

A Semi-supervised Approach to Word Sense Disambiguation (CS446 Class Project)

Haoruo Peng(hpeng7@illinois.edu)

Shyam Upadhyay(upadhya3@illinois.edu)

December 2, 2013

Abstract

We investigate the task of word sense disambiguation using a semi-supervised approach. Word sense disambiguation is the task of identifying the correct sense for a word which can possibly admit multiple senses. We use the Semeval-2007 dataset to evaluate the performance of our system. We use a graph based approach to disambiguate senses; the underlying assumption is that the sense of the neighboring words can assist in inferring the sense of the target word. In this respect, the senses are learnt in a somewhat joint-manner. Using appropriate weights in the graph constructed for all possible senses, we can use the shortest path algorithm to obtain the correct senses.

1 Introduction

In natural language, a word may be associated with possibly multiple meanings, depending on the context in which the word occurs. For instance, the word *pen* has the following senses according to Wordnet [5]:

- *pen* : a writing implement with a point from which ink flows
- *pen* : an enclosure for confining livestock

Word sense disambiguation(WSD) is the problem of determining the correct sense of a word in a given sentence. For example, determining the correct sense of the word “pen” in the following passage: *Little John was looking for his toy box. Finally he found it. The box was in the pen. John was very happy.* WSD can be viewed as a multi class classification problem, where each word admits several possible senses and the task is to identify the correct sense of a given word given its context and knowledge sources. The inherent difficulty of WSD is evident from the fact that the target classes change for each word in the lexicon. In this respect, WSD involves training n different classifiers, one for each word in a lexicon of size n . WSD is a key component for natural language processing systems which involves semantic interpretation of text. Tasks such as machine translation, information retrieval, data mining, web data analysis can greatly benefit from text disambiguation tools.

2 Background

WSD has been described as an AI-complete problem [2]. Researchers have done much progress to WSD problem achieving sufficiently high levels of accuracy on a variety of word types and ambiguities. A rich variety of techniques have been used, from dictionary-based methods [4] that use the knowledge encoded in lexical resources, to supervised machine learning methods [3] in which a classifier is trained for each distinct word on a corpus of manually sense-annotated examples, to completely unsupervised methods [10] that cluster occurrences of words, thereby inducing word senses.

Supervised learning approaches have been remarkably successful for performing word sense disambiguation. But the lack of sense-tagged data poses a severe bottleneck. To address this problem, researchers have resorted to semi-supervised learning algorithms. The Yarowsky algorithm [10] was an early example of such an algorithm. It uses the ‘One sense per collocation’ and the ‘One sense per discourse’ properties of human languages for word sense disambiguation. From observation, words tend to exhibit only one sense in most given discourse and in a given collocation. The common motif in approaches like [10], [1] is that they allow both labeled and unlabeled data. Training involves iteratively labeling the unlabeled data using an initial seeding labeled set. The seed labeled data is used to learn a classifier which is used to assign labels to unlabeled data, which is merged with the initial seed data to obtain a larger dataset for the next iteration. Finally, the classifier is learned from the extended labeled dataset. More recently, researchers [8] have leveraged word alignment information from parallel corpora to aid in obtaining coarse grained senses. The aim of word-alignment task is to align a word in a given sentence to its translated counterpart in another sentence in a different language. Unlike sense-tagged datasets, good quality parallel corpora are readily available.

Other semi-supervised techniques use large quantities of untagged corpora to provide co-occurrence information that supplements the tagged corpora. These techniques have the potential to help in the adaptation of supervised models to different domains. The reader is encouraged to peruse a more general survey provided in [7].

3 Task and Data

Now describe the task and data in more detail.

3.1 The Task

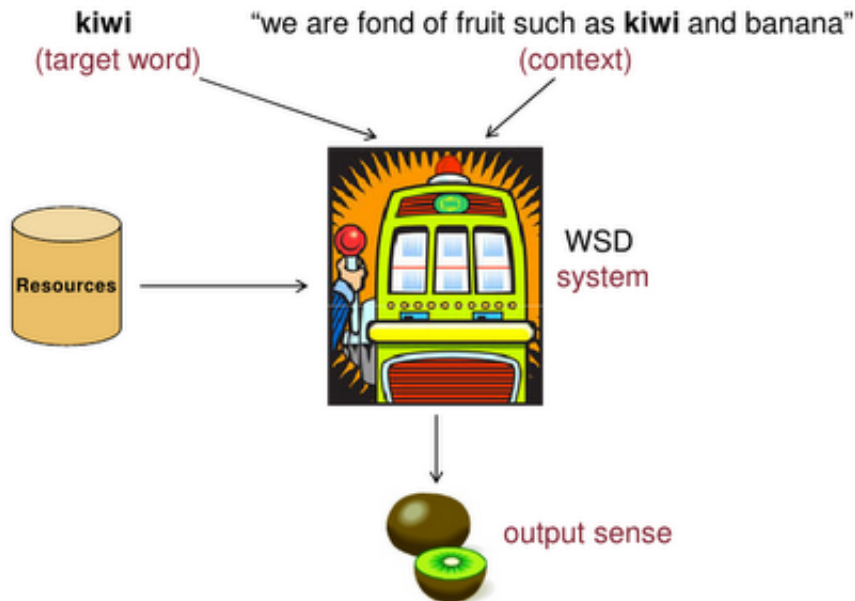


Figure 1: An example of word sense disambiguation system. Image courtesy [6]

We are given a document D and sequence of target words $(w_1, w_2 \dots w_n)$ in the document. Each word w_i admits a set of candidate senses $S_i = (s_{i1}, s_{i2} \dots s_{in_i})$ where $|S_i| = n_i$. The task is to assign each word w_i

the most appropriate sense from its context in D .

3.2 The Data

Describe the data you use to train and evaluate your models. Describe where you got it from (include references/citations to published works, or URLs!). Describe and give examples for the features that you have access to. We use datasets from **SemEval-2007** "<http://www.senseval.org>". We choose a specific dataset from "Task # 7: Coarse-grained English all-words (Coarse AW)" to do both the training and testing. It is an ongoing series of evaluations of computational semantic analysis systems; it evolved from the Senseval word sense evaluation series. In the chosen dataset, it tags approximately 6,000 words of five running texts with coarse senses. Coarse senses will be based on a clustering of the WordNet sense inventory obtained via a mapping to the Oxford Dictionary of English (ODE), a long-established dictionary which encodes coarse sense distinctions. The dataset contains the coarse-grained sense inventory which is prepared semi-automatically: starting from an automatic clustering of senses produced by Navigli (2006) with the Structural Semantic Interconnections (SSI) algorithm, and then manually validate the clustering for the words occurring in the text. For each content word the dataset provides lemma and part of speech.

Our work also requires some supplemental materials such as "Wordnet" database and "WS4J" library. We use the the latest released version for linux-Wordnet 3.0 from <http://wordnet.princeton.edu/wordnet/download/current-version/>. We compute the similarity metrics by utilizing WS4J library from <https://code.google.com/p/ws4j/>.

4 The Models

4.1 Baseline Models

In order to know how difficult the task is and how well we are doing, we need to know how well a suitable baseline model would perform. Define a baseline model for your task. This may not necessarily be a learned model. A naive baseline for word sense disambiguation will be to output the most frequent sense of the word. On coarse grained senses, using this baseline gives an accuracy ranging from 50-60%.

4.2 Existing Models

If people have worked on this task before, summarize (and cite) some of the existing models It-makes-sense

4.3 Proposed Model(s)

Your models and your procedure for learning them go here. Describe both in detail, even if the learning procedure is standard. We employ a graph-based approach to disambiguate senses of a word. Our method is similar to [9]. We first create a weighted graph of label dependencies for all candidate senses. For every candidate sense s_{ij} for word w_i , we have a node in the graph. This node has edges to the all candidate senses of words within a pre-defined window around word w_i . These edges are weighed by the dependency score for the pair of senses. The dependency score captures the relationship between two candidate senses for nearby words. Based on the edge weights we assign scores to each node based on a graph-based measure of centrality. The score of a node denotes the "importance" of the node in the graph, taking into account its relationship with neighboring nodes.

5 Experiments

5.1 Experimental Hypotheses

Summarize the hypotheses (research questions) your experiments are designed to test (address). (Note that some of these hypotheses may emerge as you keep working on a problem; you will not necessarily have come up with all the questions you wish to address before you have started building a models for the specific task.

5.2 Experimental setup

Define test/training/dev data splits, describe how you tuned performance. Describe and your evaluation metric, and define it mathematically. List the models you will evaluate. Cite any existing tools or software you use to perform your experiments; describe what you implemented yourself. Describe how you obtained the features used by each of the models. mention maven

5.3 Experimental results

Now give the actual experimental results (use figures/tables/graphs as appropriate), and discuss whether they verify or falsify your hypotheses. How important are the various features your models use (consider ablation studies). How robust are your results? (Look at learning curves, or the variance when you perform cross-validation). Can you perform an error analysis?

6 Conclusion

Summarize your findings, and discuss their implications, e.g. for future work, or for related tasks. Discuss also the shortcomings of your proposed approach. .

Your current to-do list

Done

1. Implemented the code.
2. Obtained the Semeval 2007 dataset.
3. Added relevant related work.

Left to do

1. Running experiments with different similarity metrics.
2. Write-up.

References

- [1] Anh-Cuong Le, Akira Shimazu, Van-Nam Huynh, and Le-Minh Nguyen. Semi-supervised learning integrated with classifier combination for word sense disambiguation. *Computer Speech & Language*, 22(4):330–345, 2008.

- [2] John C Mallery. Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers. In *Master's thesis, MIT Political Science Department*. Citeseer, 1988.
- [3] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.
- [4] Rada Mihalcea. Using wikipedia for automatic word sense disambiguation. In *HLT-NAACL*, pages 196–203, 2007.
- [5] George A. Miller. Wordnet: A lexical database for english. 1995.
- [6] Roberto Navigli. Lecture 7. <http://naviglinlp.blogspot.com>.
- [7] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.
- [8] Hwee Tou Ng, Bin Wang, and Yee Seng Chan. Exploiting parallel texts for word sense disambiguation: An empirical study. In *In Proceedings of ACL03*, pages 455–462, 2003.
- [9] Ravi Sinha and Rada Mihalcea. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of the International Conference on Semantic Computing, ICSC '07*, pages 363–369, Washington, DC, USA, 2007. IEEE Computer Society.
- [10] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995.