

Joint Apposition Extraction with Syntactic and Semantic Constraints

Will Radford and James R. Curran
a-lab, School of Information Technologies
University of Sydney
NSW, 2006, Australia
{wradford, james}@it.usyd.edu.au

Abstract

Appositions are adjacent NPs used to add information to a discourse. We propose systems exploiting syntactic and semantic constraints to extract appositions from OntoNotes. Our joint log-linear model outperforms the state-of-the-art Favre and Hakkani-Tür (2009) model by $\sim 10\%$ on Broadcast News, and achieves 54.3% F-score on multiple genres.

1 Introduction

Appositions are typically adjacent coreferent noun phrases (NP) that often add information about named entities (NEs). The apposition in Figure 1 consists of three comma-separated NPs – the first NP (HEAD) names an entity and the others (ATTRS) supply age and profession attributes. Attributes can be difficult to identify despite characteristic punctuation cues, as punctuation plays many roles and attributes may have rich substructure.

While linguists have studied apposition in detail (Quirk et al., 1985; Meyer, 1992), most apposition extraction has been within other tasks, such as coreference resolution (Luo and Zitouni, 2005; Culotta et al., 2007) and textual entailment (Roth and Sammons, 2007). Extraction has rarely been intrinsically evaluated, with Favre and Hakkani-Tür’s work a notable exception.

We analyze apposition distribution in OntoNotes 4 (Pradhan et al., 2007) and compare rule-based, classification and parsing extraction systems. Our best system uses a joint model to classify pairs of NPs with features that faithfully encode syntactic and semantic restrictions on appositions, using parse trees and WordNet synsets.

$\{John\ Ake\}_h$, $\{48\}_a$, $\{a\ \text{former}\ \text{vice-president}\ \text{in}\ \text{charge}\ \text{of}\ \text{legal}\ \text{compliance}\ \text{at}\ \text{American}\ \text{Capital}\ \text{Management}\ \&\ \text{Research}\ \text{Inc.},\ \text{in}\ \text{Houston},\}_a, \dots$

Figure 1: Example apposition from OntoNotes 4

Our approach substantially outperforms Favre and Hakkani-Tür on Broadcast News (BN) at 54.9% F-score and has state-of-the-art performance 54.3% F-score across multiple genres. Our results will immediately help the many systems that already use apposition extraction components, such as coreference resolution and IE.

2 Background

Apposition is widely studied, but “grammarians vary in the freedom with which they apply the term ‘apposition’” (Quirk et al., 1985). They are usually composed of two or more adjacent NPs, hierarchically structured, so one is the *head* NP (HEAD) and the rest *attributes* (ATTRS). They are often flagged using punctuation in text and pauses in speech. Pragmatically, they allow an author to introduce new information and build a shared context (Meyer, 1992).

Quirk et al. propose three tests for apposition: i) each phrase can be omitted without affecting sentence acceptability, ii) each fulfils the same syntactic function in the resultant sentences, iii) extralinguistic reference is unchanged. Strict interpretations may exclude other information-bearing cases like *pseudo-titles* (e.g. $\{\text{President}\}_a \{\text{Bush}\}_h$), but include some adverbial phrases (e.g. $\{\text{John}\ \text{Smith}\}_h$, $\{\text{formerly}\ \text{(the}\ \text{president})_{AP}\}_a$). We adopt the OntoNotes guidelines’ relatively strict interpretation: “a noun phrase that modifies an immediately-adjacent noun phrase (these may be separated by only a comma, colon, or parenthesis).” (BBN, 2004–2007).

Unit	TRAIN _F	DEV _F	TEST _F	TRAIN	DEV	TEST
Sents.	9,595	976	1,098	48,762	6,894	6,896
Appos.	590	64	68	3,877	502	490

Table 1: Sentence and apposition distribution

Apposition extraction is a common component in many NLP tasks: coreference resolution (Luo and Zitouni, 2005; Culotta et al., 2007; Bengtson and Roth, 2008; Poon and Domingos, 2008), textual entailment (Roth and Sammons, 2007; Cabrio and Magnini, 2010), sentence simplification (Miwa et al., 2010; Candido et al., 2009; Siddharthan, 2002) and summarization (Nenkova et al., 2005). Comma ambiguity has been studied in the RTE (Srikumar et al., 2008) and generation domains (White and Rajkumar, 2008).

Despite this, few papers to our knowledge explicitly evaluate apposition extraction. Moreover, apposition extraction is rarely the main research goal and descriptions of the methods used are often accordingly terse or do not match our guidelines. Lee et al. (2011) use rules to extract appositions for coreference resolution, selecting only those that are explicitly flagged using commas or parentheses. They do not separately mark HEAD and ATTR and permit relative clauses as an ATTR. While such differences capture useful information for coreference resolution, these methods would be unfairly disadvantaged in a direct evaluation.

Favre and Hakkani-Tür (2009, FHT) directly evaluate three extraction systems on OntoNotes 2.9 news broadcasts. The first retrains the Berkeley parser (Petrov and Klein, 2007) on trees labelled with appositions by appending the HEAD and ATTR suffix to NPs – we refer to this as a Labelled Berkeley Parser (LBP). The second is a CRF labelling words using an IOB apposition scheme. Token, POS, NE and BP-label features are used, as are presence of speech pauses. The final system classifies parse tree phrases using an Adaboost classifier (Schapire and Singer, 2000) with similar features.

The LBP, IOB and phrase systems score 41.38%, 32.76% and 40.41%, while their best uses LBP tree labels as IOB features, scoring 42.31%. Their focus on BN automated speech recognition (ASR) output, which precludes punctuation cues, does not indicate how well the methods perform on textual genres. Moreover all systems use parsers or parse-label features and do not completely evaluate non-parser methods for extraction despite including baselines.

Form	#	%	Reverse form	#	%	Σ%
H t A	2109	55.9	A t H	724	19.2	75.1
A H	482	12.8	H A	205	5.4	93.3
H , A	1843	48.9	A , H	532	14.1	63.0
A H	482	12.9	H A	205	5.4	81.3
H (A	146	3.9	A (H	16	0.4	85.6
A : H	94	2.5	H : A	23	0.6	88.7
H -- A	66	1.8	A -- H	35	0.9	91.4
A - H	31	0.8	H - A	21	0.6	92.8

Table 2: Apposition forms in TRAIN with abstract (top) and actual (bottom) tokens, e.g., H t A indicates an HEAD, one token then an ATTR.

3 Data

We use apposition-annotated documents from the English section of OntoNotes 4 (Weischedel et al., 2011). We manually adjust appositions that do not have exactly one HEAD and one or more ATTR¹. Some appositions are nested, and we keep only “leaf” appositions, removing the higher-level appositions.

We follow the CoNLL-2011 scheme to select TRAIN, DEV and TEST datasets (Pradhan et al., 2011). OntoNotes 4 is made up of a wide variety of sources: broadcast conversation and news, magazine, newswire and web text. Appositions are most frequent in newswire (one per 192 words) and least common in broadcast conversation (one per 645 words) with the others in between (around one per 315 words).

We also replicate the OntoNotes 2.9 BN data used by FHT, selecting the same sentences from OntoNotes 4 (TRAIN_F/DEV_F/TEST_F). We do not “speechify” our data and take a different approach to nested apposition. Table 1 shows the distribution of sentences and appositions (HEAD-ATTR pairs).

3.1 Analysis

Most appositions in TRAIN have one ATTR (97.4%) with few having two (2.5%) or three (0.1%). HEADS are typically shorter (median 5 tokens, 95% < 7) than ATTRS (median 7 tokens, 95% < 15). Table 2 shows frequent apposition forms. Comma-separated apposition is the most common (63%) and 93% are separated by zero or one token. HEADS are often composed of NES: 52% PER and 13% ORG, indicating an entity about which the ATTR adds information.

¹Available at <http://schwa.org/resources>

Pattern and Example	P	R	F
$\{\text{ne:PER}\}_h \# \{\text{pos:NP} (\text{pos:IN ne:LOC ORG GPE})?\}_a \#$ “{Jian Zhang} _h , {the head of Chinese delegation} _a ,”	73.1	21.9	33.7
$\{\text{pos:DT gaz:role relation}\}_a \#? \{\text{ne:PER}\}_h$ “{his new wife} _a {Camilla} _h ”	45.9	9.5	15.8
$\{\text{ne:ORG GPE}\}_h \# \{\text{pos:DT pos:NP}\}_a \#$ “{Capetronic Inc.} _h , {a Taiwan electronics maker} _a ,”	60.4	6.0	10.9
$\{\text{pos:NP}\}_a \# \{\text{ne:PER}\}_h \#$ “{The vicar} _a , {W.D. Jones} _h ,”	33.7	4.5	7.9
$\{\text{ne:PER}\}_h \# \{\text{pos:NP pos:POS pos:NP}\}_a \#$ “{Laurence Tribe} _h , {Gore ’s attorney} _a ,”	82.0	4.0	7.7

Table 3: The top-five patterns by recall in the TRAIN dataset. ‘#’ is a pause (e.g., punctuation), ‘|’ a disjunction and ‘?’ an optional part. Patterns are used to combine tokens into NPs for pos:NP.

4 Extracting Appositions

We investigate different extraction systems using a range of syntactic information. Our systems that use syntactic parses generate candidates (pairs of NPs: p_1 and p_2) that are then classified as apposition or not.

This paper contributes three complementary techniques for more faithfully modelling apposition. Any adjacent NPs, disregarding intervening punctuation, could be considered candidates, however stronger syntactic constraints that only allow sibling NP children provide higher precision candidate sets. Semantic compatibility features encoding that an ATTR provides consistent information for its HEAD. A joint classifier models the complete apposition rather than combining separate phrase-wise decisions. Taggers and parsers are trained on TRAIN and evaluated on DEV or TEST. We use the C&C tools (Curran and Clark, 2003) for POS and NE tagging and the and the Berkeley Parser (Petrov and Klein, 2007), trained with default parameters.

Pattern POS, NE and lexical patterns are used to extract appositions avoiding parsing’s computational overhead. Rules are applied independently to tokenized and tagged sentences, yielding HEAD-ATTR tuples that are later deduplicated. The rules were manually derived from TRAIN² and Table 3 shows the top five of sixteen rules by recall over TRAIN. The “role” gazetteer is the transitive closure of hyponyms of the WordNet (Miller, 1995) synset `person.n.01` and “relation” manually constructed (e.g., “father”, “colleague”). Tuples are post-processed to remove spurious appo-

²There is some overlap between TRAIN and DEV_F/TEST_F with appositions from the latter used in rule generation.

sitions such as comma-separated NE lists³.

Adjacent NPs This low precision, high recall baseline assumes all candidates, depending on generation strategy, are appositions.

Rule We only consider HEADS whose syntactic head is a PER, ORG, LOC or GPE NE. We formalise *semantic compatibility* by requiring the ATTR head to match a gazetteer dependent on the HEAD’s NE type. To create PER, ORG and LOC gazetteers, we identified common ATTR heads in TRAIN and looked for matching WordNet synsets, selecting the most general hypernym that was still semantically compatible with the HEAD’s NE type.

Gazetteer words are pluralized using `pattern.en` (De Smedt and Daelemans, 2012) and normalised. We use partitive and NML-aware rules (Collins, 1999; Vadas and Curran, 2007) to extract syntactic heads from ATTRs. These must match the type-appropriate gazetteer, with ORG and LOC/GPE falling back to PER (e.g., “the champion, Apple”).

Extracted tuples are post-processed as for Pattern and reranked by the OntoNotes specificity scale (i.e., NNP > PRO > Def. NP > Indef. NP > NP), and the more specific unit is assigned HEAD. Possible ATTRs further to the left or right are checked, allowing for cases such as Figure 1.

Labelled Berkeley Parser We train a LBP on TRAIN and recover appositions from parsed sentences. Without syntactic constraints this is equivalent to FHT’s LBP system (LBP_F) and indicated by † in Tables.

Phrase Each NP is independently classified as HEAD, ATTR or None. We use a log-linear model with a SGD optimizer from scikit-learn (Pedregosa

³Full description: <http://schwa.org/resources>

Model	Full system			-syn			-sem			-both			+gold		
Pattern	44.8	34.9	39.2	-	-	-	-	-	-	-	-	52.2	39.6	45.1	
Adj NPs	11.6	58.0	19.3	3.6	65.1	6.8	-	-	-	-	-	16.0	85.3	27.0	
Rule	65.3	46.8	54.5	43.7	50.0	46.7	-	-	-	-	-	79.1	62.0	69.5	
LBP	66.3	52.2	58.4	47.8	53.0	†50.3	-	-	-	-	-	-	-	-	
Phrase	73.2	45.6	56.2	77.7	41.0	53.7	73.2	44.6	55.4	77.7	40.8	‡53.5	89.0	58.2	70.4
Joint	66.3	49.0	56.4	68.5	48.6	56.9	70.4	47.0	56.4	68.9	48.0	56.6	87.9	69.5	77.6
Joint LBP	69.6	51.0	58.9	69.6	49.6	57.9	71.5	49.0	58.2	68.3	48.6	56.8	-	-	-

Table 4: Results over DEV: each column shows precision, recall and F-score. -syn/-sem/-both show the impact of removing constraints/features, +gold shows the impact of parse and tagging errors.

et al., 2011). The binary features are calculated from a generated candidate phrase (p) and are the same as FHT’s phrase system (Phrase_F), denoted ‡ in Tables. In addition, we propose the features below and to decode classifications, adjacent apposition-classified NPs are re-ordered by specificity.

- p precedes/follows punctuation/interjection
- p starts with a DT or PRP\$ (e.g., “{the director}_a” or “{her husband}_a”)
- p ’s syntactic head matches a NE-specific *semantic* gazetteer (e.g., “{the famous actor}_a” \rightarrow PER, “{investment bank}_a” \rightarrow ORG)
- p ’s syntactic head has the POS CD (e.g., “{John Smith}_h, {34}_a, ...”)
- p ’s NE type (e.g., “{John Smith}_h” \rightarrow PER)
- Specificity rank

Joint The final system classifies *pairs* of phrases (p_1, p_2) as: HEAD-ATTR, ATTR-HEAD or None. The system uses the phrase model features as above as well as pairwise features:

- the cross-product of selected features for p_1 and p_2 : gazetteer matches, NE type, specificity rank. This models the compatibility between p_1 and p_2 . For example, if the HEAD has the NE type PER and the ATTR has the syntactic head in the PER gazetteer, for example “{Tom Cruise}_h, {famous actor}_a,” \rightarrow (p_1 : PER, p_2 : PER-gaz)
- If semantic features are found in p_1 **and** p_2
- p_1/p_2 specificity (e.g., equal, $p_1 > p_2$)
- whether p_1 is an acronym of p_2 or vice-versa

5 Results

We evaluate by comparing the extracted HEAD-ATTR pairs against the gold-standard. Correct pairs match gold-standard bounds and label. We report precision (P), recall (R) and F_1 -score (F).

Table 4 shows our systems’ performance on the multi-genre DEV dataset, the impact of removing syntactic constraints, semantic features and

parse/tag error. Pattern performance is reasonable at 39.2% F-score given its lack of full syntactic information. All other results use parses and, although it has a low F-score, the Adjacent NPs’ 65.1% recall, without syntactic constraints, is the upper bound for the parse-based systems. Statistical models improve performance, with the joint models better than the higher-precision phrase model as the latter must make two independently correct classification decisions. Our best system has an F-score of 58.9% using a joint model over the de-labelled trees produced by the LBP. This indicates that although our model does not use the apposition labels from the tree, the tree is a more suitable structure for extraction. This system substantially improves on our implementation of FHT’s LBP_F (†) and Phrase_F (‡) systems by 8.6% and 5.4%⁴.

Removing syntactic constraints mostly reduces performance in parse-based systems as the system must consider lower-quality candidates. The F-score increase is driven by higher precision at minimal cost to recall. Removing semantic features has less impact and removing both is most detrimental to performance. These features have less impact on joint models; indeed, joint performance using BP trees increases without the features, perhaps as joint models already model the syntactic context.

We evaluate the impact of parser and tagger error by using gold-standard resources. Gold-standard tags and trees improve recall in all cases leading to F-score improvements (+gold). The pattern system is reasonably robust to automatic tagging errors, but parse-based models suffer considerably from automatic parses. To compare the impact of tagging and parsing error, we configure the joint system to use gold parses and automatic NE tags and vice versa. Using automatic tags does not greatly impact performance (-1.3%), whereas

⁴We do not implement the IOB or use LBP features for TRAIN as these would require n-fold parser training.

Model	P	R	F
LBP _F †	53.1	46.9	49.8
Phrase _F ‡	71.5	30.2	42.5
Pattern	44.8	34.3	38.8
LBP	63.9	45.1	52.9
Joint LBP	66.9	45.7	54.3

Table 5: Results over TEST: FHT’s (top) and our (bottom) systems.

Error	BP	LBP	δ
PP Attachment	5,585	5,396	-189
NP Internal Structure	1,483	1,338	-145
Other	3,164	3,064	-100
Clause Attachment	3,960	3,867	-93
Modifier Attachment	1,523	1,700	177
Co-ordination	3,095	3,245	150
NP Attachment	2,615	2,680	65
Total	30,189	29,859	-330

Table 6: Selected BP/LBP parse error distribution.

using automatic parses causes a drop of around 20% to 57.7%, demonstrating that syntactic information is crucial for apposition extraction.

We compare our work with Favre and Hakkani-Tür (2009), training with TRAIN_F and evaluating over TEST_F—exclusively BN data. Our implementations of their systems, Phrase_F and LBP_F, perform at 43.6% and 44.1%. Our joint LBP system is substantially better, scoring 54.9%.

Table 5 shows the performance of our best systems on the TEST dataset and these follow the same trend as DEV. Joint LBP performs the best at 54.3%, 4.5% above LBP_F.

Finally, we test whether labelling appositions can help parsing. We parse DEV trees with LBP and BP, remove apposition labels and analyse the impact of labelling using the Berkeley Parser Analyser (Kummerfeld et al., 2012). Table 6 shows the LBP makes fewer errors, particularly NP internal structuring, PP and clause attachment classes at the cost of modifier attachment and co-ordination errors. Rather than increasing parsing difficulty, apposition labels seem complementary, improving performance.

6 Conclusion

We present three apposition extraction techniques. Linguistic tests for apposition motivate strict syntactic constraints on candidates and semantic features encode the addition of compatible informa-

tion. Joint models more faithfully capture apposition structure and our best system achieves state-of-the-art performance of 54.3%. Our results will immediately benefit the large number of systems with apposition extraction components for coreference resolution and IE.

Acknowledgements

The authors would like to thank the anonymous reviewers for their suggestions. Thanks must also go to Benoit Favre for his clear writing and help answering our questions as we replicated his dataset and system. This work has been supported by ARC Discovery grant DP1097291 and the Capital Markets CRC Computable News project.

References

- BBN. 2004–2007. Co-reference guidelines for english ontonotes. Technical Report v6.0, BBN Technologies.
- Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics, Honolulu, Hawaii.
- Elena Cabrio and Bernardo Magnini. 2010. Toward qualitative evaluation of textual entailment systems. In *Coling 2010: Posters*, pages 99–107. Coling 2010 Organizing Committee, Beijing, China.
- Arnaldo Candido, Erick Maziero, Lucia Specia, Caroline Gasperin, Thiago Pardo, and Sandra Aluisio. 2009. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42. Association for Computational Linguistics, Boulder, Colorado.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 81–88. Association

- for Computational Linguistics, Rochester, New York.
- James Curran and Stephen Clark. 2003. Language independent ner using a maximum entropy tagger. In Walter Daelemans and Miles Osborne, editors, *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 164–167.
- Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *Journal of Machine Learning Research*, 13:2013–2035.
- Benoit Favre and Dilek Hakkani-Tür. 2009. Phrase and word level strategies for detecting appositions in speech. In *Proceedings of Interspeech 2009*, pages 2711–2714. Brighton, UK.
- Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the wall street corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059. Jeju Island, South Korea.
- Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford’s multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the CoNLL-2011 Shared Task*. URL pubs/conll1st2011-coref.pdf.
- Xiaoqiang Luo and Imed Zitouni. 2005. Multilingual coreference resolution with syntactic features. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 660–667. Association for Computational Linguistics, Vancouver, British Columbia, Canada.
- Charles F. Meyer. 1992. *Apposition in Contemporary English*. Cambridge University Press, Cambridge, UK.
- George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.
- Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun’ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796. Coling 2010 Organizing Committee, Beijing, China.
- Ani Nenkova, Advaith Siddharthan, and Kathleen McKeown. 2005. Automatically learning cognitive status for multi-document summarization of newswire. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 241–248. Association for Computational Linguistics, Vancouver, British Columbia, Canada.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Slav Petrov and Dan Klein. 2007. Learning and inference for hierarchically split PCFGs. In *Proceedings of the 22nd AAAI Conference of Artificial Intelligence*, pages 1642–1645. Vancouver, Canada.
- Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659. Association for Computational Linguistics, Honolulu, Hawaii.
- Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27. Portland, OR USA.
- Sameer S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. OntoNotes: A unified relational semantic representation. In *Proceedings of the International Conference on Semantic Computing*, pages 517–526. Washington, DC USA.
- Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. General Grammar Series. Longman, London, UK.

- Dan Roth and Mark Sammons. 2007. Semantic and logical inference model for textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 107–112. Association for Computational Linguistics, Prague.
- Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168.
- Advaith Siddharthan. 2002. Resolving attachment and clause boundary ambiguities for simplifying relative clause constructs. In *Proceedings of the ACL Student Research Workshop (ACLSRW 2002)*, pages 60–65. Association for Computational Linguistics, Philadelphia.
- Vivek Srikumar, Roi Reichart, Mark Sammons, Ari Rappoport, and Dan Roth. 2008. Extraction of entailed semantic relations through syntax-based comma resolution. In *Proceedings of ACL-08: HLT*, pages 1030–1038. Columbus, OH USA.
- David Vadas and James R. Curran. 2007. Parsing internal noun phrase structure with collins’ models. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 109–116. Melbourne, Australia.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2011. OntoNotes Release 4.0. Technical report, Linguistic Data Consortium, Philadelphia, PA USA.
- Michael White and Rajakrishnan Rajkumar. 2008. A more precise analysis of punctuation for broad-coverage surface realization with CCG. In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 17–24. Coling 2008 Organizing Committee, Manchester, England.