground terms, and there is a lack of named entity information. Establishment of arguments as well as multiple roles for the same entity is largely successful, although there are mismatches with `die-01` and `turn-02`, reflecting that the semantic analysis stage lacks information for how PropBank roles for arguments should be allocated.

5 Experiments

The method of this paper is evaluated on the shared task evaluation data, which includes 1053 sentences. The smatch score on the evaluation data is 0.47. Table 1 also reports smatch score on the LDC2015E86 dataset, which includes 1371 test sentences. The scores are calculated with Smatch v2.0.2 (Cai and Knight, 2013), which evaluates the precision, recall, and *F1* of the concepts and relations all together. The score for Task 8 is higher than the performance on the LDC2015E86 test data. Reasons for the difference include parser performance being better on the evaluate data, and there being fewer non-compositional aspects of representation in the evaluate data.

6 Conclusion

To sum up, this paper has described a modularised approach for building meaning representations, with a key role for an adapted Tarskian satisfaction relation for a Dynamic Semantics as the method to integrate and connect information sourced from a syntactic parser, semantic role labeller and named entity recogniser. Task performance was limited by not using the training data provided by the task, in particular: lacking information to allocate PropBank roles, neglecting wikification, and missing entity information to replicate non-compositional aspects of the Abstract Meaning Representation (AMR) specification (Banarescu et al., 2015). Nevertheless,

this contribution indicates that AMRs are not far removed from what a compositional semantics can achieve, which is of interest for connecting to results from the formal semantics literature, such as gaining a treatment for quantification, as well as for relating to “Sembank” resources built with Discourse Representation Theory/Dynamic Semantics, such as the Groningen Meaning Bank⁸ (Basile et al., 2012) and Treebank Semantics Corpus⁹ (Butler and Yoshimoto, 2012).

Appendix: Implementation¹⁰

Assuming `text` is some original (multi-)sentence segmented data, `text.psd` contains the output from a parser trained on the PPCMBE, `text.mate` is output from `mateplus`, and `text.ner` is output from the Stanford Named Entity Recognizer, the following pipeline creates fully parsed data as described in section 2.

```
cat text.psd | add_functional text.mate |
add_sense text.mate | add_tsv_ner text.ner
| add_lemma text.mate | tree_select_word_2
> fullparse.psd
```

A script to recover the potentially multiple sentence segmentation of `text` obscured by parsing is achieved thus:

```
cat fullparse.psd | parse_discourse_split
text > segment.sh
```

The following pipeline achieves the semantic analysis of section 3, as well as the conversion to penman/AMR notation of section 4.

```
cat fullparse.psd | prepare_PPCMBE |
segment.sh | parse_normalize -propbank
-free -bind | see_sct -free -reset |
run_sct -penman | penman_like_amr |
pretty_penman
```

Acknowledgements

This paper has benefitted from the comments of three anonymous reviewers, discussions with Pascual Martínez-Gómez, Masaaki Nagata, and Kei Yoshimoto, and most particularly advice from the SemEval Task 8 organiser Jonathan May, all of

⁸<http://gmb.let.rug.nl>

⁹<https://github.com/ajb129/tscorpus>

¹⁰All pipeline programs are from: <https://github.com/ajb129>

which is very gratefully acknowledged. This research is supported by the Japan Society for the Promotion of Science (JSPS), Research Project Number: 15K02469.

References

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2015. Abstract Meaning Representation (AMR) 1.2.2 Specification. <https://github.com/amrisi/amr-guidelines/blob/master/amr.md>.
- V. Basile, J. Bos, K. Evang, and N.J. Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the 8th Int. Conf. on Language Resources and Evaluation*. Istanbul, Turkey.
- Claire Bonial, Olga Babko-Malaya, Jinho D. Choi, Jena Hwang, and Martha Palmer. 2010. *PropBank Annotation Guidelines*. Center for Computational Language and Education Research, Institute of Cognitive Science, University of Colorado at Boulder, 3rd edn.
- Anders Björkelund, Love Hafdel, and Pierre Nugues. 2014. Multilingual semantic role labeling. In *Proceedings of The Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 43–48. Boulder, Colorado.
- Alastair Butler. 2010. *The Semantics of Grammatical Dependencies*, vol. 23 of *Current Research in the Semantics/Pragmatics Interface*. Bingley: Emerald.
- Alastair Butler. 2015. *Linguistic Expressions and Semantic Processing: A Practical Approach*. Heidelberg: Springer-Verlag.
- Alastair Butler and Kei Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology - LiLT* 7(1):1–22.
- Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proc. of the ACL 2013*.
- Donald Davidson. 1967. The logical form of action sentences. In N. Rescher, ed., *The Logic of Decision and Action*. Pittsburgh: University of Pittsburgh Press. Reprinted in: D. Davidson, 1980. *Essays on Actions and Events*. Clarendon Press, Oxford, pages 105–122.
- Paul Dekker. 2012. *Dynamic Semantics*, vol. 91 of *Studies in Linguistics and Philosophy*. Dordrecht: Springer Verlag.
- Tsaiwei Fang, Alastair Butler, and Kei Yoshimoto. 2014. Parsing Japanese with a PCFG treebank grammar. In *Proceedings of the Twentieth Annual Meeting of the Association of Natural Language Processing*, pages 432–435. Sapporo, Japan.
- Jenny Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370.
- Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.
- Anthony Kroch, Beatrice Santorini, and Ariel Diertani. 2010. *The Penn-Helsinki Parsed Corpus of Modern British English (PPCMBE)*. Department of Linguistics, University of Pennsylvania. CD-ROM, second edition, (<http://www.ling.upenn.edu/hist-corpora>).
- Seth Kulick, Anthony Kroch, and Beatrice Santorini. 2014. The Penn Parsed Corpus of Modern British English: First parsing results and analysis. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 662–667. Baltimore, Maryland, USA: Association for Computational Linguistics.
- Christian Matthiessen and John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter Publishers.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411.
- Michael Roth and Kristian Woodsend. 2014. Composition of word representations improves semantic role labelling. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 407–413. Doha, Qatar.
- Beatrice Santorini. 2010. Annotation manual for the Penn Historical Corpora and the PCEEC (Release 2). Tech. rep., Department of Computer and Information Science, University of Pennsylvania, Philadelphia. (<http://www.ling.upenn.edu/histcorporata/annotation>).
- Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259.
- C. F. M. Vermeulen. 2000. Variables as stacks: A case study in dynamic model theory. *Journal of Logic, Language and Information* 9:143–167.