Word Sense Disambiguation for Romanian Lexical Data

B659 : Final Project

Arpit Khandelwal
School of Informatics and Computing
Indiana University
Bloomington, Indiana
arkhande@indiana.edu

Venkatesh Raizaday
School of Informatics and Computing
Indiana University
Bloomington, Indiana
vraizada@indiana.edu

Dakshi Kumar
School of Informatics and Computing
Indiana University
Bloomington, Indiana
dakumar@indiana.edu

***Abstract*—In this paper, we talk about an ensemble based approach for word sense disambiguation and how such an approach intuitively gives a method for classifying unlabeled data with high confidence which in turn can be used as a semi supervised method for WSD. We chose Lesk based classifier, support vector machine and memory based learning as the components of our ensemble.**

Keywords—word sense disambiguation; ensemble approach; MBL; support vector machine; lesk algorithm;

I. Introduction

Word sense disambiguation can be defined as selecting the most appropriate meaning for a word, based on the context in which it occurs. For example, suppose *bill* has the following set of possible meanings: a piece of currency, pending legislation, or a bird jaw. When used in the context of *The Senate bill is under consideration*, a human reader immediately understands that *bill* is being used in the legislative sense. Since, a computer does not have the liberty of having a language model or common sense, we need techniques to perform the task of word sense disambiguation for it.

In this paper, we try to implement an ensemble based approach for WSD by using multiple individually strong classifiers and grouping their results using a majority classifier to see if the performance is indeed boosted by this process. Another research aspect is to use such a model to confidently predict work senses of unannotated data and in turn use them for training our classifiers to see if their performance improves.

In order to investigate the problem we use the SensEval-3 Romanian Lexical Sample data set for our experiments. The results show that using an ensemble approach can increase the performance of individually strong classifiers. Although our experiment with neural networks as ensemble did not give expected results but something as simple as a majority rule improves the performance substantially for some words. Adding new data using semi supervised learning also improves the performance. The improvement is not substantial but the increase is consistent over all words in the study.

II. Related work

The task of WSD is a historical one in the field of Natural Language Processing (NLP). In fact, it was conceived as a fundamental task of Machine Translation (MT) already in the late 1940s [Weaver 1949]. During the 1970s the problem of WSD was attacked with AI approaches aiming at language understanding (e.g., Wilks [1975]) and during the 1990's massive employment of statistical methods took place. The 2000's brought an advent of applying machine learning techniques to NLP problems and since then a multitude of supervised, unsupervised and semi supervised techniques have been used to conquer the issue of word sense disambiguation. We focus our attention to the various semi supervised approaches for word sense disambiguation.

One of the earliest highlighted work was done by Mihalcea and Moldovan [1999]. The paper uses search on the Internet data to obtain sense tagged corpora. They used WordNet information to formulate queries consisting of similarity list or definitions of word senses, and obtained additional training data for word senses from Internet using existing search engine.

The popularity of semi supervised methods for WSD can be attributed to the requirement of large amount of annotated data for supervised methods as stated in Su et al [2004] which uses a statistical method KPCA (Kernel Principal Component Analysis) to project input vector from their original space R^n to a high dimensional feature space F. The approach is similar to that for support vector machines. Niu, Ji and Tan [2005] use a feature clustering algorithm for dimensionality reduction of features to improve model scalability and incorporate WSD model in NLP systems. Cuong Le et al [2006] focuses on an approach to enlarge the labelled data which is obtained from unlabeled data. They classify the unlabeled data based on any supervised learning algorithm, the ones with high confidence are then added to the labelled data. To decrease the error rate they have used one more classifier. This is done iteratively until a pre-specified value is reached. Then they have built a new classifier based on the median and max rule that take advantage of initial supervised classifier and the last classifier used to build extended labeled data.

Zheng, Ji and Tan (2007) proposed a method to naturally partition of mixed data (labelled data+ unlabeled data) by maximizing a stability criterion defined on classification results from an extended label propagation algorithm over all the possible values of model order (or the number of classes) in mixed data. It uses a model order identification based partially supervised classification algorithm, which can classify unlabeled data into positive and negative examples, and further group negative examples into a natural number of clusters. The model order identification algorithm with the extended label propagation algorithm as base classifier outperforms SVM and the model order identification algorithm with semi-supervised k-means clustering as the base classifier when labeled data is incomplete.

Data sparsity is one of the problems that has made WSD a challenging task and memory based learning is specifically suited to such problems. Dinu and Kubler [2007] present a memory based technique which chooses a restricted set of features which is further shortened using feature selection. This approach outperformed all systems that participated in the SENSEVAL-3 competition on the Romanian data. Another memory based learning approach presented in Kubler and Zhekova [2009] expresses a hypothesis on quantity versus quality of data being used in semi supervised approaches. Adding only good quality data – data classified with high confidence measure, gives better results than adding all the newly annotated data.

Xuri Tang et. al. (2010) proposed an approach based on Word-Class based selection references. It exploits the syntagmatic and paradigmatic semantic redundancy in the semantic system and uses association computation and minimum description length for the task of WSD. The results on Predicate Object collocations are 8% higher than semantic-association based baseline in Chinese.

More recent works like Merhbene, Zouagzi and Zrigui [2013] propose a weighted directed graph based Arabic word sense disambiguation which uses Arabic word net to create sense clusters which are then converted to semantic tree for each word sense. Each sentence in test set is converted to a weighted directed graph and a similarity measure is calculated across all the semantic trees for various senses. Taghipour and Ng [2015] tackle the problem by using a continuous-space representation of words (word embedding) rather than considering words as discrete entities. The model is evaluated on two general-domain lexical sample tasks, an all-words task, and also a domain specific dataset and shows that word embedding consistently improve the accuracy of a supervised word sense disambiguation system, across different datasets.

III. METHODOLGY

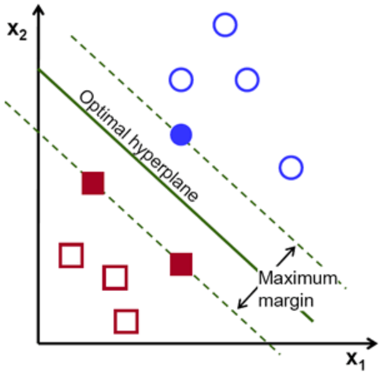
1. Memory Based Learning Approach

A memory based classifier usually avoids generalization as it stores all the examples and thus not leave out any exceptions and as we have also seen in paper [10] that Memory Based Learning approach is typically suited for WSD. We use Tilburg Memory-Based Learner (TiMBL) implementation for memory-based learning. It implements various k-nearest neighbor (k-NN) algorithms. A k-NN classifier simply uses a distance metric to find k most similar instances from the training data for each instance of a test data. It then uses majority voting to classify the training data. We have used various different metrics and different weighing techniques provided by TiMBL. The feature vector we used for in this approach is w_{-3}, w_{-2}, w_{-1}, w_{+1}, w_{+2}, w_{+3}. We have trained and tested the classifier on SENSEVAL-3 Romanian data set. We have also used another feature vector which comprised of part of speech tags along with our previous features.

2. Support Vector Machines

Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition SVM's can also perform non-linear classification by using a technique called the kernel method. It basically projects the input points into a high dimensional space and then tries to linearly separate them. In our implementation we use SVM multiclass [11] which is an open source project.



The feature vector used is w_{-3}, w_{-2}, w_{-1}, w_{+1}, w_{+2}, w_{+3}. Since SVM takes numerical values as input we create six dictionaries for each of the feature vector. Then we browse through the training data to fill dictionary for each feature. For every instance, the word occurring in a corresponding dictionary is marked as 1 and the rest of the words are marked as 0. The feature representation now becomes n-dimensional vector which is the concatenation of all the dictionaries with values 0 and 1.

When including the part of speech tags for experimentation the feature representation changes to the form $w_{-3/POS}$, $w_{-2/POS}$, $w_{-1/POS}$, $w_{+1/POS}$, $w_{+2/POS}$, $w_{+3/POS}$. Rest of the procedure remains the same.

SVM's are good with large vectors and their performance gets better as sparsity in data increases. This happens because the points in feature space can be easily separated if the space is sparse. This is the reason why we selected SVM as one of our machine learners.

3. Lesk Algorithm

The lesk algorithm [12] is one of the earliest algorithm proposed for solving the problem of word sense disambiguation. It is based on the assumption that the local context of a word changes with its sense that is, for each sense of a word there are some specific words in its neighborhood that would be present in most of its occurrences.

We performed experiments for features w_{-3}, w_{-2}, w_{-1}, w_{+1}, w_{+2}, w_{+3} and w_{-2}, w_{-1}, w_{+1}, w_{+2} and the later gave us better performance. After inclusion of part of speech tags the feature space becomes $w_{-2/POS}$, $w_{-1/POS}$, $w_{+1/POS}$, $w_{+2/POS}$.

The algorithm works to create a dictionary of words and their respective counts for each sense. Then for each test instance we calculate the normalized score of intersection with each sense dictionary. The sense with highest score gets assigned to the instance.

The shear simplicity of the algorithm is a compelling reason to choose it. The results as seen in the following section are quite surprising as the results are at par with SVM and MBL.

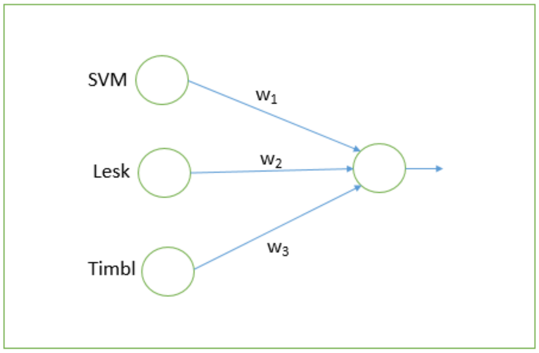
4. The Ensemble

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a weighted vote of their predictions. Bayesian networks were the original ensembles and usually they are used with a bunch of weak classifiers to create a single strong classifiers.

In this paper we use neural networks and majority classifier as the two ensemble approaches.

Majority classifiers are very basic probabilistic models which assign same weights to all input and classify sense according to majority. In our case, the sense that is reported by atleast 2 out of 3 classifiers is produced as the final output. If for some case each classifier outputs different sense and there is no clear majority, the sense with maximum probability gets assigned.

Neural networks are machine learning models which take inspiration from human brain. The model is a set of interconnected neurons with each connection being assigned a numeric weight. We have used a 2 layered neural network with 3 neurons in the input layer for each classifier and a single neuron in the output layer. The choice for a neural network as an ensemble is not very usual but intuitively it assigns weights to each classifier using back propagation algorithm on every input. So in a way it is assigning probabilistic values to each classifier by using a machine learning framework.



5. Semi Supervised Learning

Generating enough annotated data has always been a problem for supervised approaches in Natural Language Processing, hence use of semi supervised learning by classifying unannotated data and adding those rows to the training set of the classifier seems to be a good solution to the problem.

Previous works on semi supervised learning especially Dinu & Kubler [1] have shown that just adding all the classified data can indeed hamper the performance of the classifier and adding instances based on a confidence measure can in turn increase the performance.

Keeping that in mind we define our semi supervised approach to be as follows: Add an instance to the training set if all three classifiers give out the same sense as output. Having such a clause only adds fewer rows than having a majority rule but the probability of this instance being correct becomes 0.98 (By applying Bayes law to data in Table 1). This reduces the chance of noise being added to the training set.

IV. Results and Discussion

Table 1 and 2 show the results for various individual classifiers. Table 1 contains the results for classification results without using the Part of speech tags. Table 2 on the other hand shows the results with the use of part of speech tags.

A. Result Tables

TABLE I.

Word	Classifier without POS		
	TIMBL	***SVM***	***LESK***
Accent.n	82.02%	86.21%	78.16%
Citi.v	82.96%	80.00%	81.53%
Delfin.n	100%	93.33%	100%
Oficial.a	62.88%	60.42%	65.62%
Val.n	82.22%	85.12%	85.12%

Fig. 1. Results of classifiers without POS

TABLE II.

Word	Classifier with POS		
	TIMBL	***SVM***	***LESK***
Accent.n	87.64%	88.51%	77.01%
Citi.v	83.45%	86.92%	82.30%
Delfin.n	100%	100%	100%
Oficial.a	56.70%	60.42%	68.75%
Val.n	81.95%	85.12%	86.77%

Fig. 2. Results of classifiers with POS

It is interseting to see that no single classifier is good for all the words which is another testemant to the no free lunch theorem. Another intersting thing to note is that POS tags improve the performance in most cases. Delfin seems to be the easiest word with 100% accuracy and oficial the hardest. Nouns and verbs fare better than adjectives hinting towards the fact that adjectives are harder to disambiguate. In the experimental setup for ensembles, majority classifier did not need any training but for neural network we had to perform some training. To do that we ran the ensemble classifiers on the training data and trained the neural network on these outputs. The results for both techniques are mentioned in tables below. Again figure 3 has results for classifier without POS tags and figure 4 has results with POS tags. We use the scorer provided under the Senseval-3 competition to generate the fine grained, coarse grained and mixed results.

TABLE III

Word	Majority			Neural		
	Fine	***Coarse***	***Mixed***	***Fine***	***Coarse***	***Mixed***
Accent.n	85.83%	88.54%	87.2%	78.86%	84.21%	81.36%
Citi.v	83.21%	84.88%	84.03%	82.2%	84.78%	83.14%
Delfin.n	100%	100%	100%	100%	100%	100%
Oficial.a	60.29%	64.88%	63.1%	55.34%	59.75%	58.29%
Val.n	86.88%	86.88%	86.88%	81.3%	81.3%	81.3%

Fig. 3. Results of ensembles without POS

TABLE IV

Word	Majority			Neural		
	Fine	***Coarse***	***Mixed***	***Fine***	***Coarse***	***Mixed***
Accent.n	87.62%	90.8%	88.51%	81.56%	86.14%	84.37%
Citi.v	87.81%	89.95%	88.75%	82.84%	85.13%	83.62%
Delfin.n	100%	100%	100%	100%	100%	100%
Oficial.a	63.97%	66.25%	65.13%	57.61%	61.45%	60.23%
Val.n	87.22%	87.22%	87.22%	82.73%	82.73%	82.73%

Fig. 4. Results of ensembles with POS

It is interesting to see contradicting results for both our classifiers. The average performance went up for majority classifiers but for neural networks the perfomance got consistently lower. This hints that the training neural network for an ensemble needs a bit more tact as our approach ends up overfitting the model. Something as simple as the majority classifier shows consistent increase in performance which shows that we can indeed improve the performance of multiple strong classifiers using an ensemble approach.

For semi supervised approach we included the instances to the train set which were voted the same sense by all 3 classifiers. This criteria reduced the number of training example we added but almost nullified the probability of noise being added to the training set. All 3 classifiers were again trained on the new training set and their results were given as input to the ensembles to generate data for figure 5 and figure 6. Again figure 5 has values for classification without POS tags and figure 6 with them.

TABLE V

Word	Majority			Neural		
	Fine	***Coarse***	***Mixed***	***Fine***	***Coarse***	***Mixed***
Accent.n	85.83%	88.72%	87.34%	78.92%	84.96%	82.43%
Citi.v	83.54%	85.37%	84.56%	82.33%	85.14%	83.67%
Delfin.n	100%	100%	100%	100%	100%	100%
Oficial.a	60.29%	64.88%	63.1%	55.34%	59.75%	58.29%
Val.n	87.02%	87.02%	87.02%	81.3%	81.3%	81.3%

Fig. 5. Results of ensembles without POS (semi supervised)

TABLE VI

Word	Majority			Neural		
	Fine	***Coarse***	***Mixed***	***Fine***	***Coarse***	***Mixed***
Accent.n	87.62%	90.92%	88.53%	81.56%	86.14%	84.37%
Citi.v	87.96%	90.14%	89.72%	82.75%	85.54%	83.68%
Delfin.n	100%	100%	100%	100%	100%	100%
Oficial.a	63.97%	66.25%	65.13%	57.61%	61.45%	60.23%
Val.n	87.62%	90.92%	88.53%	81.56%	86.14%	84.37%

Fig. 6. Results of ensembles with POS (semi supervised)

The tables above show a consistent increase in the performance. The increase is not substantial which might happen because of the low increse in training data given our hard conditions.

B. Conclusion

Through our experiments we have shown 3 things. First, inclusion of part of speech tags improves the average performance of a classifier. For no single classfier did we get a reduced performance for all the words. In fact for the ensemble we get better results for almost all words when using POS tags. Second, ensembles end up improving the performance of multiple strong classifiers. Ensembles have been used to combine weak classifiers to make a strong one like in adaboost algorithm. Our experiments show that these can be used to improve performances of strong classifiers as well. Third, increasing the size of the dataset by adding newly classified data with respect to a confidence measure increases the performance of our classifier.

Some future work can be done to improve the performance of our framework. The feature vector we selected is very rudimentary and as shown in [10] using a simple but more informative feature vector can improve the performance a lot. Also use of more classifiers can be done as ensembles approaches generally have lot more than just 3 classifiers. Using a conditional probability based classifier seems another option as it is more complex than a majority classifier but simple enough to not extensive training like in the case of neural networks. Our hard constraint on adding newly classified data to ends up rejecting a lot of data. The constraint can be relaxed by using a majority rule or using probabilities.

References

[1] Georgiana Dinu and Sandra Kubler, 2007. Sometimes less is more: Romanian word sense disambiguation revisited. In proceedings of the International Conference on Recent Advances in Natural Language processing, RANLP

[2] Milhacea, R., Moldovan, D. 1999. An automatic method for generating sense tagged corpora. Proceedings of the 16th National conference on Artificial Intelligence and Eleventh Conference on Innovative application of artificial intelligence, Orlando, Florida, USA, pg. 461-466.

[3] Zheng-Yu Niu, Dong-Hong Ji, and Chew Lim Tan. 2007. Learning model order from labeled and unlabeled data for partially supervised classification, with application to word sense disambiguation. Comput. Speech Lang. 21, 4 (October 2007), 609-619

[4] Wu, Dekai and Su, Weifeng and carpuat, Marine .A kernel PCA method for superiorword sense disambiguation, Proceedings of the 42nd Meeting of the association for Computational Linguistics (ACL'04), Main Volume, 2004, July, Barcelona, Spain, 637-644

[5] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL-44). Association for Computational Linguistics, Stroudsburg, PA, USA, 129-136. DOI=http://dx.doi.org/10.3115/1220175.1220192

[6] Anh-Cuong Le, Akira Shimazu, Van-Nam Huynh, and Le-Minh Nguyen. 2008. Semi-supervised learning integrated with classifier combination for word sense disambiguation. Comput. Speech Lang. 22, 4 (October 2008), 330-345. DOI=http://dx.doi.org/10.1016/j.csl.2007.11.001

[7] Xuri Tang, Xiaohe Chen, Weiguang Qu, and Shiwen Yu. 2010. Semi-supervised WSD in selectional preferences with semantic redundancy. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 1238-1246.

[8] Taghipour, Kaveh and Ng, Hwee Tou. Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, May--June,2015, Denver, Colorado, Association for Computational Linguistics

[9] A. Zouaghi and L. Merhbene and M. Zrigui. Word Sense disambiguation for Arabic language using the variants of the Lesk algorithm.

[10] Sandra Kubler and Desislava Zhekova Semi-Supervised Learning for Word Sense Disambiguation: Quality vs. Quantity.2009

[11] https://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html

[12] Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation, pages 24-26, New York, NY, USA. ACM.