

The Role of Word Sense Disambiguation in Automated Text Categorization

José María Gómez Hidalgo¹, Manuel de Buenaga Rodríguez¹, and José Carlos Cortizo Pérez²

¹ Universidad Europea de Madrid
Villaviciosa de Odón, 28670, Madrid (Spain)
jmgomez,buenaga@uem.es
WWW: <http://www.esi.uem.es/~jmgomez/~buenaga>

² AINet Solutions
Fuenlabrada, 28943, Madrid (Spain)
jccp@ainetsolutions.com
WWW: <http://www.ainetsolutions.com/>

Abstract. Automated Text Categorization has reached the levels of accuracy of human experts. Provided that enough training data is available, it is possible to learn accurate automatic classifiers by using Information Retrieval and Machine Learning Techniques. However, performance of this approach is damaged by the problems derived from language variation (specially polysemy and synonymy). We investigate how Word Sense Disambiguation can be used to alleviate these problems, by using two traditional methods for thesaurus usage in Information Retrieval, namely Query Expansion and Concept Indexing. These methods are evaluated on the problem of using the Lexical Database WordNet for text categorization, focusing on the Word Sense Disambiguation step involved. Our experiments demonstrate that rather simple dictionary methods, and baseline statistical approaches, can be used to disambiguate words and improve text representation and learning in both Query Expansion and Concept Indexing approaches.

1 Introduction

Automated Text Categorization (ATC) – the automatic assignment of documents to pre-defined categories – is one of the most important text classification tasks nowadays. Automated text classifiers can be employed to route e-mail messages to e-mail folders [1], to populate Web directories with Web pages [2], or to filter out inappropriate e-mail or Web content, like *spam* [3] or pornography [4].

While it is possible to build text classifiers by hand, the most popular approach is using Information Retrieval and Machine Learning techniques to automate the process. In short, given a set of manually classified documents, they are represented as feature vectors and fed into a learning algorithm, which produces an automatic classifier. There is empirical evidence that ATC systems built this way can achieve human levels of performance, specially when categories are subject or topic oriented, like *arts*, *computers* or *entertainment* [5].

However, like Information Retrieval (IR), ATC faces the problem of language variability. The same way polysemy (one word with two or more senses) make ambiguous queries to retrieve irrelevant results, it also can make an ATC classifier to incorrectly label a new document. Also, failing to recognize synonyms make relevant documents not retrieved for a query; the effect in ATC is similar. These problems have been faced since the very early days in Information Retrieval [6], often through the utilization of lexical-semantic resources like thesauri (e.g. Roget’s Thesaurus) or Lexical Data Bases (LDBs, e.g. WordNet [7]), and making use of Word Sense Disambiguation (WSD).

WSD consists on the identification of the actual sense of a word in a context. It plays a key role in the usage of lexical-semantic resources, because it is used to map actual word occurrences to their suitable semantic classes (or senses). The increased interest in WSD, represented by the SENSEVAL competitions³, has led to important performance improvements, although its accuracy is much below other frequent Natural Language Processing tasks, like POS-Tagging.

Being ATC different to IR in many ways, we focus on integrating lexical-semantic resources in it, and studying the role of WSD. Our general goal is improving ATC effectiveness by addressing language variability. We have extended two traditional models of integration of thesauri in IR to ATC, namely Query Expansion and Concept Indexing, and tested relatively simple WSD methods for it. Our WSD methods are based on the two dominant models nowadays, dictionary and learning based. We use the LDB WordNet and the test collections Reuters-21578 and SemCor in our experiments.

Our hypothesis is that it is possible to improve ATC by using LDBs and simple WSD methods. This is due to the fact that in ATC, there is much information available *a priori* in the form of manually labeled documents, which are not available in IR. The results of our experiments confirm the hypothesis, and encourage future work on integrating other resources in ATC.

2 Word Sense Disambiguation in Text Classification

In this section, we review the usage of WSD in IR and in ATC, and the current state-of-the-art in WSD methods. The work in IR can be extended to ATC in the form of two models that are described in the next sections.

2.1 WSD in Information Retrieval

Since the early days in Information Retrieval (see e.g. [6]), automatic *thesauri* (synonym dictionaries, or lists of semantically related word classes) have been used with the aim of improving retrieval effectiveness. The underlying idea is to overcome language variation problems, specially polysemy and synonymy. If a query submitted to a retrieval engine contains a polysemous word – but intended by the user to be interpreted with only one of its senses (e.g. “bank”

³ See <http://senseval.org/>.

in the financial sense), there are many chances that the engine will return false matches, corresponding to word occurrences with a different sense (e.g. “bank” as a place to sit on). Also, if a query contains a word that has several synonyms (e.g. “astronaut”), only documents in which the query word occur will be retrieved, probably missing other relevant documents which its synonyms occur in (e.g. documents with the word “cosmonaut”). The first situation affects precision, while the second mainly hurts recall.

Given a semantic classification of words (e.g. a thesaurus), in which related words are grouped in classes, it can be used to improve retrieval effectiveness in two basic ways:

- By replacing word occurrences in documents and queries by the semantic classes they refer to. This way, only the correct meaning of a word will be used for retrieval, avoiding false matches due to polysemy, and improving precision. Most likely, recall will also be improved, because synonyms will be in the same semantic classes, and documents in which they occur will be indexed with regard to the same semantic class.
- By replacing word occurrences by their synonyms or semantically related words (words in the same class). This way, all documents containing a word will also contain their synonyms, and they will be retrieved when any of the words in the class are used in the query. This has the effect of improving recall. We must note that is more efficient to replace only words in queries, and indexing the documents by using only the words occurring in them.

We call the former method *concept indexing*, and the latter *query expansion*. In any case, a WSD system is required. In concept indexing, both queries and documents must be disambiguated, in order to identify the right semantic classes used, and taking them as indexing units. In query expansion, ambiguous words (pertaining to two or more semantic classes) must be disambiguated, allowing their replacement by the right set of synonyms.

Both methods have been applied when using WordNet in Information Retrieval since WordNet’s very beginning⁴. The first work was done by Voorhees, dating back to the beginning of the previous decade. In [8] she used a form of concept indexing focused in the WordNet nouns subset, showing a general decrease of performance. Query expansion with WordNet synsets and conceptual relations were after tested by her on a TREC collection, with no better results [9]. These experiences have led her to state that “linguistic techniques must be essentially perfect to help” [10], meaning in this context that Word Sense Disambiguation low effectiveness has a direct impact in the usage of WordNet for Information Retrieval.

However, other works have demonstrated that WordNet can improve text retrieval. In particular, indexing with WordNet synsets has been tested on a (rather artificial) collection in [11], showing that: (1) under perfect disambiguation, concept indexing with WordNet synset greatly improves retrieval effectiveness; and

⁴ See <http://enr.smu.edu/~rada/wnb/> for a comprehensive bibliography. In this paper, we review only those works we consider specially relevant to our research.

(2) up to a 60% disambiguation errors can be tolerated, while improving performance.

Summing up all experiences, there is an intuition but no clear results supporting that Word Sense Disambiguation can improve Information Retrieval. But in this work, we demonstrate that WSD does improve ATC.

3 Learning Based ATC

The most popular model for building ATC systems nowadays, is based on IR and Machine learning techniques, and involves the following steps [5]:

- First, documents in a manually classified collection (the *training collection*) are represented as attribute or feature vectors, in which features are usually stemmed words, after deleting the most frequent ones using a stop list. The value of an attribute in a document vector (its *weight*) can be binary (1 if the stem occur in the document, and 0 otherwise), Term Frequency (TF, the number of times the stem occurs in the document), or TF.IDF (being IDF the Inverse Document Frequency, a function of the times the stem occurs in the whole document collection). This document representation is regarded in the literature as the *bag of words* model, and it corresponds to the standard Vector Space Model in Information Retrieval [12].
- Secondly, attributes are filtered according to an feature quality metric, in order to reduce the vector space dimensionality, allowing learning and generally improving the overall accuracy of the obtained system. Effective quality metrics include Information Gain, χ^2 and the Document Frequency [13].
- Finally, the document vectors are taken as examples by a Machine Learning algorithm, which builds a classification function or *classifier*, based on the previous attributes. The classifier can take the form of a set of rules, a decision tree, a linear discrimination function, etc., depending on the algorithm used. The most effective learners used for this problem include Support Vector Machines, *k*-Nearest Neighbors, and classifier committees like Boosting [14, 5].

Effectiveness of ATC systems built this way is comparable to humans. In words by Sebastiani: “*Automated TC (. . .) has reached effectiveness levels comparable to those of trained professionals.*” [5].

However, ATC faces the same language variability problems as IR. Polysemy makes automatic classifiers wrongly classify new documents. For instance, a document containing “bank” in the sense of a financial institution, may be classified in the *environment* category, if the system understands the word in the bank of a river sense. Also, and regarding synonymy, a classifier trained on documents on which the word “astronaut” occur (in the *space* category), may fail to recognize another document in which only the word “cosmonaut” occurs.

4 WSD in Automated Text Categorization

Automated Text Categorization offers new opportunities for improving effectiveness using Word Sense Disambiguation. The limited information and short life of text retrieval queries contrast with populated, long living categories. Also, information quality metrics used in ATC for dimensionality reduction allow to select appropriate indexing units in text representation. This makes usage of lexical-semantic resources in ATC even more promising than in IR, and WSD less critical for it.

Most works in WSD for ATC (focused in the LDB WordNet), have adopted the concept indexing model from IR. The basic idea of concept indexing with WordNet synsets is recognizing the synsets to which words in texts refer, and using them as terms for representation of documents in a Vector Space Model. Synset weights in documents can be computed using the same formulas for word stem terms in the bag of words representation.

Experiments focused on concept indexing with WordNet synsets for TC have mixed results. On one side, lack of disambiguation has led to loss of effectiveness in some works. On the other, it is not clear that full disambiguation is absolutely required to obtain a document representation more effective than the bag of words model. We discuss three works specially relevant.

- Scott and Matwin [15] have tested a text representation in which WordNet synsets corresponding to the words in documents, and their hypernyms, were used as indexing units with the rule learner Ripper on the Reuters-21578 test collection. The results of the experiments were discouraging, probably due to the fact that no disambiguation at all is performed, and to the inability of Ripper to accurately learn in a highly dimensional space.
- Fukumoto and Suzuki [16] have performed experiments extracting synonyms and hypernyms from WordNet nouns in a more sophisticated fashion. First, synsets are not used as indexing units; instead, words extracted from synsets whose words occur in the documents are used. Second, the height to which the WordNet hierarchy is scanned is dependent on the semantic field (location, person, activity, etc.), and estimated during learning. These experiments were performed with Support Vector Machines on the Reuters-21578 test collection, and their results are positive, with special incidence on rare (low frequency) categories. Notably, no sense disambiguation was performed.
- Petridis *et al.* [17] used WordNet synsets as indexing units with several learning algorithms on the SemCor text collection. In this collection, all words and collocations have been manually disambiguated with respect to WordNet synsets. The lazy learner k-Nearest Neighbors, the probabilistic approach Naive Bayes, and a Neural Network were tested on several text representations. The concept indexing approach performed consistently better than the bag of words model, being the Neural Network the best learner.

The work by Scott and Matwin suggests that some kind of disambiguation is required. The work by Fukumoto and Suzuki allows to suppose that no full

disambiguation is needed. Finally, the work by Petridis *et al.* demonstrates that perfect disambiguation is effective, over a limited number of learning algorithms and an correctly disambiguated text collection. Positive evidence is scarce yet, and our own recent experiments with this model show mixed results [18].

Much less work has been devoted to the query expansion method in ATC. As far as we know, there is only our previous work [19, 20], followed by [21]. In [19], we proposed a model for learning-based ATC, in which the category names are enriched with WordNet synonymy information, under optimal WSD conditions, and focusing a class of linear learning algorithms. We further automated the WSD step using a heavy knowledge-based algorithm in [20]. In these works, we have employed the LDB WordNet, and the standard ATC test collection Reuters-21578. From these works, it can be stated that the query expansion method is effective, and that perfect or complex WSD methods work for the problem. The work has been adapted to other semi-supervised learning algorithms in [21]. Regarding this method, we address here the simplification of the WSD, adopting a simple dictionary-based WSD method, with positive results.

4.1 Current methods in WSD

Following [22], there two basic methods for WSD:

- Dictionary based WSD, in which only the information in a dictionary (or lexical-semantic resource) is used. The most simple method involves labeling each word occurrence in a context, with the sense in which the definition is more similar to the context.
- Supervised WSD, in which the dictionary is taken only as a reference (if any), and the main source of information is a training collection of sense usage samples. A word occurrence is labeled by an automatic classifier, trained on these samples.

Recent SENSEVAL competitions (specially in the All-Words English Task) demonstrate that:

- Supervised methods perform better than dictionary methods, but there is an increasing interest on integrating both methods.
- Top performing systems are only performing slightly better than statistical baseline methods, still far from human performance.

Instead of using highly complex methods, as those tested in SENSEVAL, we make use of two simple WSD methods, a dictionary based one for the Query Expansion integration, and a baseline statistical approach in the Concept Indexing method. Therefore, as our results are positive, we demonstrate that the characteristics of the ATC task make perfect WSD not needed.

5 Query Expansion method

In this section, we describe the Query Expansion method for integrating WordNet information in ATC, and we test it on the standard Reuters-21578. We make use of a simple dictionary method for WSD, showing that full WSD is not needed.

5.1 Description

In ATC, categories are best described by the set of documents they contain. However, and since categories are mostly provided for human use (like in Web Directories, or the subject keywords in libraries), they have attached a name and sometimes, a short description. We interpret these names as queries in IR, adapting the Query Expansion method the following way⁵:

- For each category, its name is extracted, and searched in the LDB WordNet for its senses (synsets). In the case of multi-word names, if the whole name is not found, it is divided in words, and each of them searched individually. For each category, a set of potential synsets is obtained.
- The correct synset is identified for each ambiguous expression or word. This is a WSD process we detail below. From this step, we get a synset per category, or per word in multi-word categories.
- The words obtained are converted into a term weight vector, each word with its IDF weight in the training collection. For each category, a vector is produced.
- We take the term-weight vector for each category, as the initial vector for a linear learning algorithm (the Rocchio one in our experiments). We run the algorithm on the training collection, generating a prototype vector.

For the learning step, training documents are represented as term-weight vectors according to the Vector Space Model: terms are word stems occurring in between the 1% and 10% of documents; weights are TF.IDF like. When a new document is to be classified, it is represented as a term-weight vector, and its cosine similarity to each category prototype is computed. The document is assigned to the category if the similarity exceeds a predefined threshold.

The WSD step is quite simple. For each of the potential synsets, its neighborhood is found (considering semantic links in WordNet as a graph). Each candidate synset is associated to a set of words extracted from the closest synsets, and we compute the overlap between this set of words and the whole set of words of documents in the category. The synset that shows higher overlap is selected.

This is a very simple WSD based on dictionaries, that exploits the information available for the category in the form of training documents, and the information available for senses, in the form of words in closest synsets.

⁵ This method is fully described in [19, 20].

5.2 Experiments

In order to test this WSD method, we have extended previous experiments to this WSD algorithm for this work. We make use of the Rocchio linear learning algorithm as described in [19]. We work on the Reuters-21578 test collection, ModApte Split.

The Reuters-21578 collection consists of 21,578 newswire articles from Reuters collected during 1987. Documents in Reuters deal with financial topics, and were classified in several sets of financial categories by personnel from Reuters Ltd. and Carnegie Group Inc. Documents vary in length and number of categories assigned, from 1 line to more than 50, and from none categories to more than 8. There are five sets of categories: TOPICS, ORGANIZATIONS, EXCHANGES, PLACES, and PEOPLE. As others before, we have selected the 90 TOPICS for our experiments. In the ModApte Split, there are 9,603 documents in the training collection, and 3,299 in the test collection.

In the Table 1, we present the results of our experiments, comparing the usage of WordNet with the dictionary WSD method, and a baseline learning method based only on the Rocchio algorithm. We show the F_1 ⁶ values obtained by macro (MF1) and micro averaging (mF1), grouping the categories by the number of training documents available for them (Docs/Topic). This way, we also show how WordNet information is specially valuable in the case of less popular categories, on which learning is limited by insufficient training data. Also, we provide the number of categories (#Topics), the average number of documents in the training (AvgTr) and test (AvgTs) collections, for each of the groups of categories.

Table 1. Query Expansion and dictionary WSD evaluation results

Docs/Topic	#Topics	AvgTr	AvgTs	No-WN		WN	
				MF1	mF1	MF1	mF1
1-5	20	1,15	1,55	0,140	0,179	0,397	0,464
6-19	16	7,56	4,13	0,163	0,202	0,373	0,419
21-50	19	20,11	10,89	0,191	0,191	0,545	0,562
51-99	14	46,79	19,21	0,329	0,327	0,601	0,595
100-999	19	199,47	71,89	0,629	0,647	0,645	0,651
1000-	2	2262,50	903,00	0,864	0,864	0,812	0,805
TOTAL	90	105,51	41,61	0,303	0,696	0,517	0,712

The results shown in the table are very encouraging. Comparing the four right columns, it can be seen that macro-averaged improvements are bigger than

⁶ F_1 is the standard quality metric in ATC. It averages recall and precision. Macro-averaged F_1 gives equal importance to all categories, while micro-averaged F_1 gives more importance to popular categories. It is important to show both [5].

micro-averaged ones. Thus, those categories with less learning documents are get more benefit from the WordNet enrichment. This is also confirmed by the fact that performance improvements are achieved for all category groups, except the two more frequent categories. This also suggests that the weights used in the Rocchio algorithm (and in general, the importance given to the information extracted from WordNet, versus the information in the training collection) can be balanced in proportion with the number of training documents.

In general, ATC effectiveness is substantially improved. In our opinion, perfect WSD is not required; the method can tolerate the mistakes performed by a simple dictionary WSD method, still improving effectiveness.

6 Concept Indexing method

In this section, we present the Concept Indexing method for integrating lexical-semantic information in learning based ATC. Also, we detail our baseline statistical WSD method, and test it in a systematic way. The results of our experiments show that no perfect WSD is required for alleviating language variability problems in ATC.

6.1 Description

The basic model for Concept Indexing, sketched above, consist of using concepts (WordNet synsets in our case) as indexing units in text representation. The rest of the process for learning based ATC is kept as usual. Instead of this simple formulation, and following [8] ideas, we have appended two representations: a word based representation, and a concept based one. This one, a text document is represented as a term and concept weight vector.

We let the selection step to decide which are the most representative indexing units; depending on the category, there can be more synsets than words, or the opposite, as indexing units. This way, we exploit the selection methods and the information contained both in the training document collection, and the lexical-semantic resource.

6.2 Word Sense Disambiguation method

This method requires WSD, because it is needed to identify the correct sense (synset) for each word in a document. In previous experiments [18], we found some evidence supporting that perfect WSD was required if only concepts were used as indexing units. Since we use here synsets and words, we can use a simple WSD based on sense frequency and POS tags. For each word, we assign it the most frequent sense for the POS tag it has in the context. This method is used as baseline in SENSEVAL experiments, and many sophisticate WSD are unable to reach its effectiveness.

6.3 Experiments

We have performed several series of experiments, using the SemCor text collection. The SemCor text collection is a Semantic Concordance, a corpus tagged with WordNet senses in order to supplement WordNet itself (for instance, for researching or showing examples of sense usage). However, SemCor has been adapted and used for testing IR by Gonzalo *et al.* [11], and used for evaluating TC by Petridis *et al.* [17]. Moreover, there are not other collections tagged with conceptual information in depth, and so, indexing with “perfect” disambiguation can hardly be tested without SemCor. However, we have changed manually labeled references to second and other senses to sense one, simulating the statistical WSD method proposed above. SemCor has 15 genre oriented classes, and 147 documents. Given this scarce information, we have used the 10-fold cross validation evaluation methodology [5].

In the Table 2, we present the results of our experiments. We have tested a synset based representation, a word based representation, and the combined approach, using a representative range of high performance learning algorithms. From those tested in the literature [5], we have selected the following ones⁷: the probabilistic approach Naive Bayes (NB); the decision tree learner C4.5 (C45); the Support Vector Machines kernel method (SVM); and the AdaBoost meta-learner applied to Naive Bayes (ABNB). We present macro (MF1) and micro averaged (mF1) F_1 values.

Table 2. Concept indexing and statistical WSD evaluation results

Alg.	Synsets		Words		Combined	
	MF1	mF1	MF1	mF1	MF1	mF1
NB	0,631	0,739	0,635	0,750	0,702	0,837
C45	0,258	0,391	0,270	0,382	0,280	0,396
SVM	0,502	0,773	0,482	0,730	0,467	0,763
ABNB	0,638	0,759	0,638	0,760	0,682	0,824

Given the results, it is clear that the top performing method is NB operating on the combined representation. While AdaBoost usually improves its base method, and this is observed on the synset and word based representations, it has not been able to do so on the combined representation. This may be due to the fact that AdaBoost improves *weak* methods, but Naive Bayes is proved very accurate on this particular problem. Also, and more in general, the combined representation improves synset and word based representations for all the algorithms, except for Support Vector Machines. The decision tree learner can be considered a weak baseline for this problem.

⁷ We exclude some references for brevity. Please find a sample at [5].

Under the light of these results, we believe that no advanced WSD is needed, and still ATC can be greatly improved. Still, bigger improvements may be achieved by using the nearest synsets to those occurring in the documents (through hyponymy and meronymy WordNet relations). Also, these results should be confirmed by testing similar representations on a bigger test collection, like the Reuters-21578 one.

7 Conclusions

With the aim of reducing language variability, we have sketched two method for integrating lexical-semantic information in learning ATC. These methods, based on traditional work in IR, are called Query Expansion and Concept Indexing. In both methods, WSD plays a key role, either disambiguating categories names, or training and testing documents. We have designed simple WSD methods for each model, based on current work in the area.

We have conducted several series of experiments with the LDB WordNet and the standard Reuters-21578 and SemCor collections. The results of these experiments demonstrate that no heavy, full WSD is required, still improving ATC effectiveness. This results are very encouraging, and contrast with the mixed results presented in other works in IR and ATC.

References

1. Zhdanova, A.V., Shishkin, D.V.: Classification of email queries by topic: Approach based on hierarchically structured subject domain. In Yin, H., Allinson, N., Freeman, R., Keane, J., Hubbard, S., eds.: Proceedings of IDEAL-02, 3rd International Conference on Intelligent Data Engineering and Automated Learning, Manchester, UK, Springer Verlag, Heidelberg, DE (2002) 99–104 Published in the “Lecture Notes in Computer Science” series, number 2412.
2. Mladenić, D.: Turning YAHOO! into an automatic Web page classifier. In Prade, H., ed.: Proceedings of ECAI-98, 13th European Conference on Artificial Intelligence, Brighton, UK, John Wiley and Sons, Chichester, UK (1998) 473–474
3. Gómez, J.: Evaluating cost-sensitive unsolicited bulk email categorization. In: Proceedings of SAC-02, 17th ACM Symposium on Applied Computing, Madrid, ES (2002) 615–620
4. Hepple, M., Ireson, N., Allegrini, P., Marchi, S., Montemagni, S., Gómez, J.: NLP-enhanced content filtering within the POESIA project. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004). (2004)
5. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* **34** (2002) 1–47
6. Van Rijsbergen, C.J.: *Information Retrieval*. Butterworths, London (1979)
7. Miller, G.A.: WordNet: A lexical database for English. *Communications of the ACM* **38** (1995) 39–41
8. Voorhees, E.M.: Using wordnet to disambiguate word sense for text retrieval. In: Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval, Pittsburgh, US (1993) 171–180

9. Voorhees, E.M.: Query expansion using lexical-semantic relations. In Croft, W.B., van Rijsbergen, C.J., eds.: Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval, London, UK, Springer Verlag (1994) 61–70
10. Voorhees, E.: Using WordNet for text retrieval. In: WordNet: An Electronic Lexical Database. MIT Press (1998)
11. Gonzalo, J., Verdejo, F., Chugur, I., Cigarrán, J.: Indexing with WordNet synsets can improve text retrieval. In: Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems. (1998)
12. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison Wesley (1989)
13. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: Proc. Of the 14th International Conf. On Machine Learning. (1997)
14. Yang, Y., Liu, X.: A re-examination of text categorization methods. In Hearst, M.A., Gey, F., Tong, R., eds.: Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, US, ACM Press, New York, US (1999) 42–49
15. Scott, S.: Feature engineering for a symbolic approach to text classification. Master's thesis, Computer Science Dept., University of Ottawa, Ottawa, CA (1998)
16. Fukumoto, F., Suzuki, Y.: Learning lexical representation for text categorization. In: Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources. (2001)
17. Petridis, V., Kaburlasos, V., Fragkou, P., Kehagias, A.: Text classification using the σ -FLNMAP neural network. In: Proceedings of the 2001 International Joint Conference on Neural Networks. (2001)
18. Gómez, J., Cortizo, J., Puertas, E., Ruíz, M.: Concept indexing for automated text categorization. In: Natural Language Processing and Information Systems: 9th International Conference on Applications of Natural Language to Information Systems, NLDB 2004, Salford, UK, June 23-25, 2004, Proceedings. Lecture Notes in Computer Science, Vol. 3136, Springer (2004) 195–206
19. de Buenaga Rodríguez, M., Gómez Hidalgo, J., Díaz Agudo, B.: Using wordnet to complement training information in text categorization. In Nicolov, N., Mitkov, R., eds.: Recent Advances in Natural Language Processing II: Selected Papers from RANLP'97. Volume 189 of Current Issues in Linguistic Theory (CILT)., John Benjamins (2000) 353–364
20. Ureña-López, L.A., Buenaga, M., Gómez, J.M.: Integrating linguistic resources in TC through WSD. Computers and the Humanities **35** (2001) 215–230
21. Benkhalifa, M., Mouradi, A., Bouyahf, H.: Integrating external knowledge to supplement training data in semi-supervised learning for text categorization. Information Retrieval **4** (2001) 91–113
22. Manning, C., Schütze, H.: 16: Text Categorization. In: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, US (1999) 575–608