UBC-ZAS: A k-NN based Multiclassifier System to perform WSD in a Reduced Dimensional Vector Space

Ana Zelaia, Olatz Arregi and Basilio Sierra
Computer Science Faculty
University of the Basque Country
ana.zelaia@ehu.es

Abstract

In this article a multiclassifier approach for word sense disambiguation (WSD) problems is presented, where a set of k-NN classifiers is used to predict the category (sense) of each word. In order to combine the predictions generated by the multiclassifier, Bayesian voting is applied. Through all the classification process, a reduced dimensional vector representation obtained by Singular Value Decomposition (SVD) is used. Each word is considered an independent classification problem, and so different parameter setting, selected after a tuning phase, is applied to each word. The approach has been applied to the lexical sample WSD subtask of SemEval 2007 (task 17).

1 Introduction

Word Sense Disambiguation (WSD) is an important component in many information organization and management tasks. Both, word representation and classification method are crucial steps in the word sense disambiguation process. In this article both issues are considered. On the one hand, Latent Semantic Indexing (LSI) (Deerwester et al., 1990), which is a variant of the vector space model (VSM) (Salton and McGill, 1983), is used in order to obtain the vector representation of the corresponding word. This technique compresses vectors representing word related contexts into vectors of a lower-dimensional space. LSI, which is based on Singular Value Decomposition (SVD) (Berry and Browne, 1999) of matrices, has shown to have the ability to extract the relations among features representing words by means of their context of use, and has been successfully applied to Information Retrieval tasks.

On the other hand, a multiclassifier (Ho et al., 1994) which uses different training databases is constructed. These databases are obtained from the original training set by random subsampling. The implementation of this approach is made by a model inspired in bagging (Breiman, 1996), and the k-NN classification algorithm (Dasarathy, 1991) is used to make the sense predictions for testing words.

Our group (UBC-ZAS) has participated in the lexical sample subtask of SemEval-2007 for task 17, which consists on 100 different words for which a training and testing database have been provided.

The aim of this article is to give a brief description of our approach to deal with the WSD task and to show the results obtained. In Section 2, our approach is presented. In Section 3, the experimental setup is introduced. The experimental results are presented and discussed in Section 4, and finally, Section 5 contains some conclusions and comments on future work.

2 Proposed Approach

In this article a multiclassifier based WSD system which classifies word senses represented in a reduced dimensional vector space is proposed.

In Figure 1 an illustration of the experiment performed for each one of the 100 words can be seen. First, vectors in the VSM are projected to the reduced space by using SVD. Next, random subsampling is applied to the training database TD to obtain

different training databases TD_i. Afterwards the k-NN classifier is applied for each TD_i to make sense label predictions. Finally, Bayesian voting scheme is used to combine predictions, and c_j will be the final sense label prediction for testing word q.

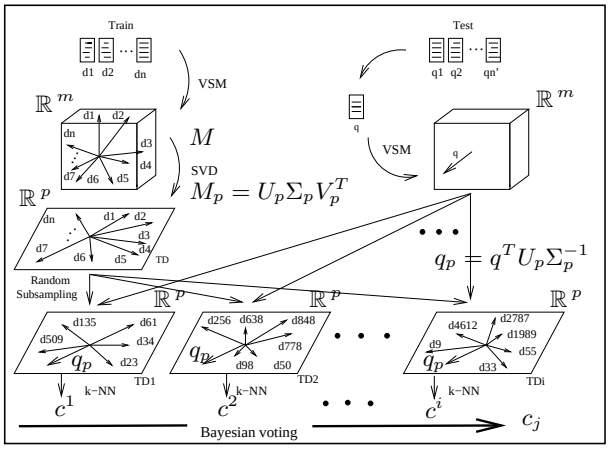

Figure 1: Proposed approach for WSD task

In the rest of this section, the preprocessing applied, the SVD dimensionality reduction technique, the k-NN algorithm and the combination of classifiers used are briefly reviewed.

2.1 Preprocessing

In order to obtain the vector representation for each of the word contexts (documents, cases) given by the organizers of the SemEval-2007 task, we used the features extracted by the UBC-ALM participating group (Agirre and Lopez de Lacalle, 2007). These features are local collocations (bigrams and trigrams formed with the words around the target), syntactic dependencies (object, subject, noun-modifier, preposition, and sibling) and Bag-of-words features (basically lemmas of the content words in the whole context, and in a $\pm$4-word window).

2.2 The SVD Dimensionality Reduction Technique

The classical Vector Space Model (VSM) has been successfully employed to represent documents in text categorization tasks. The newer method of Latent Semantic Indexing (LSI) [1] (Deerwester et al., 1990) is a variant of the VSM in which documents are represented in a lower dimensional space created from the input training dataset. The SVD technique used by LSI consists in factoring term-document matrix M into the product of three matrices, $M = U\Sigma V^T$ where Σ is a diagonal matrix of singular values, and U and V are orthogonal matrices of singular vectors (term and document vectors, respectively).

For classification purposes (Dumais, 2004), the training and testing documents are projected to the reduced dimensional space, $q_p = q^T U_p \Sigma_p^{-1}$, by using p singular values and the cosine is usually calculated to measure the similarity between training and testing document vectors.

2.3 The k-NN classification algorithm

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing case, the k-NN classifier ranks its nearest neighbors among the training word senses, and uses the sense of the k top-ranking neighbors to predict the corresponding to the word which is being analyzed (Dasarathy, 1991).

2.4 Combination of classifiers

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). A widely used technique to implement this approach is *bagging* (Breiman, 1996), where a set of training databases TD_i is generated by selecting n training cases drawn randomly with replacement from the original training database TD of n cases. When a set of $n_1 < n$ training cases is chosen from the original training collection, the bagging is said to be applied by random subsampling. In fact, this is the approach used in our work and the n_1 parameter has been selected via tuning.

According to the random subsampling, given a testing case q, the classifier will make a label prediction c^i based on each one of the training databases TD_i. One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value $\text{cv}_{c_j}^i$ is calculated for each training database TD_i and sense c_j to be predicted. These confidence values have been calculated based on the training collection. Confidence values are summed

[1] http://lsi.research.telcordia.com, http://www.cs.utk.edu/∼lsi

by sense; the sense c_j that gets the highest value is finally proposed as a prediction for the testing examples.

3 Experimental Setup

In the approach proposed in this article there are some decisions that need to be taken, because it is not clear (1) how many examples should be selected from the TD of each word in order to create each one of the TD_i; (2) which is the appropriate dimension to be used in order to represent word related contexts (cases) for each word database; (3) which is the appropriate number of TD_i that should be created (number of classifiers to be used) and (4) which is the appropriate number of neighbors to be considered by the k-NN algorithm.

Therefore, a parameter tuning phase was carried out in order to fix the parameters. We decided to adjust them for each word independently.

In the following, the parameters are introduced and the tuning process carried out is explained. For two of the parameters (the number of classifiers and the number of neigbors for k-NN), the tuning phase was performed based on our previous experiments on document categorization tasks.

3.1 The size of each TD_i

As it was mentioned, the multiclassifier is implemented by random subsampling, where a set of $n_1 < n$ vectors is chosen from the original training collection of n examples for a given word (n is a different value for each one of the 100 words). Consequently, the size of each TD_i will vary depending on the value of n_1. The selection of different numbers of cases was experimented for each word in two different ways:

a) according to the following equation:

$$n_1 = \sum_{i=1}^{s}(2 + \lfloor \frac{t_i}{j} \rfloor), \qquad j = 1, \ldots, 10$$

where t_i is the total number of training cases in the sense c_i and s is the total number of senses for the given word. By dividing parameter t_i by j, the number of cases selected from each sense preserves the proportion of cases per sense in the original one. However, it has to be taken into account that some of the senses have a very low number of cases assigned to them. By summing 2, at least 2 cases will be selected from each sense. In order to decide the optimal value for parameter j, the classification experiment was carried out varying j from 1 to 10 for each word.

b) selecting a fixed number of cases for each of the senses which appeared for the word in the training database. Again, in the tuning phase, different numbers of cases (from 1 to 10) have been used for each of the 100 words in order to select a value for each of the words.

We optimized the size of each TD_i for each word by selecting the number of cases sometimes by procedure a) and sometimes by b).

3.2 The dimension of the reduced Vector Space Model

Taking into account the wide differences among the training case numbers for different words, we decided to project vectors representing them to different reduced dimensional spaces. The selection of those dimensions is based on the number of training cases available for each word, and limited to 500; the used dimensions vary from 19 (for the word *grant*) to 481 (for the word *part*).

3.3 The number of classifiers (TD_i)

Based on previous experiments carried out for document categorization (Zelaia et al., 2006), we decided to create 30 classifiers for some words and 50 for others, i.e. 30 or 50 individual k-NN algorithms will be used by the multiclassifier in order to combine opinions by Bayesian voting.

3.4 Number of neigbors for k-NN

Based on our previous experiments, we decided to use $k = 1$ and $k = 5$, and to select the best for each of the words. The cosine similarity measure is used in order to find the nearest or the 5 nearests.

4 Experimental Results

The experiment was conducted by considering the optimal values for parameters tuned by using the training case set.

Results published in this section were calculated by the SemEval-2007 organizers. Table 1 shows accuracy rates obtained by the 13 participants in the SemEval-2007, 17 task, lexical sample WSD subtask.

System	Accuracy	System	Accuracy
1.	0.887	8.	0.803
2.	0.869	**9.**	**0.799**
3.	0.864	10.	0.796
4.	0.857	11.	0.743
5.	0.851	12.	0.538
6.	0.851	13.	0.521
7.	0.838		

Table 1: Accuracy rates obtained by the 13 participants. SemEval-2007, 17 task (Lexical Sample)

The result obtained by our system is 0.799 (the 9th among 13 participants), 1 point over the mean accuracy (0.786).

5 Conclusions and Future Work

Results obtained show that the construction of a multiclassifier, together with the use of Bayesian voting to combine label predictions, plays an important role in the improvement of results. We also want to remark that we used the SVD dimensionality reduction technique in order to reduce the vector representation of cases.

The approach presented in this paper was already used in a document categorization task. However, we never used it for WSD task. Therefore, in order to adapt the method to the new task, we fixed some parameters based on our previous experiments (30-50 classifiers, $k = 1, 5$ for the k-NN algorithm) and tuned some other parameters by experimenting quite a high number of TD_i sizes and using different dimensions for each word. However, we noticed that the application of our approach to a different task is not straightforward. Greater effort will have to be made in order to tune the different parameters to this specific task of WSD.

One of the main difficulties we found was the difference in the number of training cases, comparing with the high number usually available in other tasks like text categorization.

As future work, we can think of applying a new preprocessing approach in order to extract better features from the training database which could help the SVD technique improving the accuracy after a dimensionality reduction is applied. The use of Wordnet may help.

6 Acknowledgements

This research was supported by the University of the Basque Country by the project "ANHITZ 2006: Language Technologies for Multilingual Interaction in Intelligent Environments", IE 06-185

We wish to thank to the UBC-ALM group for helping us extracting learning features.

References

E. Agirre and O. Lopez de Lacalle. 2007. Ubc-alm: Combining k-nn with svd for wsd. submited for publication to SemEval-2007.

M.W. Berry and M. Browne. 1999. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM Society for Industrial and Applied Mathematics, ISBN: 0-89871-437-0, Philadelphia.

L. Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

B.V. Dasarathy. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*. IEEE Computer Society Press.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

T.G. Dietterich. 1998. Machine learning research: Four current directions. *The AI Magazine*, 18(4):97–136.

S. Dumais. 2004. Latent semantic analysis. In *ARIST (Annual Review of Information Science Technology)*, volume 38, pages 189–230.

T.K. Ho, J.J. Hull, and S.N. Srihari. 1994. Decision combination in multiple classifi er systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75.

G. Salton and M. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.

A. Zelaia, I. Alegria, O. Arregi, and B. Sierra. 2006. A multiclassifi er based document categorization system: profi ting from the singular value decomposition dimensionality reduction technique. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications*, pages 25–32.