Exploiting Semantic Relationships for Word Sense Disambiguation

Tyler Folkman Sabarish Kumar

May 15, 2015

Abstract

The task of word sense disambiguation has historically been approached by creating lexical and syntactic features from the context surrounding the word of interest. These methods have been deeply studied, but little work has been done on incorporating semantic relationships to help disambiguate the word of interest. We present a novel approach that leverages the Abstract Meaning Representation semantic parse of a sentence to derive semantic features. todo: We show that ...

1 Introduction

Word sense disambiguation (WSD) is the task of identifying the sense of a word with possible meanings based on the context in which the word is used. This task has been explored fairly deeply in part because of the Senseval[1] competitions. These competitions provide data with the purpose to evaluate the strengths and weaknesses of various WSD systems. Through studying these systems, it suggests that there is no best method for solving WSD systems. Instead, features and learning algorithms are dependent on one another[2]. In fact, Pedersen et al. have shown that a simple ensemble method using unigrams, bigrams, and part of speech features has the potential to achieve state of the art results[3]. We extend this work by exploring the effectiveness of including additional features based on the semantics of the context. To extract semantics we make use of a rooted, labeled graph called an Abstract Meaning Representation which aims to semantically represent a sentence[4]. todo: summarize results and conclusions and novelness / why important / interesting

2 Abstract Meaning Representation

Abstract Meaning Representation (AMR) aims to assign the same AMR to sentences that have the same basic meaning. For example, the sentences, "The man described the mission as a disaster" and "As the man described it, the mission was a disaster." should both produce the same AMR. This AMR looks like this:

```
(d / describe-01
    :arg0 (m / man)
    :arg1 (m2 / mission)
    :arg2 (d / disaster))
```

To help do this, the AMR representation makes use of PropBank framesets. To create the AMR parse for our sentences we make use of the pre-trained JAMR parser[5]. This parser was developed by Jeffrey Flanigan from Carnegie Mellon University and is pre-trained on 18,779 sentences.

3 Problem Definition and Algorithm

todo: Describe in reasonable detail the algorithm you are using to address this problem. A psuedocode description of the algorithm you are using is frequently useful. Trace through a concrete example, showing how your algorithm processes this example. The example should be complex enough to illustrate all of the important aspects of the problem but simple enough to be easily understood. If possible, an intuitively meaningful example is better than one with meaningless symbols.

3.1 Lexical and Syntactic Features

To understand whether or not semantic features extracted from the AMR parse help improve current systems we build a baseline model based on the

Syntalex system developed by Pedersen et al[6]. To provide a concrete example, consider the following sentence:

The accident appeared to have little effect on the Christmas party, except to lengthen it considerably.

In this sentence the word of interest is accident. The sentence or sentences provided with the word of interest are called the context. For our baseline system we extract unigrams and bigrams from the context as our lexical features. We also extract the part of speech tags from the word of interest and the two words before and after for the sentence that contains the word of interest. If no words are available we tag it as unknown. Lastly, we extract the following features from the syntactic parse of the sentence containing the word of interest: the head word of the phrase housing the word of interest, the head word of its parent phrase, the phrase housing the target word, and the parent phrase. To get the syntactic parse and the parts of speech we make use of the Stanford parser[7]. To make our parse features more clear, in our example, the head word and head word of the parent are both 'The', the phrase housing the word of interest is 'NP' , and the parent phrase is 'S.'

3.2 Semantic Features

todo: describe the semantic features we extract

4 Experimental Evaluation

4.1 Methodology

To evaluate our results, we make use of the Senseval-1 data. We obtained a fixed version of the data[8] from Ted Pedersen and used the gold standard data and mappings from Senseval's website[1]. These data contain 35 words in English with noun, verb, adjective, and indeterminate parts of speech. The indeterminate words were words for which the goal was to also determine the major word class. We ignored these words for our purposes and selected the following five words to test our results: excess, the verb version of float, brilliant, accident, and the verb version of promise. These words were selected because they provide at least one word from each part of speech and represent words with small and large amounts of training data.

Word	Part of Speech	Training Size	Testing Size
Excess	N	178	186
Float	V	183	229
Brilliant	A	441	229
Accident	N	1,234	267
Promise	V	1,163	224

todo: What are criteria you are using to evaluate your method? What specific hypotheses does your experiment test? Describe the experimental methodology that you used. What are the dependent and independent variables? What is the training/test data that was used, and why is it realistic or interesting? Exactly what performance data did you collect and how are you presenting and analyzing it? Comparisons to competing methods that address the same problem are particularly useful.

4.2 Results

As an additional baseline, we also include the average scores from the Senseval-1 competiton[1]. We compare on fine-grained recall results for S systems. We use fine-grained results because we only count scores for identical matches and S systems because we are doing supervised training. Lastly, we evaluate our results using recall which is defined as the percentage of right answers on all instance in the test set[9].

todo: also include senseval best system?

Word	Senseval Average
Excess	0.652
Float	0.402
Brilliant	0.443
Accident	0.802
Promise	0.741

todo: Present the quantitative results of your experiments. Graphical data presentation such as graphs and histograms are frequently better than tables. What are the basic differences revealed in the data. Are they statistically significant?

4.3 Discussion

todo: Is your hypothesis supported? What conclusions do the results support about the strengths and weaknesses of your method compared to other methods? How can the results be explained in terms of the underlying properties of the algorithm and/or the data.

5 Related Work

todo: Answer the following questions for each piece of related work that addresses the same or a similar problem. What is their problem and method? How is your problem and method different? Why is your problem and method better?

6 Future Work

todo: What are the major shortcomings of your current method? For each shortcoming, propose additions or enhancements that would help overcome it.

7 Conclusion

todo: Briefly summarize the important results and conclusions presented in the paper. What are the most important points illustrated by your work? How will your results improve future research and applications in the area?

References

[1] http://www.senseval.org/

[2] K.L. Lee and H.T. Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 41?48.

[3] Pedersen, Ted, Mohammad, Saif "Combining Lexical and Syntactic Features for Supervised Word Sense Disambiguation." Proceedings of the

Conference on Computational Natural Language Learning (CoNLL), May 6-7, 2004, Boston, MA

[4] "Abstract Meaning Representation for Sembanking" (L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, N. Schneider), Proc. Linguistic Annotation Workshop, 2013.

[5] JAMR - AMR Parser: https://github.com/jflanigan/jamr

[6] Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3), pp. 159-162, July 25-26, 2004, Barcelona, Spain.

[7] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

[8] http://www.d.umn.edu/ tpederse/data.html

[9] Edmonds, Philip. "SENSEVAL: The evaluation of word sense disambiguation systems." ELRA newsletter 7.3 (2002): 5-14.