Revisiting Selectional Preferences for Coreference Resolution

Benjamin Heinzerling[*]
AIPHES
Heidelberg Institute for
Theoretical Studies
benjamin.heinzerling@h-its.org

Nafise Sadat Moosavi[*]
Heidelberg Institute for
Theoretical Studies
nafise.moosavi@h-its.org

Michael Strube
Heidelberg Institute for
Theoretical Studies
michael.strube@h-its.org

Abstract

Selectional preferences have long been claimed to be essential for coreference resolution. However, they are mainly modeled only implicitly by current coreference resolvers. We propose a dependency-based embedding model of selectional preferences which allows fine-grained compatibility judgments with high coverage. We show that the incorporation of our model improves coreference resolution performance on the CoNLL dataset, matching the state-of-the-art results of a more complex system. However, it comes with a cost that makes it debatable how worthwhile such improvements are.

1 Introduction

Selectional preferences have long been claimed to be useful for coreference resolution. In his seminal work on "Resolving Pronominal References" Hobbs (1978) proposed a semantic approach that requires reasoning about the "demands the predicate makes on its arguments." For example, selectional preferences allow resolving the pronoun *it* in the text *"The Titanic hit an iceberg. It sank quickly."* Here, the predicate *sink* 'prefers' certain subject arguments over others: It is plausible that a ship sinks, but implausible that an iceberg does.

Work on the automatic acquisition of selectional preferences has shown considerable progress (Dagan and Itai, 1990; Resnik, 1993; Agirre and Martinez, 2001; Pantel et al., 2007; Erk, 2007; Ritter et al., 2010; Van de Cruys, 2014). However, today's coreference resolvers (Martschat and Strube, 2015; Wiseman et al., 2016; Clark and Manning, 2016a, i.a.) capture selectional preferences only implicitly at best, e.g., via a given mention's dependency governor and other contextual features.

Since negative results do not often get reported, there is no clear evidence in the literature regarding the non-utility of particular knowledge sources. Consequently, an absence of the explicit modeling of selectional preferences in the recent literature is an indicator that incorporating this knowledge source has not been very successful for coreference resolution.

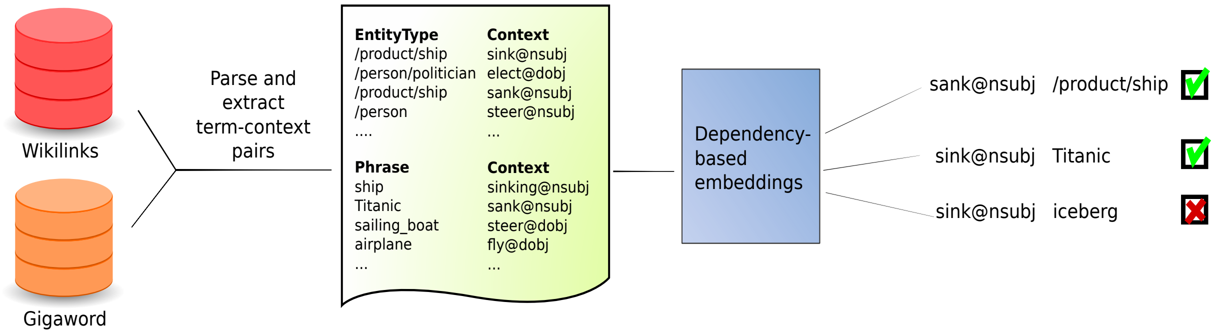
More than ten years ago, Kehler et al. (2004) declared the "non-utility of predicate-argument structures for pronoun resolution" and observed that minor improvements on a small dataset were due to fortuity rather than selectional preferences having captured meaningful world knowledge relations.

The claim by Kehler et al. (2004) is based on selectional preferences extracted from a, by current standards, small number of 2.8m predicate-argument pairs. Furthermore, they employ a simple (linear) maximum entropy classifier, which requires manual definition of feature combinations and is unlikely to fully capture the complex interaction between selectional preferences and other coreference features. Therefore, it is worth revisiting how a better selectional preference model affects the performance of a more complex coreference resolver.

In this work, we propose a fine-grained, high-coverage model of selectional preferences and study its impact on a state-of-the-art, non-linear coreference resolver. We show that the incorporation of our selectional preference model improves the performance. However, it is debatable whether such small improvements, that cost notable extra time or resources, are advantageous.

[*] These authors contributed equally to this work.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1332–1339
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics


Wikilinks
Gigaword
Parse and extract term-context pairs
Dependency-based embeddings
sank@nsubj /product/ship ✓
sink@nsubj Titanic ✓
sink@nsubj iceberg ☒

EntityType	Context
/product/ship	sink@nsubj
/person/politician	elect@dobj
/product/ship	sank@nsubj
/person	steer@nsubj
....	...

Phrase	Context
ship	sinking@nsubj
Titanic	sank@nsubj
sailing_boat	steer@dobj
airplane	fly@dobj
...	...

Figure 1: Dependency-based embedding model of selectional preferences.

2 Modeling Selectional Preferences

The main design choice when modeling selectional preferences is the selection of a relation inventory, i.e. the concepts and entities that can be relation arguments, and the semantic relationships that hold between them.

Prior work has studied many relation inventories. Predicate-argument statistics for word-word pairs (*eat, food*)[1] are easy to obtain but do not generalize to unseen pairs (Dagan and Itai, 1990). Class-based approaches generalize via word-class pairs (*eat, /nutrient/food*) (Resnik, 1993) or class-class pairs (*/ingest, /nutrient/food*) (Agirre and Martinez, 2001), but require disambiguation of words to classes and are limited by the coverage of the lexical resource providing such classes (e.g. WordNet).

Other possible relation inventories include semantic representations such as FrameNet frames and roles, event types and arguments, or abstract meaning representations. While these semantic representations are arguably well-suited to model meaningful world knowledge relationships, automatic annotation is limited in speed and accuracy, making it difficult to obtain a large number of such "more semantic" predicate-argument pairs. In comparison, syntactic parsing is both fast and accurate, making it trivial to obtain a large number of accurate, albeit "less semantic" predicate-argument pairs. The drawback of a syntactic model of selectional preferences is susceptibility to lexical and syntactic variation. For example, *The Titanic sank* and *The ship went under* differ lexically and syntactically, but would have the same or a very similar representation in a semantic framework such as FrameNet.

Our model of selectional preferences (Figure 1) overcomes this drawback via distributed representation of predicate-argument pairs, using (syntactic) dependencies that were specifically designed for semantic downstream tasks, and by resolving named entities to their fine-grained entity types.

Distributed representation. Inspired by Structured Vector Space (Erk and Padó, 2008), we embed predicates and arguments into a low-dimensional space in which (representations of) predicate slots are close to (representations of) their plausible arguments, as should be arguments that tend to fill the same slots of similar predicates, and predicate slots that have similar arguments. For example, *captain* should be close to *pilot*, *ship* to *airplane*, the subject of *steer* close to both *captain* and *pilot*, and also to, e.g., the subject of *drive*. Such a space allows judging the plausibility of unseen predicate-argument pairs.[2]

We construct this space via dependency-based word embeddings (Levy and Goldberg, 2014). To see why this choice is better-suited for modeling selectional preferences than alternatives such as word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014), consider the following example:

$$\begin{array}{ccccc} \text{captain} & \xleftarrow{\text{nsubj}} & \text{steers} & \xrightarrow{\text{dobj}} & \text{ship} \\ :: & & & & :: \\ \text{pilot} & \xleftarrow{\text{nsubj}} & \text{steers} & \xrightarrow{\text{dobj}} & \text{airplane} \end{array}$$

Here, *captain* and *ship*, have high syntagmatic similarity, i.e., these words are semantically related and tend to occur close to each other. This also holds for *pilot* and *airplane*. In contrast, *captain* and *pilot*, as well as *ship* and *airplane* have high paradigmatic similarity, i.e., they are seman-

[1] Examples due to Agirre and Martinez (2001).

[2] Prior work generalizes to unseen predicate-argument pairs via WordNet synsets (Resnik, 1993), a generalization corpus (Erk, 2007), or tensor factorization (Van de Cruys, 2010). Closest to our approach is neural model by Van de Cruys (2014), which, however, has much lower coverage since it is limited to 7k verbs and 30k arguments.

tically similar and occur in similar contexts. A model of selectional preferences requires paradigmatic similarity: The representations of *captain* and *pilot* in such a model should be similar, since they both can plausibly fill the subject slot of the predicate *steer*. Due to their use of linear context windows, word2vec and GloVe capture syntagmatic similarity, while dependency-based embeddings capture paradigmatic similarity (cf. Levy and Goldberg, 2014).

Enhanced++ dependencies. Due to distributed representation, our model generalizes over syntactic variation such as active/passive alternations: For example, *steer@dobj*[3] is highly similar to *steer@nsubjpass* (see Appendix for more examples). To further mitigate the effect of employing syntax as a proxy for semantics, we use Enhanced++ dependencies (Schuster and Manning, 2016). Enhanced++ dependencies aim to support semantic applications by modifying syntactic parse trees to better reflect relations between content words. For example, the plain syntactic parse of the sentence *Both of the girls laughed* identifies *Both* as subject of *laughed*. The Enhanced++ representation introduces a subject relation between *girls* and *laughed*, which allows learning more meaningful selectional preferences: Our model should learn that girls (and other humans) laugh, while learning that an unspecified *both* laughs is not helpful.

Fine-grained entity types. A good model of selectional preferences needs to generalize over named entities. For example, having encountered sentences like *The Titanic sank*, our model should be able to judge the plausibility of an unseen sentence like *The RMS Lusitania sank*. For popular named entities, we can expect the learned representations of *Titanic* and *RMS Lusitania* to be similar, allowing our model to generalize, i.e., it can judge the plausibility of *The RMS Lusitania sank* by virtue of the similarity between *Titanic* and *RMS Lusitania*. However, this will not work for rare or emerging named entities, for which no, or only low-quality, distributed representations have been learned. To address this issue, we incorporate fine-grained entity typing (Ling and Weld). For each named entity encountered during training, we generate an additional training instance by replacing the named entity with its entity type, e.g. *(Titanic, sank@nsubj)* yields *(/product/ship, sank@nsubj)*.

3 Implementation

We train our model by combining term-context pairs from two sources. Noun phrases and their dependency context are extracted from GigaWord (Parker et al., 2011) and entity types in context from Wikilinks (Singh et al., 2012). Term-context pairs are obtained by parsing each corpus with the Stanford CoreNLP dependency parser (Manning et al., 2014). After filtering, this yields ca. 1.4 billion phrase-context pairs such as *(Titanic, sank@nsubj)* from GigaWord and ca. 12.9 million entity type-context pairs such as *(/product/ship, sank@nsubj)* from Wikilinks. Finally, we train dependency-based embeddings using the generalized word2vec version by Levy and Goldberg (2014), obtaining distributed representations of selectional preferences. To identify fine-grained types of named entities at test time, we first perform entity linking using the system by Heinzerling et al. (2016), then query Freebase (Bollacker et al., 2008) for entity types and apply the mapping to fine-grained types by Ling and Weld.

The plausibility of an argument filling a particular predicate slot can now be computed via the cosine similarity of their associated embeddings. For example, in our trained model, the similarity of *(Titanic, sank@nsubj)* is 0.11 while the similarity of *(iceberg, sank@nsubj)* is -0.005, indicating that an iceberg sinking is less plausible.

4 Do Selectional Preferences Benefit Coreference Resolution?

We now investigate the effect of incorporating selectional preferences, implicitly and explicitly, in coreference resolution.

Figure 2 shows the selectional preference similarity of 10.000 coreferent and 10.000 non-coreferent mention pairs sampled randomly from the CoNLL 2012 training set. As we can see, while coreferent mention pairs are more similar than non-coreferent mention pairs according to the selectional preference similarity, there is not a direct relation between the similarity values and the coreferent relation. This indicates that coreference does not have a linear relation to the selectional preference similarities. However, it is worth investigating how these similarity values affect the overall performance when they are combined with

[3] In this work, a predicate's argument slots are denoted *predicate@slot*.

	MUC			B^3			CEAF$_e$			CoNLL	LEA		
	R	P	F$_1$	R	P	F$_1$	R	P	F$_1$	Avg. F$_1$	R	P	F$_1$
baseline	70.09	80.01	74.72	57.64	70.09	63.26	54.47	63.92	58.82	65.60	54.02	66.45	59.59
−gov	70.10	79.96	74.71	57.51	70.31	63.27	54.41	64.08	58.85	65.61	53.93	66.76	59.66
+SP	70.85	79.31	74.85	58.93	69.16	63.64	55.25	63.78	59.21	65.90	55.29	65.53	59.98
Reinforce	70.98	78.81	74.69	58.97	69.05	63.61	55.66	63.28	59.23	65.84	55.31	65.32	59.90

Table 1: Results on the CoNLL 2012 test set.

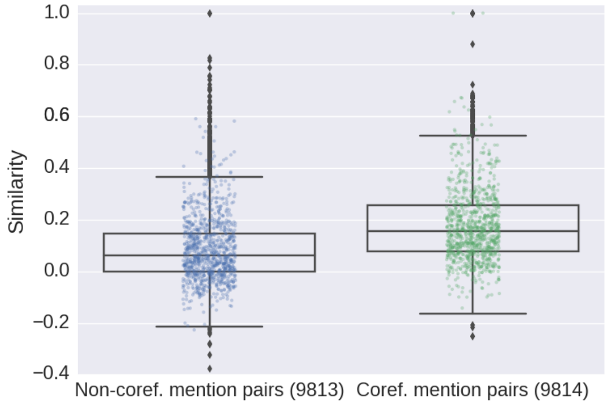


Figure 2: Selectional preference similarities of 10k coreferent and 10k non-coreferent mention pairs. Lines and boxes represent quartiles, diamonds outliers, points subsamples with jitter. Coreferent mention pairs are more similar than non-coreferent mention pairs with a Matthews correlation coefficient of 0.30, indicating weak to moderate correlation.

	MUC	B^3	CEAF$_e$	CoNLL	LEA
			development		
baseline	74.10	63.95	59.73	65.93	60.16
+embedding	74.38	64.42	60.45	66.42	60.65
+binned sim.	74.36	64.54	60.21	66.37	60.77
			test		
baseline	74.72	63.26	58.82	65.60	59.59
+embedding	74.53	63.41	59.03	65.66	59.69
+binned sim.	74.85	63.64	59.21	65.90	59.98

Table 2: Incorporating the selectional preference model as new embeddings (+embedding) vs. as new pairwise features (+binned sim.).

other knowledge sources in a non-linear way.

We select the ranking model of deep-coref (Clark and Manning, 2016b) as our baseline. deep-coref is a neural model that combines the input features through several hidden layers. *Baseline* in Table 1 reports our baseline results on the CoNLL 2012 test set. The results are reported using *MUC* (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), *CEAF$_e$* (Luo, 2005), the average F_1 score of these three metrics, i.e. CoNLL score, and *LEA* (Moosavi and Strube, 2016b). deep-coref includes the embeddings of the dependency governor of mentions. Combined with the relative position of a mention to its governor, deep-coref may be able to implicitly capture selectional preferences to some extent. *−gov* in Table 1 represents deep-coref performance when governors are not incorporated. As we can see, the exclusion of the governor information does not affect the performance. This result shows that the implicit modeling of selectional preferences does not provide any additional information to the coreference resolver.

For each mention, we consider (1) the whole mention string, (2) the whole mention string without articles, (3) mention head, (4) context representation, i.e. governor@dependency-relation, and (5) entity types if the mention is a named entity. We obtain an embedding for each of the above properties if they exist in the selectional preference model, otherwise we set them to unknown.

For each (antecedent, anaphor) pair, we consider all the acquired embeddings of anaphor and antecedent. We try two different ways of incorporating this knowledge into deep-coref including: (1) incorporating the computed embeddings directly as a new set of inputs, i.e. *+embedding* in Table 2. We add a new hidden layer on top of the new embeddings and combine its output with outputs of the hidden layers associated with other sets of inputs; and (2) computing a similarity value between all possible combinations of the antecedent-anaphor acquired embeddings and then binarizing all similarity values, i.e. *+binned sim.* in Table 2.

Providing selectional preference embeddings directly to deep-coref adds more complexity to the baseline coreference resolver. Yet, it performs on-par with *+binned sim.* on the development set and generalizes worse on the test set. *+SP* in Table 1 is the performance of *+binned sim.* on the test set. As we can see from the results, adding selectional

does [**that**]$_{ante}$ really impact the case ... [it]$_{ana}$ just shows	(impact@nsubj,shows@nsubj)
[it]$_{ante}$ will ask a U.S. bankruptcy court to allow [it]$_{ana}$	(ask@nsubj,allow@dobj)
[a **strain** that has n't even presented [itself]$_{ana}$]$_{ante}$	(presented@nsubj,presented@dobj)

Table 3: Examples of +SP correct links on the development set that do not exist in the baseline output.

Error type	Mention type		
	Proper	Common	Pronoun
Recall	-28	-29	-53
Precision	+18	+74	+61

Table 4: Differences in the number of recall and precision errors on the CoNLL'12 test set in comparison to the baseline.

preferences as binary features improves over the baseline.

Reinforce in Table 1 presents the results of the reward-rescaling model of Clark and Manning (2016a) that are so far the highest reported results on the official test set. The reward rescaling model of Clark and Manning (2016a) casts the ranking model of Clark and Manning (2016b) in the reinforcement learning framework which considerably increases the training time, from two days to six days in our experiments.

We analyze how our selectional preference model affects the resolution of various types of mentions. We use Martschat and Strube (2014)'s toolkit [4] to perform recall and error analyses. The differences in the number of recall and precision errors in +*SP* compared to *baseline* on the test set are reported in Table 4.

By using our selectional preference features, the number of recall errors decreases for all types of mentions. The recall error reduction is more prominent for pronouns. On the other hand, the number of precision errors increases for all types of mentions. The increase in the precision error is the highest for common nouns. Overall, +*SP* creates about 260 more links than *baseline*.

Table 3 lists a few examples from the development set in which +*SP* creates a link that *baseline* does not. It also includes the similarity that has a high value for the linked mentions and probably is the reason for creating the link. For instance, in the first example, based on our model, similarity(*impact@nsubj,shows@nsubj*) is known and it is also higher than similarity(*impact@dobj,shows@nsubj*).

In order to estimate a higher bound on the expected performance boost, we run the *baseline* and +*SP* models only on anaphoric mentions. By using anaphoric mentions, the performance improves by one percent, based on both the CoNLL score and *LEA*. This result indicates that the incorporation of selectional preferences creates many links for non-anaphoric mentions, which in turn decreases precision. Therefore, the overall performance does not improve substantially when system mentions are used. deep-coref incorporates anaphoricity scores at resolution time. One possible way to further improve the results of +*SP* is to incorporate anaphoricity scores at the input level. In this way, the coreference resolver could learn to use selectional preferences mainly for mentions that are more likely to be anaphoric. However, given that the F_1 score of current anaphoricity determiners or singleton detectors is only around 85 percent (Moosavi and Strube, 2016a, 2017), the effect of using system anaphoricity scores might be small.

5 Conclusions

We introduce a new model of selectional preferences, which combines dependency-based word embeddings and fine-grained entity types. In order to be effective, a selectional preference model should (1) have a high coverage so it can be used for large datasets like CoNLL, and (2) be combined with other knowledge sources in a nonlinear way. Our selectional preference model slightly improves coreference resolution performance, but considering the extra resources that are required to train the model, it is debatable whether such small improvements are advantageous for solving coreference.

Acknowledgments

This work has been supported by the German Research Foundation (DFG) as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1 and the Klaus Tschira Foundation, Heidelberg, Germany.

[4] https://github.com/smartschat/cort

References

Eneko Agirre and David Martinez. 2001. Learning class-to-class selectional preferences. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*, pages 15–22.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 28–30 May 1998, pages 563–566.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data,* Vancouver, B.C., Canada, 10–12 June 2008, pages 1247–1250.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* Austin, Tex., 1–5 November 2016, pages 2256–2262.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Berlin, Germany, 7–12 August 2016.

Tim Van de Cruys. 2010. A non-negative tensor factorization model for selectional preference induction. *Natural Language Engineering*, 16(4):417–437.

Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26–35, Doha, Qatar.

Ido Dagan and Alon Itai. 1990. Automatic processing of large corpora for the resolution of anaphora references. In *Proceedings of the 13th International Conference on Computational Linguistics,* Helsinki, Finland, 20–25 August 1990, volume 3, pages 330–332.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* Prague, Czech Republic, 23–30 June 2007, pages 216–223.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 897–906.

Benjamin Heinzerling, Alex Judea, and Michael Strube. 2016. HITS at TAC KBP 2015: Entity discovery and linking, and event nugget detection. In *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 16–17 November 2015.

Jerry R. Hobbs. 1978. Resolving pronominal references. *Lingua*, 44:311–338.

Andrew Kehler, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Boston, Mass., 2–7 May 2004, pages 289–296.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

Xiao Ling and Daniel S. Weld. Fine-grained entity recognition. In *Proceedings of the 26th Conference on the Advancement of Artificial Intelligence,* Toronto, Ontario, Canada, 22–26 July 2012, pages 94–100.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pages 25–32.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Sebastian Martschat and Michael Strube. 2014. Recall error analysis for coreference resolution. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pages 2070–2081.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:405–418.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the ICLR 2013 Workshop Track.*

Nafise Sadat Moosavi and Michael Strube. 2016a. Search space pruning: A simple solution for better coreference resolvers. In *Proceedings of the*

2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, Cal., 12–17 June 2016, pages 1005–1011.

Nafise Sadat Moosavi and Michael Strube. 2016b. Which coreference evaluation metric do you trust? A proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Berlin, Germany, 7–12 August 2016, pages 632–642.

Nafise Sadat Moosavi and Michael Strube. 2017. Use generalized representations, but do not forget surface features. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 1–7, Valencia, Spain.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning inferential selectional preferences. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 564–571.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition. LDC2011T07.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pages 1532–1543.

Philip Resnik. 1993. *Selection and Information: A Class-based Approach to Lexical Relationships*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Penn.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, University of Massachusetts, Amherst.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 45–52, San Mateo, Cal. Morgan Kaufmann.

Sam Wiseman, Alexander M. Rush, and Stuart Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* San Diego, Cal., 12–17 June 2016. To appear.

Appendix

Query	Most sim. predicate slots	Most sim. entity types	Most sim. phrases
sink@nsubj	sink@nsubj:xsubj	/product/ship	Sea_Diamond
	sink@nsubjpass	/event/natural_disaster	Prestige_oil_tanker
	sinking@nmod:of	/finance/stock_exchange	Samina
	slide@nsubj	/astral_body	Estonia_ferry
	capsizing@nmod:of	/person/religious_leader	k-159
	plunge@nsubj	/finance/currency	Navy_gunboat
	sink@nmod:along_with	/military	Dona_Paz
	sinking@nsubj	/geography/glacier	ferry_Estonia
	tumble@nsubj	/product/airplane	add-fisk-independent-nytsf
	slip@nsubj	/transit	Al-Salam_Boccaccio
ship	capsize@nmod:of	/product/ship	vessel
	some@nmod:aboard	/train	cargo_ship
	experience@nmod:aboard	/product/airplane	cruise_ship
	afternoon@nmod:aboard	/transit	boat
	pier@nmod:for	/product/spacecraft	freighter
	escort@nmod:including	/location/bridge	container_ship
	lift-off@nmod:of	/broadcast/tv_channel	cargo_vessel
	disassemble@nsubjpass:xsubj	/location	Navy_ship
	near-collision@nmod:with	/living_thing	warship
	Conger@compound	/chemistry	tanker
steer@dobj	guide@dobj	/broadcast/tv_channel	business_way
	steer@nsubjpass	/product/car	newr_nbkg_nwer_ndjn
	shepherd@dobj	/organization/sports_team	BahrainDinar
	steering@nmod:of	/product/ship	reynard-honda
	nudge@dobj	/product/spacecraft	zigzag_course
	pilot@dobj	/event/election	team_home
	propel@dobj	/medicine/medical_treatment	U.S._energy_policy
	maneuver@dobj	/building/theater	williams-bmw
	divert@dobj	/education/department	interest-rate_policy
	lurch@nsubj	/product/airplane	trimaran
/product/ship	Repulse@conj:and	/product/airplane	battleship_Bismarck
	destroyer@amod	/train	pt_boat
	capsize@nmod:of	/product/car	battleship
	experience@nmod:aboard	/park	USS_Nashville
	near-collision@nmod:with	/military	USS_Indianapolis
	line@cc	/event/natural_disaster	k-159
	brig@conj:and	/award	frigate
	-lrb-@nmod:on	/geography/island	warship
	Umberto@conj:and	/person/soldier	Oriskany
	rumour@xcomp	/location/body_of_water	sister_ship

Figure 3: Most similar terms for the queries *sink@nsubj*, *ship*, *steer*, and */product/ship*.