Word Sense Disambiguation using Supervised and Semi-Supervised Techniques

Tejashree Khot
Indiana University
tpkhot@iu.edu

Sanjana Agarwal
Indiana University
sa14593@iu.edu

Abstract

This paper describes an approach to Romanian word sense disambiguation task for Senseval-3. The contributions of this paper are: (1) Romanian word sense disambiguation based on supervised machine learning algorithms. (2) Romanian word sense disambiguation based on semi-supervised machine learning algorithm.

1 Introduction

This paper describes to build a Romanian word sense disambiguation task for Senseval-3 from the Romanian Lexical Sample data set. We have developed a classifier for each of the following words: accent.n, citi.v, delfin.n, oficial.a, and val.n. We have built a feature set and optimized its parameters non-exhaustively. We have also made use of the official scorer to evaluate our results. The first section will describe the related work and the learning algorithms that we have used. The second section contains the methodology used, and how we approached the supervised and the semi-supervised task. Finally, the last section includes the results, conclusion and future work.

2 Related Work

Furthermore, Xuri Tang et al. (2010), proposes a semi-supervised approach for WSD in Word-Class based selectional preferences. There are four types of formalization models employed: Word model, Class model, Class-only model, Word-class model. The approach exploits syntagmatic semantic redundancy (among all possible sense collocations for a word collocation, the most appropriate is the one in which senses exhibit the most redundant information between each other) and paradigmatic semantic redundancy (among all possible sense collocations for a word collocation, the most appropriate is the one which is also implicitly or explicitly expressed by other synonymous, metonymic or metaphorical word collocations) in the semantic system and uses association computation and minimum description length for the task of WSD. The experiments show that the approach proposed is fairly encouraging in disambiguation of polysemous predicates,

especially under semi-supervised conditions when a small portion of data is annotated.

Innovative approaches like feature clustering with semi-supervised algorithms was also done in the past. Zheng-Yu Niu et al. (2005) presents one such approach. They proposed a semisupervised feature clustering algorithm. It can deal with both seen and unseen features in feature clustering process. For the problem, they've considered a total of n ambiguous words. The first l words are the labeled words and the remaining words are unlabeled. Thus, with the help of a semi-supervised algorithm they've predicted the sense of the ambiguous word by the use of the label information and similarity information among the words. The experimental results on SENSEVAL-3 data showed that feature clustering aggressively reduced the dimensionality of feature space while still maintaining state of the art sense disambiguation accuracy. Furthermore, when combined with a semi-supervised WSD algorithm, semi-supervised feature clustering outperformed supervised feature clustering and other dimensionality reduction techniques. Additional experiments on sampled SENSEVAL-3 data indicated that the semi-supervised feature clustering method is robust to the noise in small labeled data, which achieved better performance than supervised feature clustering.

Some have tried to approach the problems of semi-supervised methods (eg. Cuong Le et al. (2006)). In this paper they've showed two problems of semi-supervised learning for self-training algorithm. Thus, they've built a new bootstrapping algorithm with several variants based on the problems. Like, to determine a subset of new labeled examples at each extension of labeled data (the first problem), they used Naive Bayes classifier, and SVM to decrease error rate of new labeled examples. And to determine how to generate the final classifier when the process of extending labeled data is completed (the second problem), they used two strategies of classifier combination including median and max rules to utilize both advantages of the last classifier (built based on the extended labeled data) and the initial supervised classifier. Thus, Naive Bayes gave an improved accuracy over SVM for about 1.9% and the proposed solutions were effective for improving semi-supervised learning.

Dinu, Georgiana, and Sandra Kübler (2007) wrote a paper on Romanian WSD and experiments with features of memory-based classifier. It states that the feature set needs to be tightly controlled and achieved optimal results with on average seven features per word. This occurs because MBL methods are affected by irrelevant or redundant features. However, this could be caused by idiosyncrasies in the Romanian data set or limited size of the training data.

3 Learning Algorithms

3.1 Supervised Learning

The data that we have, consists of labels. The purpose of these labels is that when via a training method, we make predictions, some of these predictions can go wrong. Thus, with the help of labels, we can correct the wrongly classified items. Examples are: Classification and regression. Detecting whether the mail is spam/not-spam. Predicting housing price when we are given the data including the price and square feet of the house.

3.2 Semi-supervised Learning

Semi-supervised learning is a class of supervised learning tasks and techniques that also make use of unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. It falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data).

3.3 Support Vector Machines (SVMs)

Support Vector Machines are supervised machine learning models used for classification and regression. In SVMs, the separate classes are separated by a clear gap which is made as wide as possible. The new data points or examples are then classified into the classes depending on the side of the gap they fall in. The vectors or data points at the boundary conditions are known as the support vectors.

3.4 Label Propagation and Label Spreading

Within complex networks, real networks tend to have community structure. Label propagation is an algorithm for finding communities. It has two built-in kernel methods (RBF and KNN). Choice of kernel effects both scalability and performance of the algorithms. Label Spreading is similar to the basic Label Propagation algorithm, but uses affinity matrix based on the normalized graph Laplacian and soft clamping across the labels.

4 Methodology

4.1 Choice of Machine Learning Classifier

In the initial experiments conducted by our team, we applied Scikit-Learn's SVM classifier to perform word sense disambiguation on accent.n, citi.v, delfin.n, oficial.a, and val.n. We observed that SVM gives a good accuracy for the given words. The results will be tabulated in the further sections. In this paper, we have performed an exhaustive study of word sense disambiguation using SVM.

4.2 Feature Selection

As our full context features, we have used SVM Bag of Words. Along with Bag of Words, we have added an additional feature 'relative position,' that gives the relative position of the particular word in the sentence.

For immediate context features, we have added Part of Speech (POS) tagging of the word's two immediate neighbors.

Thus, with two neighboring words, the feature set becomes: w_{-2}, w_{-1}, w_1, w_2 and bag-of-words.

The graph showed in Fig. 1 compares the accuracies of:

1) Full context features.
2) Full context features + Immediate Context Features.

Thus, we see that adding POS tagging to the Bag of words, increases the accuracy by a considerate amount in some cases while keeping it the same in some.

4.3 Approach

We have preprocessed the files and created new files with the required fields only (Instance ID, sense ID, lexelt item, etc). Next, with the training and the test

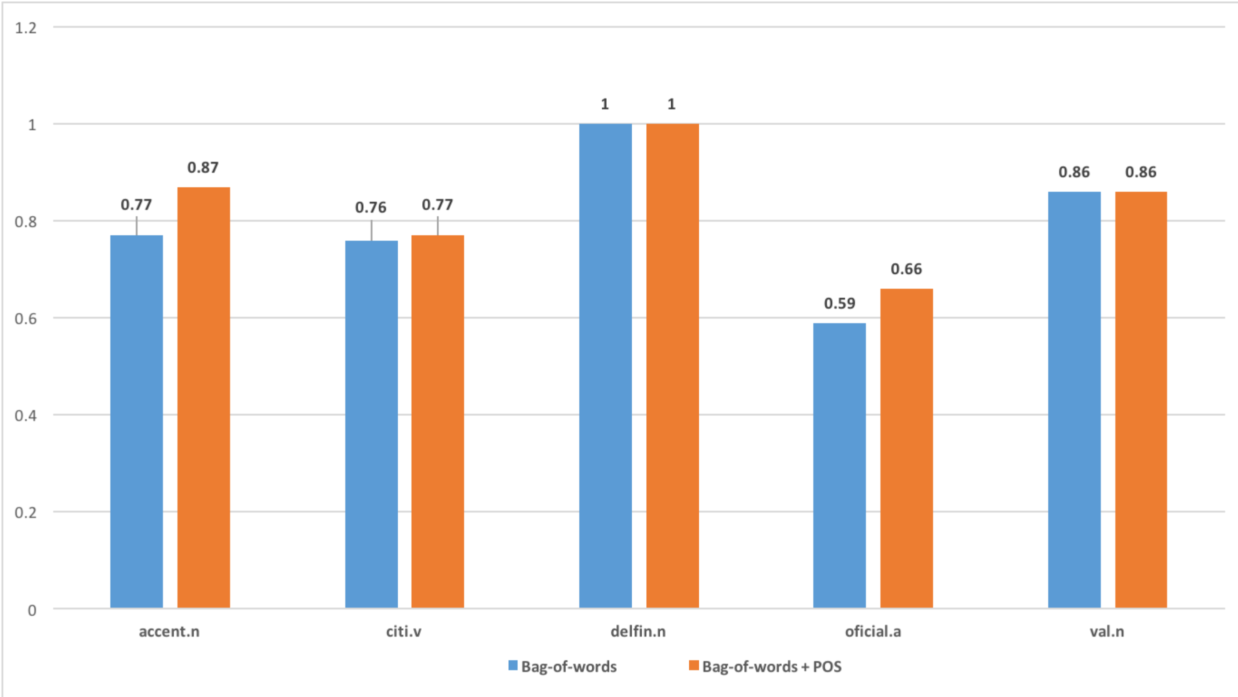

Fig 1. Accuracies of the five words for different feature sets.

files and the given key file, we have created a word specific test file. After that, we applied the SVM classifier on the test file that generated an answer file. It also gave us the corresponding accuracy after applying the SVM classifier. This similar approach was used to perform the same tasks for POS tagging.

5. Experiments

5.1 Dataset

We have used the Romanian Lexical Sample dataset. It consists of 50 words, that cover all the open class parts of speech, with various degrees of ambiguity, and for each such word collect a set of examples from a large Romanian corpus. For our experiment, we are given three nouns, one adjective and one verb.

Nouns – accent, delfin, val.
Adjective – oficial
Verb – citi

6. Approach to Semi-Supervised Task

The above mentioned approaches have been used for semi-supervised task as well. We have used SVM classifier to get the accuracy of the unlabeled data.

Our first task for semi-supervised approach is to label the unlabeled data. This can be achieved using two techniques provided by scikit-learn, namely, Label Propagation and Label Spreading. We have used both on different words to check for different accuracies.

Also, both these algorithms support two kernels, Radial Basis Function kernel and k-Nearest Neighbor kernel. Thus, to label the unlabeled data, we had to follow a similar approach. Firstly, we used the unlabeled training file to generate the unlabeled file specific to a word. After that, we used the same file with the training file of the word, and applied Label Propagation/Label Spreading to it, to fit the semi-supervised model. This generates an output file which is used with the training file and SVM is applied on it to get the accuracy. The results will be discussed in the next section.

7. Results

The results are calculated in three parts. We have applied the scorer function on each of the results to obtain a better score. The scorer function takes three parameters: fine-grained, coarse-grained, mixed-

	accent.n	citi.v	delfin.n	oficial.a	val.n
Baseline	0.736	0.746	1.0	0.51	0.835
Full Context Features (without scorer)	0. 770	0.76	1.0	0.594	0.863
Full Context Features (with scorer)	i. 0.883 ii. 0.883 iii.0.883	i. 0.837 ii. 0.922 iii.0.853	i. 1.0 ii. 1.0 iii.1.0	i. 0.635 ii. 0.635 iii.0.635	i. 0.88 ii. 0.88 iii.0.88
Bag of Words + POS (without scorer)	0.896	0.767	1.0	0.66	0.863
Bag of Words + POS (with scorer)	i. 0.896 ii. 0.896 iii.0.896	i. 0.845 ii. 0.922 iii.0.86	i. 1.0 ii. 1.0 iii.1.0	i. 0.708 ii. 0.708 iii.0.708	i. 0.88 ii. 0.88 iii.0.88

Table 1: Results for supervised approach. i. Fine-grained ii. Coarse-grained iii. Mixed-grained

	accent.n	citi.v	delfin.n	oficial.a	val.n
Baseline	0.736	0.746	1.0	0.51	0.835
Semi Supervised (without scorer)	0.896	0.76	1.0	0.66	0.863
Semi Supervised (with scorer)	i. 0.896 ii. 0.896 iii.0.896	i. 0.837 ii. 0.922 iii.0.853	i. 1.0 ii. 1.0 iii.1.0	i. 0.708 ii. 0.708 iii.0.708	i. 0. 863 ii. 0. 863 iii.0. 863

Table 2: Results for semi-supervised approach. i. Fine-grained ii. Coarse-grained iii. Mixed-grained

grained. We have tabulated the results for all three in Table 1. Some of the parameters were tweaked for each word differently.

We used scikit-learn over Timbl because it is more user-friendly and we can perform the semi-supervised approaches more easily with scikit-learn than with Timbl. As you can see from the results, of the supervised approach, the POS tagged features+Bag of Words most of the times gives better results. Also, the scorer function enhances the results even more.

Table 2, summarizes the results of the semi-supervised approach and as we can see, the scorer function gives more improvement. In some cases, the semi-supervised gives a better accuracy, while in some equivalent and only in one case it gives a lesser accuracy.

8. Conclusion

In this project, we have done a study on Support Vector Machines and used it for both supervised and semi-supervised algorithms. We have got good accuracies using SVM and we believe, that it is one of the best approaches available to use. The POS tags helped to improve the performance in most of the cases.

Thus, we believe that there is no perfect algorithm made for word sense disambiguation and that we need to try various algorithms and compare their performance to know which one gives better results, depending on the size and the type of the dataset.

9. Future Work

As future work, we would like to clean the data more and remove the delimiters and the stop words. We would also add some more features and see the comparison of selecting more than two immediate neighbors. We have also thought of comparing the performance of SVM with other classification algorithms.

10. Acknowledgements

We would like to thank our Professor Sandra Kubler for helping us with the project and help us explore the intersecting world of machine learning and computational linguistics. We would also like to thank the University Information Technology Services for allowing us access to Karst.

11 References

Xuri Tang, Xiaohe Chen, Weiguang Qu, Shiwen Yu, *Semi-supervised WSD in selectional preferences with semantic redundancy*, Association for Computational Linguistics Stroudsburg, PA, USA

Zheng-Yu Niu, Dong-Hong Ji, Chew Lim Tan, *Word sense disambiguation using label propagation based semi-supervised learning,* Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics

Anh-Cuong Le, Akira Shimazu, and Le-Minh Nguyen, *Investigating Problems of Semi-Supervised Learning for Word Sense Disambiguation,*

Georgiana Dinu and Sandra Kübler, *Sometimes Less Is More: Romanian Word Sense Disambiguation Revisited*

https://en.wikipedia.org/wiki/Supervised_learning

http://scikit-learn.org/stable/modules/label_propagation.html#label-propagation

https://en.wikipedia.org/wiki/Semi-supervised_learning