

Integrating and Evaluating WSD in the Adaptation of a Lexical Database in Text Categorization Task

L. Alfonso Ureña López¹, M. De Buenaga², M. García¹ and J.M. Gómez²

¹ Dpto. de Informática, Universidad de Jaén, Avda. Madrid 35.
23071 Jaén, Spain
{laurena, mgarcia}@ujaen.es

² Dpto. de Inteligencia Artificial, Universidad Europea de Madrid. Madrid. Spain
{buenaga, jmgomez}@dinar.uem.es

Abstract. Improvement in the accuracy of identifying the correct word sense (WSD) will give better results for many natural language processing tasks. In this paper, we present a new approach using WSD as an aid for Text Categorization (TC). This approach integrates a set of linguistics resources as knowledge sources. So, our approach, for TC using the Vector Space Model, integrates two different resources in text content analysis tasks: a lexical database (WORDNET) and training collections (Reuters-21578). We present the WSD task to TC application. Specifically, we apply WSD to the process of resolving ambiguity in categories WORDNET, so we complement training phases. We have developed experiments to evaluate the improvements obtained by the integration of the resources in TC task and for application of WSD in this task, obtaining a high accuracy in disambiguating category senses of WORDNET.

1 Introduction

The task of Word Sense Disambiguation (WSD) is to identify the correct sense of a word in a particular context. Improvement in the accuracy of identifying the correct word sense will result in better for many natural language processing (NLP) tasks[5](i. e. in Text Categorization (TC)[2], machine translation[3], accent restoration[18], Information Retrieval (IR)[5, 14],etc). However, WSD task has not been applied to these NLP applications.

We present a new approach using WSD as an aid for TC specific task. This approach integrates a set of lexical resources as knowledge sources. Because we make the basic assumption that the more informed a system is, the better it performs, so, our approach for both TC[2] and WSD[16, 4] uses the Vector Space Model (VSM)[12] as a uniform way to integrate two different resources in text content analysis tasks: a lexical database³ (WORDNET[10]) and training

³ A lexical database is a reference system that accumulates information on the lexical items of one or several languages. In this view, machine-readable dictionaries can also be regarded as primitive lexical databases. Current lexical databases include WORDNET, EuroWordNet, EDR and Roget's Thesaurus. WordNet's large coverage and frequent utilization has led us to use it for our experiments.

collections (Reuters-21578 y SemCor). The first task, TC[7, 2] is the classification of documents according to a set of one or more pre-existing categories. TC is a difficult and useful operation frequently applied to the assignment of subject categories to documents, to route and filter texts, or as a part of natural language processing systems. The other task, WSD[19, 1, 15] is marking each word token in a text with some marker identifying its semantic category, where the semantic category is the sense of the word token from some lexicon or dictionary.

Among many training approaches that have been employed, we have selected the Rocchio and the Widrow-Hoff algorithms. We have combined the utilization of each algorithm with WORDNET, using the Vector Space Model for this task. This combination shows that an integrated approach combining a training collection and a lexical database performs better than the isolated use of a training collection. We present the WSD[16] task to TC[2] application. Specifically, we apply WSD to the process of disambiguate categories in WORDNET, so we complement the training phase.

2 Learning algorithms

The basic idea in training based approaches to text classification tasks is that a set of manually classified items can be used to predict the assignment of new items to the classes. Training algorithms provide a way to calculate the weight vectors for the classes. Basically, the training process assigns a weight to a term in a class vector, in proportion to the number of occurrences of the term in items manually assigned to the class, and in proportion to the importance of the term in the collection too.

We have selected the Rocchio[11] and the Widrow-Hoff[17] algorithms to compute the term weights for classes in our approach. The first one is an algorithm traditionally used for Relevance Feedback in IR. The second one comes from Machine Learning. Both algorithms give the chance of integrating an initial representation computed by the utilization of an external resource like WORDNET[2].

2.1 The Rocchio Algorithm

The Rocchio algorithm produces a new weight vector wc_k from an existing one wc_k^0 and a collection of training items. The component i of the vector wc_k is computed by the formula:

$$wc_{ik} = \alpha wc_{ik}^0 + \beta \frac{\sum_{l \in C_k} wd_{il}}{n_k} + \gamma \frac{\sum_{l \notin C_k} wd_{il}}{P - n_k} \quad (1)$$

Where wc_{ik}^0 is the initial weight of the term i for the category k . wd_{il} is the weight of the term i for the training document l . C_k is the set of indexes of documents assigned to the class k and n_k the number of these documents. The parameters α , β and γ control the relative impact of the initial, positive and negative weights respectively in the new vector. As Lewis[9], we have used the values $\alpha = 16$ and $\gamma = 4$. The value of α is set to 20, in order to balance the importance of initial and training weights

2.2 The Widrow-Hoff Algorithm

The Widrow-Hoff algorithm starts with an existing weight vector wc_k^0 and sequentially updates it once for every training document. The component i of the vector $wc^l + 1_k$ is obtained from the l th document and from the l th vector by the formula:

$$wc_{ik}^{l+1} = wc_{ik}^l + 2\eta(wd_l \cdot wc_k^l - y_l)wd_{il} \quad (2)$$

Where wc_{ik}^l is the weight of the term i in the l th vector for category k . wd_l is the term weight vector for document l . wc_k^l is the l th vector for category k . y_l is 1 if the l th document is assigned to the category k and 0 in other cases, and wd_{il} is the weight of term i in the l th document. The constant η is the learning rate, which controls how quickly the weight vector is allowed to change, and how much influence each new document has on it. A value typically used for η is $\frac{1}{4X^2}$, being X the maximum value of the norm of vectors that represent training documents.

3 Integrating WORDNET in Text Categorization

The combination of information from WORDNET and from the training collection is performed by the use of initial weights for categories. The utilization of WORDNET is based in the assumption that the name of a category can be a good predictor of its occurrence. For instance, the occurrence of the word “barley” suggests that a news article should be classified into the BARLEY category. The prediction of more general categories like EARN (*earnings*) should instead rely on the occurrence of semantically more independent terms like “dollar” or “invest”. Lexical Databases contain many kinds of information on lexical items: concepts; synonymy and other lexical relations; hyponymy and the other conceptual relations; etc. For instance, WORDNET represents concepts as synonyms sets, or *synsets*. Using WORDNET, synonyms for names of categories can be found, and then used to predict categories assignments. In our approach, we focussed on the relation in WORDNET. We have performed a “category expansion”, similar to query expansion in IR. For any category, its closer *synsets* are selected, and any term belonging to them is added to the term set. We have taken only concepts that are candidates to represent the meaning of each category. The selection of candidate *synsets* can be considered as a disambiguation process. This disambiguation process has been automated, as is shown in section four.

Evaluation in text classification operations exhibits great heterogeneity. Several metrics and test collections have been used for different approaches or works. The VSM promotes evaluation based in *recall* and *precision*[8, 7]. We have computed precision at 11 recall levels, taking the average precision as the number which describes the overall performance of each technique. Precision averages are produced at each recall level for all the categories. So, each category has the

same influence in final results, whether very frequent or not. These results are shown in Table 1.

We have obtained better results through the integrated approach rather than the approach based only on training. With the integration of WORDNET, average precision achieves an improvement of 20 points for both algorithms.

Table 1. Overall results from our experiments in TC.

	Training		Training+WordNet	
	Rocchio	Widrow-Hoff	Rocchio	Widrow-Hoff
0.0	0.567	0.565	0.733	0.703
0.1	0.478	0.484	0.703	0.659
0.2	0.423	0.427	0.661	0.610
0.3	0.362	0.375	0.601	0.555
0.4	0.315	0.331	0.573	0.530
0.5	0.270	0.279	0.556	0.511
0.6	0.224	0.225	0.503	0.469
0.7	0.175	0.179	0.416	0.436
0.8	0.147	0.149	0.359	0.412
0.9	0.119	0.122	0.296	0.351
1.0	0.109	0.111	0.201	0.289
Avg.	0.290	0.295	0.509	0.502

4 Disambiguating Categories in WORDNET

As we have seen in the previous section, when WORDNET (*synsets*) information is integrated with training information, a problem related to the senses ambiguity in the different *synsets* of a category arises.

This problem was solved by using a new WSD algorithm, based on lexical resources integration as in TC task. This WSD approach has shown promising results confirmed by our previous work [16, 4] and present results.

We have combined WORDNET information with the Rocchio and Widrow-Hoff algorithms to produce senses representation. We constructed a weight vector for the category to be disambiguated, basically using the frequency of the terms that appear in a contextual window around the word, that is, using its context [15]. We have used the paragraph as contextual window, where the weight vector for word i with sense j is $s_{ji} = \langle ws_{j1}, ws_{k1}, \dots, ws_{kn} \rangle$. Where ws_{ki} is the weight of the surrounding word. We also constructed weight vectors for each associated *synset* to disambiguate. The weight vector for a synset c_k , is $c_k = \langle wc_1, wc_{k1}, \dots, wc_{kn} \rangle$. The similarity between the word (context) and each sense is obtained with the formula:

$$sim(s_{ji}, c_i) = \frac{\sum_{i=1}^N ws_{ji} \cdot wc_i}{\sum_{i=1}^N ws_{ji}^2 \cdot \sum_{i=1}^N wc_i^2} \quad (3)$$

Secondly, weights for terms vectors can be computed making use of the well-known formula[13] based on term frequencies. We have used the expression:

$$ws_{ji} = t_{ji} \cdot \log_2(n/f_i) \quad (4)$$

Where t_{ji} is the frequency of term j with sense i in *contextual windows*. n number of senses of term i and f_i the number of *contextual windows* where the term i occurs. The expression (4) is the weight used for any document in our approach, and thus in formulae (1) and (2).

Evaluation in WSD exhibits great heterogeneity. We have used *macroaveraging* and *microaveraging* as metrics to evaluate our system[8]. The results are shown in Table 2. We have obtained a high precision in the lexical ambiguity resolution applied to a specific task as is TC.

Table 2. WSD results applied to TC task.

Precision	Training+ WordNet	
	Rocchio	Widrow-Hoff
<i>macroaveraging</i>	92.2%	92.0%
<i>microaveraging</i>	90.5%	90.6%

5 Conclusion and Future Work

In this paper, we have presented a new approach using WSD as an aid for TC. This approach integrates a set of linguistics resources as knowledge sources. Our approach for TC uses the Vector Space Model to integrate two different lexical resources: a lexical database (WORDNET) and training collections (Reuters-21578 and SemCor).

TC is the classification of documents according to a set of one or more pre-existing categories. TC is a difficult and useful operation frequently applied to the assignment of subject categories to documents. We present the WSD task to TC application. Specifically, we apply WSD to the process of disambiguate categories in WORDNET, so we complement training phase. We have developed experiments to evaluate the improvements obtained by the integration of the resources in TC task and for application of WSD in this task. Using both Reuters and WORDNET, we have obtained better results for the integrated approach than for the approach based only on training. With the integration of WORDNET, average precision achieves an improvement of 20 points for both algorithms.

We have applied our automatic WSD method (also based on training collection and lexical database) to the *synsets* associates to WORDNET in TC process training. We have obtained more than 92% precision in disambiguating the sense of the categories WORDNET. This way, we have realized automatically the total process of TC, applying WSD.

Currently, we are realizing new experiments to apply to any other specific tasks related to NLP, in which WSD can be very useful.

References

1. Agirre E., Rigau G.: Word sense disambiguation using conceptual density. In Proceedings of COLING 1996.
2. Buenaga Rodríguez M., Gómez Hidalgo J.M., Díaz Agudo B.: Using WordNet to Complement Training Information in Text Categorization. Second International Conference on Recent Advances in Natural Language Processing, 1997.
3. Brown P. B., Pietra S. A., Pietra V. J.: Word Sense Disambiguation Using Statistical Methods. In Proc. of ACL, pp. 264-270, 1991.
4. Díaz Esteban, A., Buenaga Rodríguez, M., Ureña López, L. A., García Vega, M.: Integrating Linguistic Resources in an Uniform Way for Text Classification Tasks. In Proceedings of the First International Conference on Language Resources and Evaluation. (ELRA) Granada. Spain 1998.
5. Kilgariff A.: What is word sense disambiguation good for?. Proc. Natural Language Processing Pacific Rim Symposium. Phuket, Thailand. December 1997. pp 209-214.
6. Kilgariff A.: Foreground and Background Lexicons and Word Sense Disambiguation for Information Extraction. Proc. International Workshop on Lexically Driven Information Extraction. Frascati, Italy. July 1997. pp 51-62.
7. Larkey, L.S., Croft, W.B.: Combining classifiers in text categorization. In Proceedings of the ACM SIGIR. 1996.
8. Lewis, D.: Representation and learning in information retrieval. Ph.D. Thesis, Department of Computer and Information Science, University of Massachusetts. 1992.
9. Lewis, D.D., Schapire, R.E., Callan, J.P. and Papka, R.: Training algorithms for linear text classifiers. In Proceedings of the ACM SIGIR, 1996.
10. Miller G.: WordNet: lexical database. Communications of the ACM Vol 38, No. 11. 1995.
11. Rocchio, J.J. Jr.: Relevance feedback in information retrieval. In G. Salton, editor, The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, 1971.
12. Salton G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill.1983.
13. Salton, G., Automatic Text Processing: the transformation, analysis and retrieval of information by computer. Addison Wesley. 1989.
14. Sanderson, M.: Word sense disambiguation and information retrieval. Ph.D. Thesis, Department of Computing Science, University of University of Glasgow. 1996.
15. Ureña López, L. A., García Vega, M., Buenaga Rodríguez, M., Gómez Hidalgo, J. M.: Resolución de la ambigüedad léxica mediante información contextual y el modelo del espacio vectorial. Séptima Conferencia de la Asociación Española para la Inteligencia Artificial. CAEPIA. 1997.
16. Ureña López, L. A., García Vega, M., Buenaga Rodríguez, M., Gómez Hidalgo, J. M.: Resolución automática de la ambigüedad léxica fundamentada en el modelo del espacio vectorial usando ventana contextual variable. AESLA. 1998
17. Widrow, B., Sterns., S. : Adaptative Signal Processing. Prentice-Hall, 1985.
18. Yarowsky D.: Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics, (ACL'94). 1994.
19. Yarowsky D.: Unsupervised word sense disambiguation rivalling supervised methods. In Proceedings of the 33th Annual Meeting of the ACL'95. 1995.
20. Yoshiky N., Yoshihiko N.: Co-occurrence vectors from corpora vs. distance vectors from. In Proceedings of COLING94.