UTDMet: Combining WordNet and Corpus Data for Argument Coercion Detection

Kirk Roberts and Sanda Harabagiu

Human Language Technology Research Institute University of Texas at Dallas Richardson, Texas, USA {kirk,sanda}@hlt.utdallas.edu

Abstract

This paper describes our system for the classification of argument coercion for SemEval-2010 Task 7. We present two approaches to classifying an argument's semantic class, which is then compared to the predicate's expected semantic class to detect coercions. The first approach is based on learning the members of an arbitrary semantic class using WordNet's hypernymy structure. The second approach leverages automatically extracted semantic parse information from a large corpus to identify similar arguments by the predicates that select them. We show the results these approaches obtain on the task as well as how they can improve a traditional feature-based approach.

1 Introduction

Argument coercion (a type of *metonymy*) occurs when the expected semantic class (relative to the a predicate) is substituted for an object of a different semantic class. Metonymy is a pervasive phenomenon in language and the interpretation of metonymic expressions can impact tasks from semantic parsing (Scheffczyk et al., 2006) to question answering (Harabagiu et al., 2005). A seminal example in metonymy from (Lakoff and Johnson, 1980) is:

(1) The ham sandwich is waiting for his check.

The ARG1 for the predicate *wait* is typically an animate, but the "*ham sandwich*" is clearly not an animate. Rather, the argument is coerced to fulfill the predicate's typing requirement. This coercion is allowed because an object that would normally fulfill the typing requirement (the customer) can be uniquely identified by an attribute (the ham sandwich he ordered).

SemEval-2010 Task 7 ("Argument Selection and Coercion") (Pustejovsky and Rumshisky, 2009) was designed to evaluate systems that detect such coercions and provide a "compositional history" of argument selection relative to the predicate. In order to accomplish this, an argument is annotated with both the semantic class to which it belongs (the "source" type) as well as the class expected by the predicate (the "target" type). However, in the data provided, the target type was unambiguous given the lemmatized predicate, so the remainder of this paper discusses source type classification. The detection of coercion is then simply performed by checking if the classified source type and target type are different.

In our system, we explore two approaches with separate underlying assumptions about how arbitrary semantic classes can be learned. In our first approach, we assume a semantic class can be defined a priori from a set of seed terms and that WordNet is capable of defining the membership of that semantic class. We apply the PageRank algorithm in order to weight WordNet synsets given a set of seed concepts. In our second approach, we assume that arguments in the same semantic class will be selected by similar verbs. We apply a statistical test to determine the most representative predicates for an argument. This approach benefits from a large corpus from which we automatically extracted 200 million predicate-argument pairs.

The remainder of this paper is organized as follows. Section 2 discusses our WordNet-based approach. Section 3 describes our corpus approach. Section 4 discusses our experiments and results. Section 5 provides a conclusion and direction for future work. Due to space limitations, previous work is discussed when relevant.

2 PageRanking WordNet Hypernyms

Our first approach assumes that semantic class members can be defined and acquired a priori.

Given a set of seed concepts, we mine WordNet for other concepts that may be in the same semantic class. Clearly, this approach has both practical limitations (WordNet does not contain every possible concept) and linguistic limitations (concepts may belong to different semantic classes based on their context). However, given the often vague nature of semantic classes (is a *building* an ARTI-FACT or a LOCATION?), access to a weighted list of semantic class members can prove useful for arguments not seen in the train set.

Using (Esuli and Sebastiani, 2007) as inspiration, we have implemented our own naive version of WordNet PageRank. They use sensedisambiguated glosses provided by eXtended WordNet (Harabagiu et al., 1999) to link synsets by starting with positive (or negative) sentiment concepts in order to find other concepts with positive (or negative) sentiment values. For our task, however, hypernymy relations are more appropriate for determining a given synset's membership in a semantic class. Hypernymy defines an IS-A relationship between the parent class (the *hypernym*) and one of its child classes (the hyponym). Furthermore, while PageRank assumes directed edges (e.g., hyperlinks in a web page), we use undirected edges. In this way, if HYPERNYMOF(A, B), then A's membership in a semantic class strengthens B's and vice versa.

Briefly, the formula for PageRank is:

$$\mathbf{a}^{(k)} = \alpha \mathbf{a}^{(k-1)} \mathbf{W} + (1-\alpha) \mathbf{e}$$
(1)

where $\mathbf{a}^{(k)}$ is the weight vector containing weights for every synset in WordNet at time k; $\mathbf{W}_{i,j}$ is the inverse of the total number of hypernyms and hyponyms for synset i if synset j is a hypernym or hyponym of synset i; \mathbf{e} is the initial score vector; and α is a tuning parameter. In our implementation, $\mathbf{a}^{(0)}$ is initialized to all zeros; α is fixed at 0.5; and $\mathbf{e}_i = 1$ if synset i is in the seed set S, and zero otherwise. The process is then run until convergence, defined by $|\mathbf{a}_i^{(k)} - \mathbf{a}_i^{(k-1)}| < 0.0001$ for all i.

The result of this PageRank is a weighted list containing every synset reachable by a hypernym/hyponym relation from a seed concept. We ran the PageRank algorithm six times, once for each semantic class, using the arguments in the train set as seeds. For arguments that are polysemous, we make a first WordNet sense assumption. Representative examples of the concepts generated from this approach are shown in Table 1.

ARTIFACT		DOCUMENT		
funny_wagon	.377	white_paper	.342	
liquor	.353	progress_report	.342	
iced_tea	.338	screenplay	.324	
tartan	.325	papyrus	.313	
alpaca	.325	pie_chart	.308	
Event		LOCATION		
rock_concert	.382	heliport	.381	
rodeo	.369	mukataa	.380	
radium_therapy	.357	subway_station	.342	
seminar	.347	dairy_farm	.326	
pub_crawl	.346	gateway	.320	
PROPOSITION		Sound		
dibs	.363	whoosh	.353	
white_paper	.322	squish	.353	
tall_tale	.319	yodel	.339	
commendation	.310	theme_song	.320	
field_theory	.309	oldie	.312	

Table 1: Some of the concepts (and scores) learned from applying PageRank to WordNet hypernyms.

3 Leveraging a Large Corpus of Semantic Parse Annotations

Our second approach assumes that semantic class members are arguments of similar predicates. As (Pustejovsky and Rumshisky, 2009) elaborate, predicates select an argument from a specific semantic class, therefore terms that belong in the same semantic class should be selected by similar predicates. However, this assumption is often violated: type coercion allows predicates to have arguments outside their intended semantic class. Our solution to this problem, partially inspired by (Lapata and Lascarides, 2003), is to collect statistics from an enormous amount of data in order to statistically filter out these coercions.

The English Gigaword Forth Edition corpus¹ contains over 8.5 million documents of newswire text collected over a 15 year period. We processed these documents with the SENNA² (Collobert and Weston, 2009) suite of natural language tools, which includes a part-of-speech tagger, phrase chunker, named entity recognizer, and PropBank semantic role labeler. We chose SENNA due to its speed, yet it still performs comparably with many state-of-the-art systems. Of the 8.5 million documents in English Gigaword, 8 million were successfully processed. For each predicate-argument pair in these documents, we gathered counts by argument type and argument head. The head was determined with simple heuristics from the chunk parse and parts-of-speech for each argument (arguments consisting of more than three phrase chunks were discarded). When available, named entity types (e.g., PERSON, ORGANIZATION, LO-

¹LDC2009T13

²http://ml.nec-labs.com/senna/

coffee	book	meeting	station	report	voice
drink	write	hold	own	release	hear
sip	read	attend	build	publish	raise
brew	publish	schedule	open	confirm	give
serve	title	chair	attack	issue	add
spill	sell	convene	close	comment	have
smell	buy	arrange	operate	submit	silence
sell	balance	call	fill	deny	sound
pour	illustrate	host	shut	file	lend
buy	research	plan	storm	prepare	crack
rise	review	make	set	voice	find

Table 2: Top ten predicates for the most common word in the train set for the six semantic classes.

CATION) were substituted for heads. This resulted in over 511 million predicate-argument pairs for argument types ARG0, ARG1, and ARG2. For this task, however, we chose only to use ARG1 arguments (direct objects), which resulted in 210 million pairs, 7.65 million of which were unique. The ARG1 argument was chosen because most of the arguments in the data are direct objects ³.

The "best" predicates for a given argument are defined by a ranking based on Fisher's exact test (Fisher, 1922):

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}$$
(2)

where *a* is the number of times the given argument was used with the given predicate, *b* is the number of times the argument was used with a different predicate, *c* is the number of times the predicate was used with a different argument, *d* is the number of times neither the given argument or predicate was used, and n = a+b+c+d. The top ranked (lowest *p*) predicates for the most common arguments in the training data are shown in Table 2.

4 Experiments

We have conducted several experiments to test the performance of the approaches outlined in Sections 2 and 3 along with additional features commonly found in information extraction literature. All experiments were conducted using the $SVM^{multiclass}$ support vector machine library⁴.

4.1 WordNet PageRank

We experimented with the output of our WordNet PageRank implementation along three separate dimensions: (1) which sense to use (since we did not incorporate a word sense disambiguation system), (2) whether to use the highest scoring semantic class or every class an argument belonged to, and (3) how to use the weight output by the algorithm. The results of these experiments yielded a single feature for each class that returns true if the argument is in that class, regardless of weight. This resulted in a micro-precision score of 75.6%.

4.2 Gigaword Predicates

We experimented with both (i) the number of predicates to use for an argument and (ii) the score threshold to use. Ultimately, the Fisher score did not prove nearly as useful as a classifier as it did as a ranker. Since the distribution of predicates for each argument varied significantly, choosing a high number of predicates would yield good results for some arguments but not others. However, because of size of the training data, we were able to choose the top 5 predicates for each argument as features and still achieve a reasonable microprecision score of 89.6%.

4.3 Other Features

Many other features common in information extraction are well-suited for this task. Given that SVMs can support millions of features, we chose to add many features simpler than those previously described in order to improve the final performance of the classifier. These include the lemma of the argument (both the last word's lemma and every word's lemma), the lemma of the predicate, the number of words in the argument, the casing of the argument, the part-of-speech of the argument's last word, the WordNet synset and all (recursive) hypernyms of the argument. Additionally, since the EVENT class is both the most common and the most often confused, we introduced two features based on annotated resources. The first feature indicates the most common part-of-speech for the un-lemmatized argument in the Treebank corpus. This helped classify examples such as thinking which was confused with a PROPOSITION for the predicate deny. Second, we introduced a feature that indicated if the un-lemmatized argument was considered an event in the TimeBank corpus (Pustejovsky et al., 2003) at least five times. This helped to distinguish events such as meeting, which was confused with a LOCATION for the predicate arrive.

4.4 Ablation Test

We conducted an ablation test using combinations of five feature sets: (1) our WordNet PageR-

³The notable exception to this, however, is *arrive*, where the data uses the destination argument. In the PropBank scheme (Palmer et al., 2005), this would correspond to the ARG4, which usually signifies an end state.

⁴http://svmlight.joachims.org/svm_multiclass.html



Table 3: Ablation test of feature sets showingmicro-precision scores.

		Precision	Recall
Selection vs. Coercion	Macro	95.4	95.7
Sciection vs. Coercion	Micro	95.4 96.3 96.5 96.1 100.0 100.0	96.3
Source Type	Macro	96.5	95.7
	Micro	96.1	96.1
Target Type	Macro	100.0	100.0
Target Type	Micro	96.5 96.1 100.0	100.0
Joint Type	Macro	85.5	95.2
	Micro	96.1	96.1

Table 4: Results for UTDMET on SemEval-2010 Task 7.

ank feature (WNPR), (2) our Gigaword Predicates feature (GWPA), (3) word, lemma, and part-ofspeech features (WORD), (4) WordNet synset and hypernym features (WNSH), and (5) Treebank and TimeBank features (EVNT). Of these $2^5 - 1 =$ 31 tests, 15 are shown in Table 3. The Gigaword Predicates (GWPA) was the best overall feature, but each feature set ended up helping the final score. WordNet PageRank (WNPR) even improved the score when combined WordNet hypernym features (WNSH) despite the fact that they are heavily related. Ultimately, WordNet PageRank had a greater precision, while the other Word-Net features had greater recall.

4.5 Task 7 Results

Table 4 shows the official results for UTDMET on the Task 7 data. The target type was unambiguous given the lemmatized predicate. For classifying selection vs. coercion, we simply checked to see if the classified source type was the same as the target type. If this was the case, we returned selection, otherwise a coercion existed.

5 Conclusion

We have presented two approaches for determining the semantic class of a predicate's argument. The two approaches capture different information and combine well to classify the "source" type in SemEval-2010 Task 7. We showed how this can be incorporated into a system to detect coercions as well as the argument's compositional history relative to its predicate. In future work we plan to extend this system to more complex tasks such as when the predicate may be polysemous or unseen predicates may be encountered.

Acknowledgments

The authors would like to thank Bryan Rink for several insights during the course of this work.

References

- Ronan Collobert and Jason Weston. 2009. Deep Learning in Natural Language Processing. Tutorial at NIPS.
- Andrea Esuli and Fabrizio Sebastiani. 2007. PageRanking WordNet Synsets: An Application to Opinion Mining. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, pages 424–431.
- Ronald A. Fisher. 1922. On the interpretation of χ^2 from contingency tables, and the calculation of p. 85(1):87–94.
- Sanda Harabagiu, George Miller, and Dan Moldovan. 1999. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In Proceedings of the SIGLEX Workshop on Standardizing Lexical Resources, pages 1–7.
- Sanda Harabagiu, Andrew Hickl, John Lehmann, and Dan Moldovan. 2005. Experiments with Interactive Question-Answering. In *Proceedings of the* 43rd Annual Meeting of the Association for Computational Linguistics, pages 205–214.
- George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press.
- Maria Lapata and Alex Lascarides. 2003. A Probabilistic Account of Logical Metonymy. *Computational Linguistics*, 21(2):261–315.
- Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.
- James Pustejovsky and Anna Rumshisky. 2009. SemEval-2010 Task 7: Argument Selection and Coercion. In Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, pages 88–93.
- James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003. The TIMEBANK Corpus. In *Proceedings of Corpus Linguistics*, pages 647–656.
- Jan Scheffczyk, Adam Pease, and Michael Ellsworth. 2006. Linking FrameNet to the Suggested Upper Merged Ontology. In *Proceedings of Formal Ontology in Information Systems*, pages 289–300.