Novel Semantic Features for Verb Sense Disambiguation

Dmitriy Dligach
The Center for Computational
Language and Education
Research
1777 Exposition Drive
Boulder, Colorado 80301
Dmitriy.Dligach
@colorado.edu

Martha Palmer
Department of Linguistics
University of Colorado
at Boulder
295 UCB
Boulder, Colorado 80309
Martha.Palmer
@colorado.edu

Abstract

We propose a novel method for extracting semantic information about a verb's arguments and apply it to Verb Sense Disambiguation (VSD). We contrast this method with two popular approaches to retrieving this information and show that it improves the performance of our VSD system and outperforms the other two approaches

1 Introduction

The task of Verb Sense Disambiguation (VSD) consists in automatically assigning a sense to a verb (target verb) given its context. In a supervised setting, a VSD system is usually trained on a set of pre-labeled examples; the goal of this system is to tag unseen examples with a sense from some sense inventory.

An automatic VSD system usually has at its disposal a diverse set of features among which the semantic features play an important role: verb sense distinctions often depend on the distinctions in the semantics of the target verb's arguments (Hanks, 1996). Therefore, some method of capturing the semantic knowledge about the verb's arguments is crucial to the success of a VSD system.

The approaches to obtaining this kind of knowledge can be based on extracting it from electronic dictionaries such as WordNet (Fellbaum, 1998), using Named Entity (NE) tags, or a combination of both (Chen, 2005). In this paper, we propose a novel method for obtaining semantic knowledge about words and show how it can be applied to VSD. We contrast this method with the other two approaches and compare their performances in a series of experiments.

2 Lexical and Syntactic Features

We view VSD as a supervised learning problem, solving which requires three groups of features: lexical, syntactic, and semantic. Lexical features include all open class words; we extract them from the target sentence and the two surrounding sentences. We also use as features two words on the right and on the left of the target verb as well as their POS tags. We extract syntactic features from constituency parses; they indicate whether the target verb has a subject/object and what their head words and POS tags are, whether the target verb is in a passive or active form, whether the target verb has a subordinate clause, and whether the target verb has a PP adjunct. Additionally, we implement several new syntactic features, which have not been used in VSD before: the path through the parse tree from the target verb to the verb's arguments and the subcategorization frame, as used in semantic role labeling.

3 Semantic Features

Consider the verb *prepare* for which our sense inventory defines two senses: (1) to put together, assemble (e.g. *He is going to prepare breakfast for the whole crowd*; *I haven't prepared my lecture*

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 29–32,
Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

yet); (2) to make ready (e.g. *She prepared the children for school every morning*). Knowing the semantic class of the objects *breakfast*, *lecture* and *children* is the decisive factor in distinguishing the two senses and facilitates better generalization from the training data. One way to obtain this knowledge is from WordNet (WN) or from the output of a NE-tagger. However, both approaches suffer from the same limitation: they collapse multiple semantic properties of nouns into a finite number of predefined static classes. E.g., the most immediate hypernym of *breakfast* in WN is *meal*, while the most immediate hypernym of *lecture* is *address*, which makes these two nouns unrelated. Yet, *breakfast* and *lecture* are both social events which share some semantic properties: they both can be *attended, hosted, delivered, given, held, organized* etc. To discover these class-like descriptions of nouns, one can observe which verbs take these nouns as objects. E.g. *breakfast* can serve as the object of *serve, host, attend, and cook* which are all indicative of *breakfast's* semantic properties.

Given a noun, we can dynamically retrieve other verbs that take that noun as an object from a dependency-parsed corpus; we call this kind of data **Dynamic Dependency Neighbors** (DDNs) because it is obtained dynamically and based on the dependency relations in the neighborhood of the noun of interest. The top 50[1] DDNs can be viewed as a reliable inventory of semantic properties of the noun. To collect this data, we utilized two resources: (1) MaltParser (Nivre, 2007) – a high-efficiency dependency parser; (2) English Gigaword – a large corpus of 5.7M news articles. We preprocessed Gigaword with MaltParser, extracted all pairs of nouns and verbs that were parsed as participants of the object-verb relation, and counted the frequency of occurrence of all the unique pairs. Finally, we indexed the resulting records of the form <frequency, verb, object> using the Lucene[2] indexing engine.

As an example, consider four nouns: *dinner, breakfast, lecture, child*. When used as the objects of *prepare*, the first three of them correspond to the instances of the sense 1 of *prepare*; the fourth one corresponds to an instance of the sense 2. With the help of our index, we can retrieve their DDNs. There is a considerable overlap among the DDNs of the first three nouns and a much smaller overlap between *child* and the first three nouns. E.g., *dinner* and *breakfast* have 34 DDNs in common, while *dinner* and *child* only share 14.

Once we have set up the framework for the extraction of DDNs, the algorithm for applying them to VSD is straightforward: (1) find the noun object of the ambiguous verb (2) extract the DDNs for that noun (3) sort the DDNs by frequency and keep the top 50 (4) include these DDNs in the feature vector so that each of the extracted verbs becomes a separate feature.

4 Relevant Work

At the core of our work lies the notion of distributional similarity (Harris, 1968), which states that similar words occur in similar contexts. In various sources, the notion of context ranges from bag-of-words-like approaches to more structured ones in which syntax plays a role. Schutze (1998) used bag-of-words contexts for sense discrimination. Hindle (1990) grouped nouns into thesaurus-like lists based on the similarity of their syntactic contexts. Our approach is similar with the difference that we do not group noun arguments into finite categories, but instead leave the category boundaries blurry and allow overlaps.

The DDNs are essentially a form of world knowledge which we extract automatically and apply to VSD. Other researches attacked the problem of unsupervised extraction of world knowledge: Schubert (2003) reports a method for extracting general facts about the world from treebanked Brown corpus. Lin and Pantel in (2001) describe their DIRT system for extraction of paraphrase-like inference rules.

5 Evaluation

We selected a subset of the verbs annotated in the OntoNotes project (Chen, 2007) that had at least 50 instances. The resulting data set consisted of 46,577 instances of 217 verbs. The predominant sense baseline for this data is 68%. We used

[1] In future, we will try to optimize this parameter

[2] Available at http://lucene.apache.org/

libsvm[3] for classification. We computed the accuracy and error rate using 5-fold cross-validation.

5.1 Experiments with a limited set of features

The main objective of this experiment was to isolate the effect of the novel semantic features we proposed in this paper, i.e. the DDN features. Toward that goal, we stripped our system of all the features but the most essential ones to investigate whether the DDN features would have a clearly positive or negative impact on the system performance. Lexical features are the most essential to our system: a model that includes only the lexical features achieves an accuracy of 80.22, while the accuracy of our full-blown VSD system is 82.88%[4]. Since the DDN features have no effect when the object is not present, we identified 18,930 instances where the target verb had an object (about 41% of all instances) and used only them in the experiment.

We built three models that included (1) the lexical features only (2) the lexical and the DDN features (3) the lexical and the object features. The object features consist of the head word of the NP object and the head word's POS tag. The object is included since extracting the DDN features requires knowledge of the object; therefore the performance of a model that only includes lexical features cannot be considered a fair baseline for studying the effect of the DDN features. Results are in Table 4.

Features Included in Model	Accuracy, %	Error Rate, %
Lexical	78.95	21.05
Lexical + Object	79.34	20.66
Lexical + DDN	82.40	17.60

Table 4. Experiments with object instances

As we see, the model that includes the DDN features performs more than 3 percentage points better than the model that only includes the object features (approximately 15% reduction in error rate). Also, based on the comparison of the performance of the "lexical features only" and the "lexical + DDN" models, we can claim that the knowledge of the DDNs provides richer semantic knowledge than just the knowledge of the object's head word.

5.2 Integrating the DDN features into a full-fledged VSD system

The objective of this experiment was to investigate whether the DDN features improve the performance of a full-fledged VSD system. We built two models which consisted of (1) the entire set of features (2) all the features of the first model excluding the DDN features. The entire data set (46K instances) participated in the experiment. Results are in Table 5.

Features Included in Model	Accuracy, %	Error Rate, %
All Features – DDN	82.38	17.62
All Features	82.88	17.12

Table 5. Performance of the full-fledged VSD system

The DDN features improved performance by 0.5% (3% drop in error rate). The difference between the accuracies is statistically significant (p=0.05).

5.3 Relative Contribution of Various Semantic Features

The goal of this experiment was to study the relative contribution of various semantic features to the performance of our VSD system. We built five models each of which, in addition to the lexical and syntactic features, included only certain type(s) of semantic feature: (1) WN (2) NE (3) WN and NE (4) DDN (5) no semantic features (baseline). All 46K instances participated in the experiment. The results are shown in Table 6.

Features Included in Model	**Accuracy, %**	**Error Rate, %**
Lexical + Syntactic	81.82	18.18
Lexical + Syntactic + WN	82.34	17.60
Lexical + Syntactic + NE	82.01	17.99
Lexical + Syntactic + WN + NE	82.38	17.62
Lexical + Syntactic + DDN	82.97	17.03

Table 6. Relative Contribution of Semantic Features

The DDN features outperform the other two types of semantic features used separately and in conjunction. The difference in performance is statistically significant (p=0.05).

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[4] Given this high baseline, we include error rate when reporting the results of the experiments as it is more informative

6 Discussion and Conclusion

As we saw, the novel semantic features we proposed are beneficial to the task of VSD: they resulted in a decrease in error rate from 3% to 15%, depending on the particular experiment. We also discovered that the DDN features contributed twice as much as the other two types of semantic features combined: adding the WN and NE features to the baseline resulted in about a 3% decrease in error rate, while adding the DDN features caused a more than 6% drop.

Our results suggest that DDNs duplicate the effect of WN and NE: our system achieved the same performance when all three types of semantic features were used and when we discarded WN and NE features and kept only the DDNs. This finding is important because such resources as WN and NE-taggers are domain and language specific while the DDNs have the advantage of being obtainable from a large collection of texts in the domain or language of interest. Thus, the DDNs can become a crucial part of building a robust VSD system for a resource-poor domain or language, given a high-accuracy parser.

7 Future Work

In this paper we only experimented with verbs' objects, however the concept of DDNs can be easily extended to other arguments of the target verb. Also, we only utilized the object-verb relation in the dependency parses, but the range of potentially useful relations does not have to be limited only to it. Finally, we used as features the 50 most frequent verbs that took the noun argument as an object. However, the raw frequency is certainly not the only way to rank the verbs; we plan on exploring other metrics such as Mutual Information.

Acknowledgements

We gratefully acknowledge the support of the National Science Foundation Grant NSF-0715078, Consistent Criteria for Word Sense Disambiguation, and the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022, a subcontract from the BBN-AGILE Team. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We also thank our colleagues Rodney Nielsen and Philipp Wetzler for parsing English Gigaword with MaltParser.

References

Jinying Chen, Dmitriy Dligach and Martha Palmer. 2007. Towards Large-scale High-Performance English Verb Sense Disambiguation by Using Linguistically Motivated Features. In *International Conference on Semantic Computing.* Issue , 17-19.

Jinying Chen and Martha Palmer. 2005. Towards Robust High Performance Word Sense Disambiguation of English Verbs Using Rich Linguistic Features. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing,* Korea.

Christiane Fellbaum. 1998. WordNet - an Electronic Lexical Database. The MIT Press, Cambridge, Massachusetts, London, UK.

Patrick Hanks, 1996. Contextual Dependencies and Lexical Sets. In *The Int. Journal of Corpus Linguistics*, 1:1

Zelig S. Harris. 1968. Mathematical Structures of Language. New York. Wiley.

Donald Hindle. 1990. Noun Classification from Predicate-Argument Structures. In *Proceedings of the 28th Annual Meeting of Association for Computational Linguistics*. Pages 268-275

Dekang Lin and Patrick Pantel. 2001. DIRT - Discovery of Inference Rules from Text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining*. pp. 323-328. San Francisco, CA.

Joakim Nivre, Johan Hall, Jens Nilsson, et. al. MaltParser: A language-independent system for data-driven dependency parsing. 2007. In *Natural Language Engineering*, 13(2), 95-135.

Lenhart Schubert and Matthew Tong, Extracting and evaluating general world knowledge from the Brown corpus. 2003. In *Proc. of the HLT/NAACL Workshop on Text Meaning*, May 31, Edmonton, Alberta, Canada.

Hinrich Schutze. 1998. Automatic Word Sense Discrimination. In *Computational Linguistics*, 24(1):97-123