Thesis title

Osman Başkaya

September 2014

Contents

Chapter 1

Introduction

Chapter 2

Related Works

Chapter 3

System

3.1 Pipeline

3.2 Different Representations

3.3 Different approaches

Chapter 4

Evaluation Framework for WSI Systems

4.1 Agirre's method

4.2 Complex Classifiers for mapping and some discussion

Chapter 5

Experiments on various tasks

5.1 WSI tasks

5.1.1 SemEval 2010 WSI Task

TODO: LM details should be explained somewhere above.

Dataset

The test dataset is part of OntoNotes **?**. The texts come from various news sources including CNN, ABC and others. The test set for this task consists of 100 words, 50 nouns and 50 verbs; 5285 noun instances and 3630 verb instances, 8915 instances in total. Average number of sense for nouns is 4.46 and for verbs is 3.12. The following is an example for the test data instance.

> $< swim.v.1 >$
> First of all , visibility will be very very low . $< TargetSentence >$ It 's going to be bitterly cold , and there is going to be enormous danger from jagged pieces of metal which could be swimming around in the submarine . $< /TargetSentence >$ Given the fact that there are so many dangers and that these divers are risking their own lives , I wonder if there is consideration given to the fact that this may not be really worth it .
> $< /swim.v.1 >$

Baselines

Most Frequent Baseline: Task organizers provided the FScore of most frequent baselines (MFS), which are 0.532, 0.666 and 0.587 for nouns, verbs, and all words, respectively.

Random Baselines: I calculated two types of random baselines: Random induced sense baseline and usual random baseline that uses the gold standard sense inventory. The first one is a dummy system that provides random induced senses for each instance. Since this random baseline provides arbitrary senses (i.e., the sense inventory is not the same with the gold standard sense inventory), the mapping between these induced senses and gold standard senses needs to be done. The number in the name of the baseline indicates how many different induced senses are provided for each target word. The other type of random baseline uses the correct number of sense for each target word and randomly picks a sense among those senses provided in gold standard. Results for random baselines can be seen here.

kNN-baselines for substitute vectors: These baselines are computed as follows. First, the most frequent 100 substitutes and their probabilities are found for each test instance using *FASTSUBS* algorithm **?** and a language model that built by using ukWaC **?** as corpus and SRILM **?** as a language model library. These 100 substitutes is not a probability distribution and needs to be normalized. After normalization, I obtained legitimate probability distributions and each instance is represented by its substitute distribution. Using various distance metrics (euclid, cosine, manhattan, maximum, jensen), I found the closest neighboring test instances and their distances for each instance. The two types of kNN baseline are calculated: *majority voting* and *minumum average distance*. The first type is usual kNN. Using the answers (gold standard for test data), it decides the sense of the current instance by looking labeled senses of k neighboring instances. This version does not consider the distance values. That is, the weights of the each neighbors are equal in sense deciding process. It returns the majority sense as the predicting sense. The other baseline differs from the first and it takes into account the distance between the neighbors and the instance whose sense is in question. It returns the sense that has the minimum average distance among the senses that k closest neighbors of the instance have. Results can be seen in details

kNN-baseline for embeddings: Scores for embeddings can be seen here.

5.1.2 SemEval 2013 WSI Task

Dataset

The test data for the graded word sense induction task in SemEval-2013 includes 50 terms containing 20 verbs, 20 nouns and 10 adjectives. There are a total of 4664 test instances provided. All evaluation was performed on test instances only. In addition, the organizers provided sense labeled trial data which can be used for

tuning. This trial data is a redistribution of the Graded Sense and Usage data set provided by Katrin Erk, Diana McCarthy, and Nicholas Gaylord **?**. It consists of 8 terms; 3 verbs, 3 nouns, and 2 adjectives all with moderate polysemy (4-7 senses). Each term in trial data has 50 contexts, in total 400 instances provided. Lastly, participants can use ukWaC **?**, a 2- billion word web-gathered corpus, for sense induction. Furthermore, unlike in previous WSI tasks, organizers allow participants to use additional contexts not found in the ukWaC under the condition that they submit systems for both using only the ukWaC and with their augmented corpora. The gold-standard of test data was prepared using WordNet 3.1 by 10 annotators. Since WSI systems report their annotations in a different sense inventory than WordNet 3.1, a mapping procedure should be used first. The organizers use the sense mapping procedure explained in **?**. This procedure has adopted the supervised evaluation setting of past SemEval WSI Tasks, but the main difference is that the former takes into account applicability weights for each sense which is a necessary for graded word sense. Although the data contains graded senses for some instances, I used only the instances that labeled one sense. This exclusion decreased the number of instances used in the experiments to 4122. The following section contains baseline scores for the test data on single sense instances.

Baselines

Most Frequent Baseline: Task organizers provided the FScore of most frequent baseline for single sense data, which is 0.578.

Random Baselines: I used the same procedure to calculate two types of baselines, summarized in 5.1.1. Results for random baselines can be seen here.

kNN-baselines for substitute vectors: As explained in 5.1.1, I followed the same procedure to calculate these baselines and results can be seen in details.

kNN-baseline for embeddings: Scores for embeddings can be seen here.

5.1.3 Pseudoword Dataset

5.2 All-Word task

5.3 Semantic Similarity task

Chapter 6

Sensitivity Analysis

Chapter 7

Conclusion