A Study on Automatically Extracted Keywords in Text Categorization

Anette Hulth and **Beáta B. Megyesi**
Department of Linguistics and Philology
Uppsala University, Sweden
anette.hulth@gmail.com bea@stp.lingfil.uu.se

Abstract

This paper presents a study on if and how automatically extracted keywords can be used to improve text categorization. In summary we show that a higher performance — as measured by micro-averaged F-measure on a standard text categorization collection — is achieved when the full-text representation is combined with the automatically extracted keywords. The combination is obtained by giving higher weights to words in the full-texts that are also extracted as keywords. We also present results for experiments in which the keywords are the only input to the categorizer, either represented as unigrams or intact. Of these two experiments, the unigrams have the best performance, although neither performs as well as headlines only.

1 Introduction

Automatic text categorization is the task of assigning any of a set of predefined categories to a document. The prevailing approach is that of *supervised machine learning*, in which an algorithm is trained on documents with known categories. Before any learning can take place, the documents must be represented in a form that is understandable to the learning algorithm. A trained *prediction model* is subsequently applied to previously unseen documents, to assign the categories. In order to perform a text categorization task, there are two major decisions to make: how to represent the text, and what learning algorithm to use to create the prediction model. The decision about the representation is in turn divided into two subquestions: what features to select as input and which type of value to assign to these features.

In most studies, the best performing representation consists of the full length text, keeping the tokens in the document separate, that is as unigrams. In recent years, however, a number of experiments have been performed in which richer representations have been evaluated. For example, Caropreso et al. (2001) compare unigrams and bigrams; Moschitti et al. (2004) add complex nominals to their bag-of-words representation, while Kotcz et al. (2001), and Mihalcea and Hassan (2005) present experiments where automatically extracted sentences constitute the input to the representation. Of these three examples, only the sentence extraction seems to have had any positive impact on the performance of the automatic text categorization.

In this paper, we present experiments in which keywords, that have been automatically extracted, are used as input to the learning, both on their own and in combination with a full-text representation. That the keywords are extracted means that the selected terms are present verbatim in the document. A keyword may consist of one or several tokens. In addition, a keyword may well be a whole expression or phrase, such as *snakes and ladders*. The main goal of the study presented in this paper is to investigate if automatically extracted keywords can improve automatic text categorization. We investigate what impact keywords have on the task by predicting text categories on the basis of keywords only, and by combining full-text representations with automatically extracted keywords. We also experiment with different ways of representing keywords, either as unigrams or intact. In addition, we investigate the effect of using the headlines — represented as unigrams — as input,

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 537–544,
Sydney, July 2006.

to compare their performance to that of the keywords.

The outline of the paper is as follows: in Section 2, we present the algorithm used to automatically extract the keywords. In Section 3, we present the corpus, the learning algorithm, and the experimental setup for the performed text categorization experiments. In Section 4, the results are described. An overview of related studies is given in Section 5, and Section 6 concludes the paper.

2 Selecting the Keywords

This section describes the method that was used to extract the keywords for the text categorization experiments discussed in this paper. One reason why this method, developed by Hulth (2003; 2004), was chosen is because it is tuned for short texts (more specifically for scientific journal abstracts). It was thus suitable for the corpus used in the described text categorization experiments.

The approach taken to the automatic keyword extraction is that of supervised machine learning, and the prediction models were trained on manually annotated data. No new training was done on the text categorization documents, but models trained on other data were used. As a first step to extract keywords from a document, candidate terms are selected from the document in three different manners. One term selection approach is statistically oriented. This approach extracts all uni-, bi-, and trigrams from a document. The two other approaches are of a more linguistic character, utilizing the words' parts-of-speech (PoS), that is, the word class assigned to a word. One approach extracts all noun phrase (NP) chunks, and the other all terms matching any of a set of empirically defined PoS patterns (frequently occurring patterns of manual keywords). All candidate terms are stemmed.

Four features are calculated for each candidate term: term frequency; inverse document frequency; relative position of the first occurrence; and the PoS tag or tags assigned to the candidate term. To make the final selection of keywords, the three predictions models are combined. Terms that are subsumed by another keyword selected for the document are removed. For each selected stem, the most frequently occurring unstemmed form in the document is presented as a keyword. Each document is assigned at the most twelve keywords, provided that the added regression value (given by the prediction models) is higher than an empirically defined threshold value. To avoid that a document gets no keywords, at least three keywords are assigned although the added regression value is below the threshold (provided that there are at least three candidate terms).

Assign. mean	**Corr. mean**	**P**	**R**	**F**
8.6	3.6	41.5	46.9	44.0

Table 1: The number of assigned (Assign.) keywords in mean per document; the number of correct (Corr.) keywords in mean per document; precision (P); recall (R); and F-measure (F), when 3–12 keywords are extracted per document.

In Hulth (2004) an evaluation on 500 abstracts in English is presented. For the evaluation, keywords assigned to the test documents by professional indexers are used as a gold standard, that is, the manual keywords are treated as the one and only truth. The evaluation measures are *precision* (how many of the automatically assigned keywords that are also manually assigned keywords) and *recall* (how many of the manually assigned keywords that are found by the automatic indexer). The third measure used for the evaluations is the *F-measure* (the harmonic mean of precision and recall). Table 1 shows the result on that particular test set. This result may be considered to be state-of-the-art.

3 Text Categorization Experiments

This section describes in detail the four experimental settings for the text categorization experiments.

3.1 Corpus

For the text categorization experiments we used the *Reuters-21578 corpus*, which contains 20 000 newswire articles in English with multiple categories (Lewis, 1997). More specifically, we used the *ModApte* split, containing 9 603 documents for training and 3 299 documents in the fixed test set, and the 90 categories that are present in both training and test sets.

As a first pre-processing step, we extracted the texts contained in the TITLE and BODY tags. The pre-processed documents were then given as input to the keyword extraction algorithm. In Table 2, the number of keywords assigned to the doc-

uments in the training set and the test set are displayed. As can be seen in this table, three is the number of keywords that is most often extracted. In the training data set, 9 549 documents are assigned keywords, while 54 are empty, as they have no text in the TITLE or BODY tags. Of the 3 299 documents in the test set, 3 285 are assigned keywords, and the remaining fourteen are those that are empty. The empty documents are included in the result calculations for the fixed test set, in order to enable comparisons with other experiments. The mean number of keyword extracted per document in the training set is 6.4 and in the test set 6.1 (not counting the empty documents).

Keywords	Training docs	Test docs
0	54	14
1	68	36
2	829	272
3	2 016	838
4	868	328
5	813	259
6	770	252
7	640	184
8	527	184
9	486	177
10	688	206
11	975	310
12	869	239

Table 2: Number of automatically extracted keywords per document in training set and test set respectively.

3.2 Learning Method

The focus of the experiments described in this paper was the text representation. For this reason, we used only one learning algorithm, namely an implementation of *Linear Support Vector Machines* (Joachims, 1999). This is the learning method that has obtained the best results in text categorization experiments (Dumais et al., 1998; Yang and Liu, 1999).

3.3 Representations

This section describes in detail the input representations that we experimented with. An important step for the feature selection is the dimensionality reduction, that is reducing the number of features. This can be done by removing words that are rare (that occur in too few documents or have too low term frequency), or very common (by applying a stop-word list). Also, terms may be stemmed, meaning that they are merged into a common form. In addition, any of a number of feature selection metrics may be applied to further reduce the space, for example chi-square, or information gain (see for example Forman (2003) for a survey).

Once that the features have been set, the final decision to make is what feature value to assign. There are to this end three common possibilities: a boolean representation (that is, the term exists in the document or not), term frequency, or tf*idf.

Two sets of experiments were run in which the automatically extracted keywords were the only input to the representation. In the first set, keywords that contained several tokens were kept intact. For example a keyword such as *paradise fruit* was represented as `paradise_fruit` and was — from the point of view of the classifier — just as distinct from the single token *fruit* as from *meatpackers*. No stemming was performed in this set of experiments.

In the second set of keywords-only experiments, the keywords were split up into unigrams, and also stemmed. For this purpose, we used Porter's stemmer (Porter, 1980). Thereafter the experiments were performed identically for the two keyword representations.

In a third set of experiments, we extracted only the content in the TITLE tags, that is, the headlines. The tokens in the headlines were stemmed and represented as unigrams. The main motivation for the title experiments was to compare their performance to that of the keywords.

For all of these three feature inputs, we first evaluated which one of the three possible feature values to use (boolean, tf, or tf*idf). Thereafter, we reduced the space by varying the minimum number of occurrences in the training data, for a feature to be kept.

The starting point for the fourth set of experiments was a full-text representation, where all stemmed unigrams occurring three or more times in the training data were selected, with the feature value tf*idf. Assuming that extracted keywords convey information about a document's gist, the feature values in the full-text representation were given higher weights if the feature was identical to a keyword token. This was achieved by adding the term frequency of a full-text unigram to the term

frequency of an identical keyword unigram. Note that this does not mean that the term frequency value was necessarily doubled, as a keyword often contains more than one token, and it was the term frequency of the whole keyword that was added.

3.4 Training and Validation

This section describes the parameter tuning, for which we used the training data set. This set was divided into five equally sized folds, to decide which setting of the following two parameters that resulted in the best performing classifier: what feature value to use, and the threshold for the minimum number of occurrence in the training data (in this particular order).

To obtain a baseline, we made a full-text unigram run with boolean as well as with tf*idf feature values, setting the occurrence threshold to three.

As stated previously, in this study, we were concerned only with the representation, and more specifically with the feature input. As we did not tune any other parameters than the two mentioned above, the results can be expected to be lower than the state-of-the art, even for the full-text run with unigrams.

The number of input features for the full-text unigram representation for the whole training set was 10 676, after stemming and removing all tokens that contained only digits, as well as those tokens that occurred less than three times. The total number of keywords assigned to the 9 603 documents in the training data was 61 034. Of these were 29 393 unique. When splitting up the keywords into unigrams, the number of unique stemmed tokens was 11 273.

3.5 Test

As a last step, we tested the best performing representations in the four different experimental settings on the independent test set.

The number of input features for the full-text unigram representation was 10 676. The total number of features for the intact keyword representation was 4 450 with the occurrence threshold set to three, while the number of stemmed keyword unigrams was 6 478, with an occurrence threshold of two. The total number of keywords extracted from the 3 299 documents in the test set was 19 904.

Next, we present the results for the validation and test procedures.

4 Results

To evaluate the performance, we used *precision*, *recall*, and *micro-averaged F-measure*, and we let the F-measure be decisive. The results for the 5-fold cross validation runs are shown in Table 3, where the values given are the average of the five runs made for each experiment. As can be seen in this table, the full-text run with a boolean feature value gave 92.3% precision, 69.4% recall, and 79.2% F-measure. The full-text run with tf*idf gave a better result as it yielded 92.9% precision, 71.3% recall, and 80.7% F-measure. Therefore we defined the latter as baseline.

In the first type of the experiment where each keyword was treated as a feature independently of the number of tokens contained, the recall rates were considerably lower (between 32.0% and 42.3%) and the precision rates were somewhat lower (between 85.8% and 90.5%) compared to the baseline. The best performance was obtained when using a boolean feature value, and setting the minimum number of occurrence in training data to three (giving an F-measure of 56.9%).

In the second type of experiments, where the keywords were split up into unigrams and stemmed, recall was higher but still low (between 60.2% and 64.8%) and precision was somewhat lower (88.9–90.2%) when compared to the baseline. The best results were achieved with a boolean representation (similar to the first experiment) and the minimum number of occurrence in the training data set to two (giving an F-measure of 75.0%)

In the third type of experiments, where only the text in the TITLE tags was used and was represented as unigrams and stemmed, precision rates increased above the baseline to 93.3–94.5%. Here, the best representation was tf*idf with a token occurring at least four times in the training data (with an F-measure of 79.9%).

In the fourth and last set of experiments, we gave higher weights to full-text tokens if the same token was present in an automatically extracted keyword. Here we obtained the best results. In these experiments, the term frequency of a keyword unigram was added to the term frequency for the full-text features, whenever the stems were identical. For this representation, we experimented with setting the minimum number of occurrence in training data both before and after that the term frequency for the keyword token was added to the term frequency of the unigram. The

Input feature	Feature value	Min. occurrence	Precision	Recall	F-measure
full-text unigram	bool	3	92.31	69.40	79.22
full-text unigram	tf*idf	3	92.89	71.30	80.67
keywords-only intact	bool	1	90.54	36.64	52.16
keywords-only intact	tf	1	88.68	33.74	48.86
keywords-only intact	tf*idf	1	89.41	32.05	47.18
keywords-only intact	bool	2	89.27	40.43	55.64
keywords-only intact	bool	3	87.11	42.28	56.90
keywords-only intact	bool	4	85.81	41.97	56.35
keywords-only unigram	bool	1	89.12	64.61	74.91
keywords-only unigram	tf	1	89.89	60.23	72.13
keywords-only unigram	tf*idf	1	90.17	60.36	72.31
keywords-only unigram	bool	2	89.02	64.83	75.02
keywords-only unigram	bool	3	88.90	64.82	74.97
title	bool	1	94.17	68.17	79.08
title	tf	1	94.37	67.89	78.96
title	tf*idf	1	**94.46**	68.49	79.40
title	tf*idf	2	93.92	69.19	79.67
title	tf*idf	3	93.75	69.65	79.91
title	tf*idf	4	93.60	69.74	79.92
title	tf*idf	5	93.31	69.40	79.59
keywords+full	tf*idf	3 (before adding)	92.73	**72.02**	**81.07**
keywords+full	tf*idf	3 (after adding)	92.75	71.94	81.02

Table 3: The average results from 5-fold cross validations for the baseline candidates and the four types of experiments, with various parameter settings.

highest recall (72.0%) and F-measure (81.1%) for all validation runs were achieved when the occurrence threshold was set before the addition of the keywords.

Next, the results on the fixed test data set for the four experimental settings with the best performance on the validation runs are presented.

Table 4 shows the results obtained on the fixed test data set for the baseline and for those experiments that obtained the highest F-measure for each one of the four experiment types.

We can see that the baseline — where the full-text is represented as unigrams with tf*idf as feature value — yields 93.0% precision, 71.7% recall, and 81.0% F-measure. When the intact keywords are used as feature input with a boolean feature value and at least three occurrences in training data, the performance decreases greatly both considering the correctness of predicted categories and the number of categories that are found.

When the keywords are represented as unigrams, a better performance is achieved than when they are kept intact. This is in line with the findings on *n*-grams by Caropreso et al. (2001). However, the results are still not satisfactory since both the precision and recall rates are lower than the baseline.

Titles, on the other hand, represented as unigrams and stemmed, are shown to be a useful information source when it comes to correctly predicting the text categories. Here, we achieve the highest precision rate of 94.2% although the recall rate and the F-measure are lower than the baseline.

Full-texts combined with keywords result in the highest recall value, 72.9%, as well as the highest F-measure, 81.7%, both above the baseline.

Our results clearly show that automatically extracted keywords can be a valuable supplement to full-text representations and that the combination of them yields the best performance, measured as both recall and micro-averaged F-measure. Our experiments also show that it is possible to do a satisfactory categorization having only keywords, given that we treat them as unigrams. Lastly, for higher precision in text classification, we can use the stemmed tokens in the headlines as features

Input feature	Feature value	Min. occurrence	Precision	Recall	F-measure
full-text unigram	tf*idf	3	93.03	71.69	80.98
keywords-only intact	bool	3	89.56	41.48	56.70
keywords-only unigram	bool	2	90.23	64.16	74.99
title	tf*idf	4	**94.23**	68.43	79.28
keywords+full	tf*idf	3	92.89	**72.94**	**81.72**

Table 4: Results on the fixed test set.

with tf*idf values.

As discussed in Section 2 and also presented in Table 2, the number of keywords assigned per document varies from zero to twelve. In Figure 1, we have plotted how the precision, the recall, and the F-measure for the test set vary with the number of assigned keywords for the keywords-only unigram representation.

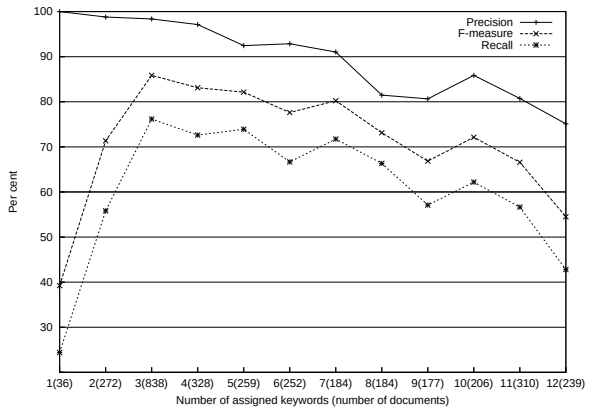

Figure 1: Precision, recall, and F-measure for each number of assigned keywords. The values in brackets denote the number of documents.

We can see that the F-measure and the recall reach their highest points when three keywords are extracted. The highest precision (100%) is obtained when the classification is performed on a single extracted keyword, but then there are only 36 documents present in this group, and the recall is low. Further experiments are needed in order to establish the optimal number of keywords to extract.

5 Related Work

For the work presented in this paper, there are two aspects that are of interest in previous work. These are in how the alternative input features (that is, alternative from unigrams) are selected and in how this alternative representation is used in combination with a bag-of-words representation (if it is).

An early work on linguistic phrases is done by Fürnkranz et al. (1998), where all noun phrases matching any of a number of syntactic heuristics are used as features. This approach leads to a higher precision at the low recall end, when evaluated on a corpus of Web pages. Aizawa (2001) extracts PoS-tagged compounds, matching predefined PoS patterns. The representation contains both the compounds and their constituents, and a small improvement is shown in the results on Reuters-21578. Moschitti and Basili (2004) add complex nominals as input features to their bag-of-words representation. The phrases are extracted by a system for terminology extraction[1]. The more complex representation leads to a small decrease on the Reuters corpus. In these studies, it is unclear how many phrases that are extracted and added to the representations.

Li et al. (2003) map documents (e-mail messages) that are to be classified into a vector space of keywords with associated probabilities. The mapping is based on a training phase requiring both texts and their corresponding summaries.

Another approach to combine different representations is taken by Sahlgren and Cöster (2004), where the full-text representation is combined with a concept-based representation by selecting one or the other for each category. They show that concept-based representations can outperform traditional word-based representations, and that a combination of the two different types of representations improves the performance of the classifier over all categories.

Keywords assigned to a particular text can be seen as a dense summary of the same. Some reports on how automatic summarization can be used to improve text categorization exist. For ex-

[1]In terminology extraction all terms describing a domain are to be extracted. The aim of automatic keyword indexing, on the other hand, is to find a small set of terms that describes a specific document, independently of the domain it belongs to. Thus, the set of terms must be limited to contain only the most salient ones.

ample, Ko et al. (2004) use methods from text summarization to find the sentences containing the important words. The words in these sentences are then given a higher weight in the feature vectors, by modifying the term frequency value with the sentence's score. The F-measure increases from 85.8 to 86.3 on the *Newsgroups* data set using Support vector machines.

Mihalcea and Hassan (2004) use an unsupervised method[2] to extract summaries, which in turn are used to categorize the documents. In their experiments on a sub-set of Reuters-21578 (among others), Mihalcea and Hassan show that the precision is increased when using the summaries rather than the full length documents. Özgür et al. (2005) have shown that limiting the representation to 2 000 features leads to a better performance, as evaluated on Reuters-21578. There is thus evidence that using only a sub-set of a document can give a more accurate classification. The question, though, is which sub-set to use.

In summary, the work presented in this paper has the most resemblance with the work by Ko et al. (2004), who also use a more dense version of a document to alter the feature values of a bag-of-words representation of a full-length document.

6 Concluding Remarks

In the experiments described in this paper, we investigated if automatically extracted keywords can improve automatic text categorization. More specifically, we investigated what impact keywords have on the task of text categorization by making predictions on the basis of keywords only, represented either as unigrams or intact, and by combining the full-text representation with automatically extracted keywords. The combination was obtained by giving higher weights to words in the full-texts that were also extracted as keywords. Throughout the study, we were concerned with the data representation and feature selection procedure. We investigated what feature value should be used (boolean, tf, or tf*idf) and the minimum number of occurrence of the tokens in the training data.

We showed that keywords can improve the performance of the text categorization. When keywords were used as a complement to the full-text representation an F-measure of 81.7% was obtained, higher than without the keywords (81.0%). Our results also clearly indicate that keywords alone can be used for the text categorization task when treated as unigrams, obtaining an F-measure of 75.0%. Lastly, for higher precision (94.2%) in text classification, we can use the stemmed tokens in the headlines.

The results presented in this study are lower than the state-of-the-art, even for the full-text run with unigrams, as we did not tune any other parameters than the feature values (boolean, term frequency, or tf*idf) and the threshold for the minimum number of occurrence in the training data.

There are, of course, possibilities for further improvements. One possibility could be to combine the tokens in the headlines and keywords in the same way as the full-text representation was combined with the keywords. Another possible improvement concerns the automatic keyword extraction process. The keywords are presented in order of their estimated "keywordness", based on the added regression value given by the three prediction models. This means that one alternative experiment would be to give different weights depending on which rank the keyword has achieved from the keyword extraction system. Another alternative would be to use the actual regression value.

We would like to emphasize that the automatically extracted keywords used in our experiments are not statistical phrases, such as bigrams or trigrams, but meaningful phrases selected by including linguistic analysis in the extraction procedure.

One insight that we can get from these experiments is that the automatically extracted keywords, which themselves have an F-measure of 44.0, can yield an F-measure of 75.0 in the categorization task. One reason for this is that the keywords have been evaluated using manually assigned keywords as the gold standard, meaning that paraphrasing and synonyms are severely punished. Kotcz et al. (2001) propose to use text categorization as a way to more objectively judge automatic text summarization techniques, by comparing how well an automatic summary fares on the task compared to other automatic summaries (that is, as an *extrinsic* evaluation method). The same would be valuable for automatic keyword indexing. Also, such an approach would facilitate comparisons between different systems, as common test-beds are lacking.

[2] This method has also been used to extract keywords (Mihalcea and Tarau, 2004).

In this study, we showed that automatic text categorization can benefit from automatically extracted keywords, although the bag-of-words representation is competitive with the best performance. Automatic keyword extraction as well as automatic text categorization are research areas where further improvements are needed in order to be useful for more efficient information retrieval.

Acknowledgments

The authors are grateful to the anonymous reviewers for their valuable suggestions on how to improve the paper.

References

Akiko Aizawa. 2001. Linguistic techniques to improve the performance of automatic text categorization. In *Proceedings of NLPRS-01, 6th Natural Language Processing Pacific Rim Symposium*, pages 307–314.

Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. 2001. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In *Text Databases and Document Management: Theory and Practice*, pages 78–102.

Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management (CIKM'98)*, pages 148–155.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, March.

Johannes Fürnkranz, Tom Mitchell, and Ellen Riloff. 1998. A case study using linguistic phrases for text categorization on the WWW. In *AAAI-98 Workshop on Learning for Text Categorization*.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, pages 216–223.

Anette Hulth. 2004. *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction*. Ph.D. thesis, Department of Computer and Systems Sciences, Stockholm University.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*. MIT-Press.

Youngjoong Ko, Jinwoo Park, and Jungyun Seo. 2004. Improving text categorization using the importance of sentences. *Information Processing and Management*, 40(1):65–79.

Aleksander Kolcz, Vidya Prabakarmurthi, and Jugal Kalita. 2001. Summarization as feature selection for text categorization. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM'01)*, pages 365–370.

David D. Lewis. 1997. Reuters-21578 text categorization test collection, Distribution 1.0. AT&T Labs Research.

Cong Li, Ji-Rong Wen, and Hang Li. 2003. Text classification using stochastic keyword generation. In *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*.

Rada Mihalcea and Samer Hassan. 2005. Using the essence of texts to improve document classification. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP 2005)*.

Rada Mihalcea and Paul Tarau. 2004. TextRank: bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*.

Alessandro Moschitti and Roberto Basili. 2004. Complex linguistic features for text classification: A comprehensive study. In Sharon McDonald and John Tait, editors, *Proceedings of ECIR-04, 26th European Conference on Information Retrieval Research*, pages 181–196. Springer-Verlag.

Arzucan Özgür, Levent Özgür, and Tunga Güngör. 2005. Text categorization with class-based and corpus-based keyword selection. In *Proceedings of the 20th International Symposium on Computer and Information Sciences*, volume 3733 of *Lecture Notes in Computer Science*, pages 607–616. Springer-Verlag.

Martin Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Magnus Sahlgren and Rickard Cöster. 2004. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 487–493.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49.