Using Bigrams in Text Categorization

RON BEKKERMAN JAMES ALLAN

Department of Computer Science
University of Massachusetts
Amherst, 01003 USA

{ronb|allan}@cs.umass.edu

December 27, 2003

Abstract

In the past decade a sufficient effort has been expended on attempting to come up with a document representation which is richer than the simple Bag-Of-Words (BOW). One of the widely explored approaches to enrich the BOW representation is in using n-grams (usually bigrams) of words in addition to (or in place of) single words (unigrams). After more than ten years of unsuccessful attempts to improve the text categorization results by applying bigrams, many researchers agree that there might be a certain limitation in usability of bigrams for text categorization. We analyze the related works and discuss possible reasons for this limitation. In addition, we demonstrate our own attempt to incorporate bigrams in a document representation based on distributional clusters of unigrams, and report (statistically insignificant) improvement to our baseline results on the 20 Newsgroups (20NG) dataset. Nevertheless, the reported result is (to our knowledge) the best categorization result ever achieved on this highly popular dataset.

1 Introduction

Text categorization is a fundamental task in Information Retrieval, and much knowledge in this domain has been accumulated in the past 25 years. The "standard" approach to text categorization has so far been using a document representation in a word-based space, i.e. as a vector in some high dimensional Euclidean space where each dimension corresponds to a word. This method relies on classification algorithms that are trained in a supervised learning manner. Since the early days of text categorization (see, e.g., Salton and McGill, 1983), the theory and practice of classifier design has significantly advanced, and several strong learning algorithms have emerged (see, e.g., Duda et al., 2000; Vapnik, 1998; Schapire and Singer, 2000). In contrast, despite numerous attempts to introduce more sophisticated techniques for document representation, the simple minded independent word-based representation, known as *bag-of-words (BOW)*, remained very effective. Indeed, to date the best multi-class, multi-labeled categorization results for the well-known Reuters-21578 dataset are based on the BOW representation (Dumais et al., 1998; Weiss et al., 1999).

The main drawback of the BOW representation is in destruction of semantic relations between words. Indeed, stable phrases, such as "White House" or "Bill Gates", are represented in the BOW as separated

words so their meaning is lost. Given a BOW of a document in which words "bill" and "gates" occur, one can suggest that the document is about accounting or gardening, but not about computer software. Whereas given a document representation that contains a phrase "bill gates", the reader will hardly be mistaken about the topic of discussion.

These fairly obvious observations led researchers to an idea of enriching the BOW representation by word phrases. In early 90s, Bag-Of-Bigrams (pairs of consequent words) was proposed as a competitive representation (Lewis, 1992). Since then, dozens of articles have been published on this topic. While some of the researchers report significant improvement in text categorization results (Mladenić and Grobelnik, 1998), many of them show only marginal improvement or even a certain decrease.

In this paper we overview the most recent literature on the problem of using bigrams for text categorization. We intentionally do not consider earlier publications, because (a) attempts to summarize their results have already been made before (see, e.g. Tan et al., 2002); and (b) the major increase in computational power and the algorithmic innovations of the past 5 years have opened way to the new generation of text processing techniques. The rest of the paper is organized as follows: in Section 2 we describe the problem of text categorization; in Section 3 we discuss related work; in Section 4 we propose our own method of incorporating bigrams and unigrams in document representations; finally, in Section 5 we outline possible reasons for the failure to improve text categorization results by using bigrams and present our conclusions.

2 Problem Statement

In its simplest form, the text categorization problem can be formulated as follows. We are given a training set $\mathcal{D}_{train} = \{(d_1, \ell_1), \ldots, (d_n, \ell_n)\}$ of labeled text documents where each document d_i belongs to a document set $\mathcal{D}$ and the label $\ell_i = \ell_i(d_i)$ of d_i is within a predefined set of categories $\mathcal{C} = \{c_1, \ldots, c_m\}$. The goal in text categorization is to devise a learning algorithm that given the training set $\mathcal{D}_{train}$ as input will generate a classifier (or a hypothesis) $h : \mathcal{D} \rightarrow \mathcal{C}$ that will be able to accurately classify unseen documents from $\mathcal{D}$.

The design of learning algorithms for text categorization has usually followed the classical approach in pattern recognition, where data instances (i.e. documents) first undergo a transformation of dimensionality reduction, then a classifier learning algorithm is applied to the low-dimensionality representations. This transformation is also performed prior to applying the learned classifier to unseen instances. The incentives in using dimensionality reduction techniques are to improve classification quality (via noise reduction) and to reduce the computational complexity of the learning algorithm and of the application of the classifier to unseen documents.

Dimensionality reduction techniques typically fall into two basic schemes:

- *Feature selection* (or *feature reduction*): These techniques attempt to select the subset of features (e.g. words in text categorization) that are most useful for the categorization task. After the selection of a suitable subset, the reduced representation of a document is computed by projecting the documents over the selected words.

- *Feature generation* (or *feature induction*): New features, which are not necessarily words, are sought for representation. Usually, the new features are synthesized from the original set of features.

There are two common approaches to feature induction. The first one combines features using disjunctions only. In this approach features are grouped into subsets and each such subset is then considered as a new feature. Any occurrence of a member of a subset is then considered as occurrence of the feature. *Stemming* and *word clustering* belong to this family of methods. The second approach groups features using only conjunctions, for example, by grouping consequent or close (in proximity) words into phrases. The use of n*-grams* is a common method in this family.

Disjunction-based methods for feature generation are quite radically different from conjunction-based methods and they achieve different goals. One crucial difference between these methods is that disjunction methods can decrease statistical sparseness while conjunction methods can only increase it. Thus, disjunction methods can decrease variance. On the other hand, conjunction methods can decrease bias. Both disjunction and conjunction methods attempt to preserve semantic relations between words and thus incorporate knowledge into the purely statistical task of categorizing textual documents.

3 Related Work

There exist two main approaches to incorporating bigrams into the document representation: the first one applies bigrams together with unigrams while the second one excludes unigrams from the representation and bases on bigrams only. It turns out that the second approach leads in most cases to a certain decrease in the categorization results in comparison to the BOW, while the first approach can potentially improve the results. This observation indicates that the (intuitively) simple BOW representation is powerful enough so the classification results cannot be probably improved by *replacing* the BOW representation but only by *extending* it.

Even in the setup of extending the BOW representation with bigrams, many researchers report only non-significant improvement. Some, in turn, achieve statistical significance of the difference from the baseline. However, this statistical significance is usually shown on rarely used datasets on which the baseline categorization results are low. These low baseline results are in many cases achieved using non-state-of-the-art classification techniques which probably implies that instead of using bigrams one could use a better classification technique in order to achieve similar improvement with the plain BOW document representation.

Let us list a few recent works in the field:

- Caropreso et al. (2001) experiment with n-grams for text categorization on the Reuters dataset. They define an n-gram as an alphabetically ordered sequence of n stems of consecutive words in a sentence (after stop words were removed). The authors use both unigrams and bigrams as document features. They extract the top-scored features using various feature selection methods including Mutual Information (see, e.g., Dumais et al., 1998). Their results indicate that in general bigrams can better predict categories than unigrams. However, despite the fact that bigrams represent the majority of the top-scored features, the use of bigrams does not yield significant improvement of the categorization results while using the Rocchio classifier. Specifically, in 20 of the 48 reported experiments a certain increase in the accuracy is observed, while in 28 others the accuracy decreases, sometimes quite sharply.

- Scott and Matwin (1999) apply a rule-based RIPPER classifier on Reuters and DigiTrad datasets, using document representation based on phrases. By phrases the authors mean *Noun Phrases* (obtained by a shallow parsing) and *Key Phrases* (the most meaningful phrases obtained by the *Extractor* system). The authors' assumption is that a rule-based classifier could benefit from the semantic power of a highly meaningful phrase. However, the results achieved by either scheme are roughly the same as their baseline with BOW representation. While combining the different representations with the BOW, the authors are able to improve their results, but still the maximum that they achieve is 85% of accuracy on Reuters, whereas the state-of-the-art result is close to 89%. Their results on the rarely used DigiTrad dataset appear significantly better (around 42% of accuracy) in comparison to their baseline as low as 36%.

- Koster and Seutter (2003) use Rocchio and Winnow classifiers on an EPO1A dataset. Their feature induction method involve combination of single words and word pairs. The word pairs are of the

Head/Modifier type, i.e. nouns are extracted with their modifiers. The authors show that when using pairs without BOW the results of both classifiers decrease, while when using both pairs and BOW the results are marginally above the BOW baseline. The authors suggest using clusters of pairs in order to overcome their statistical sparseness.

- Zhang and Lee (2003) apply BOW and bag-of-$n$grams (BON) to the problem of question classification on the TREC10 QA data. They experiment with 5 classifiers: kNN, Naive Bayes, Decision Tree, SNoW and Support Vector Machine (SVM). The authors use two dichotomies of the question collection to 6 coarse grained categories and to 50 fine grained categories. By n-grams the authors mean all continuous word sequences in questions. The results achieved on the coarse dichotomy are essentially the same for the BOW and the BON. While on the fine grained dichotomy the BON shows 1% of advantage over the BOW, which is statistically insignificant. Their highest results are obtained using the SVM classifier.

- Diederich et al. (2003) investigate the problem of authorship attribution which is a special case of the text categorization problem. They apply SVM on two text representations: BOW and a bag of all the functional words and bigrams of functional words in the text. By functional words they mean all the parts of speech excluding nouns, verbs and adjectives. The later document representation is supposed to preserve the style while suppressing the topic. The results show that the simple-minded BOW outperforms the sophisticated representation based on unigrams and bigrams of functional words.

- Tan et al. (2002) report positive results of using bigrams on Reuters and Yahoo! Science datasets. For extracting bigrams they use the following method: first, they sort words according to their document frequency and consider only highly ranked words (let us denote the set of highly ranked words as U). Then they extract bigrams such that at least one of their components belongs to U. After that the authors filter the resulting bigrams according to their *tfidf* and Mutual Information with respect to a category. The authors end up with a set of bigrams that is about 2% of the total number of unigrams considered. After such a tough filtering the bigrams should be especially relevant for the task of distinguishing between the categories. The authors show that bigrams help to increase text categorization results (using naive Bayes classifier) on Yahoo! Science dataset from 65% to 70% break-even point. The improvement is statistically significant, however the baseline is low. On Reuters the improvement is statistically insignificant and again the baseline is low (71.5% of break-even point in comparison to the state-of-the-art result of around 89%). This may indicate that the classification technique used by the authors is weak and therefore any improvement in the technique (including the application of bigrams and many others) would potentially increase the categorization results. Many researchers agree that using the Naive Bayes classifier is a poor choice for text categorization (see, e.g., Dumais et al., 1998).

- One of few relatively successful attempts of using bigrams is demonstrated by Raskutti et al. (2001), who propose a very sophisticated feature induction technique to improve the text categorization results on Reuters and ComputerSelect datasets. They apply a string distance measure which is similar to the String Kernel (Lodhi et al., 2000). Basing on this measure the authors introduce a score according to which they rank bigrams. Then they extract highly ranked bigrams so that less than 1% of the total number of bigrams are extracted. Using the SVM classifier the authors achieve a significant improvement on the ComputerSelect dataset (again the baseline is as low as 41.2% break-even point), while the improvement on the Reuters dataset is again statistically insignificant (0.9% of improvement with respect to their baseline, obtained also by Weiss et al., 1999). Nevertheless, this result on Reuters is highly noticeable: 88.8% break-even point is clearly the state-of-the-art result. The success of this technique may be explained by the fact that documents of the Reuters dataset are very well

structured (many of them are even not free text but tables) and the string similarity method used by the authors manages to capture this clear structure.

The results listed above demonstrate that bigrams have a certain potential for document representation but have never actually proved their effectiveness in the text categorization task.

4 Bigrams in Distributional Clustering

We now propose our own method for feature induction based on distributional clustering of both bigrams and unigrams. Our method is similar but still more intuitive that the one proposed by Tan et al. (2002): we choose bigrams that are clearly most successful for discriminating categories.

In this paper we do not focus on theoretical aspects of distributional clustering, for details see Bekkerman et al. (2003). We only note that the idea behind our approach to the distributional clustering of features (unigrams and bigrams) is to represent each feature as a distribution over the categories of documents in the dataset and then to cluster these distributions so that similar distributions fall into the same cluster. Each document is then represented as a distribution over the *centroids* of feature clusters.

Benefits of this approach are straightforward: we reduce dimensionality and overcome the statistical sparseness of document representations (we control the number of clusters and can therefore make the representations as compact and dense as we wish); in addition, we incorporate domain knowledge in our representation (unigrams and bigrams that are semantically related to each other potentially fall into the same clusters because their distributions over the dataset categories are similar).

The document representation method based on word distributional clustering proved itself to be highly efficient on the popular 20 Newsgroups dataset: while marginally outperforming the categorization results obtained with the BOW representation, it is two orders of magnitude more compact than the BOW representation. Moreover, the best-ever text categorization result of 91.3% accuracy on the 20 Newsgroups was achieved by a distributional clustering application (Bekkerman et al., 2003).

This paper is focused on the question of whether this result can be improved by employing bigrams of words. The previous attempts to incorporate bigrams (described in Section 3) lead us to the following conclusions:

- To improve results that have been achieved, we should enrich the existing model, rather than propose a new one. This implies that we will be considering exactly the same model as in Bekkerman et al. (2003), while our features will now be both unigrams and bigrams. To preserve the existing model as much as we can (in order to ensure at least the same performance), the number of bigrams should be considerably less than the number of unigrams.

- We should guarantee that the extracted bigrams are all highly discriminative features, moreover, they must be more discriminative than the already existing features (unigrams). Thus, we will extract bigrams that are "better" than both their components. Furthermore, we should construct bigrams from unigrams that are *themselves* good enough: we are not interested in bigrams that are "better" than their components only because these components are themselves weak in discrimination between categories.

These two considerations are the basis for our algorithm of extracting bigrams: first, for each category we sort all the unigrams according to their Mutual Information measure with respect to the category. Then we extract k_u top ranked unigrams. Let us denote a set of these unigrams as U. These are our candidates for constructing bigrams. Our new feature set is the set U and all the bigrams (that occur in the training set), both components of which are unigrams from U. We again sort the (new) features according to their Mutual Information with respect to each category and extract only bigrams whose Mutual Information score

is higher than the score of *both* their components. For each category, we acquire k_b top ranked bigrams that satisfy this condition and add them to the general pool of bigrams B. Our feature set is then all the unigrams and the set of best discriminating bigrams B. Algorithm 1 presents the pseudocode of our feature induction procedure. After that, analogously to the method proposed by Bekkerman et al. (2003), we cluster our features to k clusters and represent documents (of both training set and test set) as distributions over the cluster centroids.

Input: D – training set of documents of size N_d;
W – set of all the distinct unigrams in D;
C – set of document categories in D of size N_c;
k_u – threshold on unigrams;
k_b – threshold on bigrams
Output: New representation of documents in dataset
Procedure Feature_Induction
For all $c_i \in C$ **do**
  $W_i \leftarrow$ list of $w \in W$ sorted by $MI(w, c_i)$
  $U_i \leftarrow$ set of k_u top ranked unigrams from W_i
$U \leftarrow \bigcup_{i=1}^{N_c} U_i$ *// All top-ranked unigrams*
For all $d_i \in D$ **do**
  $LU_i \leftarrow$ list of unigrams in d_i that occur in U
  $P_i \leftarrow \emptyset$
  For $j = 2, \ldots,$ number of unigrams in LU_i **do**
    Let $b_j = (LU_i[j-1], LU_i[j])$ be the j-th bigram over LU_i
    $P_i \leftarrow P_i \cup b_j$
$F \leftarrow U \cup (\bigcup_{i=1}^{N_d} P_i)$ *// Top-ranked unigrams and their consequent pairs (bigrams)*
For all $c_i \in C$ **do**
  $F_i \leftarrow$ list of $f \in F$ sorted by $MI(f, c_i)$
  $LB_i \leftarrow$ empty list
  For $j = 1, \ldots,$ number of bigrams in F_i **do**
    Let $b_j = (w_{j1}, w_{j2})$ be the j-th bigram from F_i
    If $MI(b_j, c_i) > \max(MI(w_{j1}, c_i), MI(w_{j2}, c_i))$ **then**
      Push b_j to LB_i
  $B_i \leftarrow$ set of k_b top ranked bigrams from LB_i
$B \leftarrow \bigcup_{i=1}^{N_c} B_i$ *// All top ranked bigrams that are "better" than both their components*
For all $d_i \in D$ **do**
  $BOW_i \leftarrow$ bag of unigrams of d_i
  Represent d_i as $BOW_i \cup (P_i \cap B)$

Algorithm 1: Feature induction procedure

We applied this feature induction algorithm to the 20 Newsgroups dataset. We chose parameters k_u (number of top-ranked unigrams from which bigrams are combined) to be 5000, and k_b (number of top-ranked bigrams that are more discriminating than both their components) to be 1000.

We noticed the following indication of quality of chosen bigrams: since for each one of 20 categories we extract a set of 5000 best discriminating unigrams and then merge these sets, we expect to obtain a maximum of $5000 * 20 = 100,000$ distinct unigrams. However, we have only about 40,000 distinct unigrams, which means that the 20 sets of 5000 best discriminating unigrams are heavily overlapping. In contrast, when we extract 1000 bigrams for each of 20 categories and merge these sets together, we end up with about 19,000 bigrams of 20,000 possible. This means that the 20 sets of best discriminating bigrams are almost

non-overlapping – almost each chosen bigram is especially good for discriminating *one* category from the others.

An analysis of the extracted bigrams showed tight interconnection between the bigram components: many of the bigrams are stable phrases. In Table 4.1 we give a few examples of such bigrams.

1992 93	file stream	next year	spring training
2000 years	find number	operating system	st johns
24 bit	gamma ray	opinions mine	swap file
24 hours	gordon banks	proceeded work	thanks advance
access bus	high jacked	resource listing	today special
after 2000	human rights	right keep	too fast
black panther	instruction set	roads mountain	top ten
burn love	investors packet	running system	tower assembly
cd player	last year	san jose	turn off
chastity intellect	lets go	see note	under windows
closed roads	mail server	self defense	virtual reality
config sys	michael adams	send requests	warning please
considered harmful	mirror sites	serial number	ways escape
court order	model init	shameful surrender	white house
cs cornell	ms windows	skepticism chastity	whos next
east sun	newsletter page	special investors	windows crash
every american	newton apple	spider man	world series

Table 4.1: An example of most informative bigrams extracted from the 20NG dataset (among all its categories).

Despite these good signs, the text categorization results we obtained are not satisfactory. We applied 4-fold cross-validation and used the popular SVM classifier. The achieved result is $91.8 \pm 0.4\%$ of accuracy, whereas our baseline result of the distributional clustering setting on BOW document representations (without bigrams) is $91.3 \pm 0.4\%$. The improvement is clearly statistically insignificant. However, this result is the highest (to our knowledge) text categorization result ever achieved on the 20NG dataset. See Table 4.2 for the summary of the results.

Setting	***Accuracy***
TFIDF feature selection with Rocchio (Joachims, 1997)	90.3%
Distributional clustering of unigrams with SVM (Bekkerman et al., 2003)	$91.3 \pm 0.4\%$
Distributional clustering of unigrams and bigrams with SVM	$91.8 \pm 0.4\%$

Table 4.2: Uni-labeled categorization accuracy for 20NG, obtained using different algorithmic settings.

5 Discussion and conclusion

By using bigrams, researchers obtain a certain improvement in text categorization results only on rarely used datasets for which the baseline is very low and usually obtained by a weak classification method.

On well-known benchmark corpora, such as Reuters-21578 and 20 Newsgroups, statistically significant improvement has never been reported by research groups that employed bigrams in their document representations. This can probably be explained by two considerations: (a) the results achieved on these corpora are so high that they probably cannot be improved by any technique, because all the incorrectly classified items are basically mislabeled; and (b) the corpora are "simple" enough so only a few extracted keywords can do the entire job of distinguishing between categories. Bekkerman et al. (2003) show that the Reuters dataset is indeed an example of the "simple" datasets: when as few as 10 best discriminating words are extracted, the categorization result is above 80% break-even point (BEP) on the 10 largest categories, and when as few as 100 best discriminating words are extracted the BEP curve is already very close to its maximum. Obviously, fancy feature induction techniques would not cause an improvement in categorization results on the datasets like Reuters. Indeed, an extremely sophisticated feature induction method proposed by Raskutti et al. (2001) demonstrated an improvement of less than 1% over the baseline.

The 20 Newsgroups however does not appear to belong to the list of "simple" datasets: Bekkerman et al. (2003) show that every single word of 20NG matters to the classification, and the highest result is achieved while preserving all the words (only stopwords are removed).

So why does such a good method of incorporating bigrams not help to increase performance even on potentially tractable datasets as 20NG? Our main hypothesis is that most of the bigrams are no more informative than just random combinations of unigrams, but their addition increases the variance. Highly discriminative bigrams do exist, but their ratio to "junk" bigrams is low. These "good" bigrams are indeed able to improve the classification results, but their contribution is weak in comparison to what hundreds of thousands of unigrams can contribute.

Our hypothesis is supported by other researchers. Jasper (2003) writes at the DDLBeta Newsgroup: *Bigrams that may rank higher than their components often do not occur with enough frequency to make much of a difference. While measures like Mutual Information do take into account frequency, there is often an implicit tradeoff between frequency and the discriminatory power (e.g., as measured by something like odds ratio). For example, terms like "bill gates" in full do not occur nearly as often as simply "gates" as in "mr gates" or simply "gates". This is even more true in informal text where there are significant typos and misspellings and it is rare to see the same significant bigram used consistently.* Koster and Seutter (2003) write: *Even the most careful term selection cannot overcome the differences in Document Frequency between phrases and words.*

We can conclude that for an unrestricted text categorization task one would probably not expect dramatic effects of using bigrams. However, in domains with severely limited lexicons and high chances of constructing stable phrases the bigrams can be useful. An interesting problem is therefore a categorization application to texts written in programming languages. Applying bigrams in this setup would lead to a significant success.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003.

M. F. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Amita G. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, US, 2001.

J. Diederich, J. L. Kindermann, E. Leopold, and G. Paaß. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1/2):109–123, 2003.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd ed)*. John Wiley & Sons, Inc., New York, 2000.

S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM'98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US, 1998. ACM Press, New York, US.

R. Jasper. On bigrams for text categorization. DDLbeta newsgroup, 2003. owner-ddlbeta@scils.rutgers.edu.

T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In D. H. Fisher, editor, *Proceedings of ICML'97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

C. H. Koster and M. Seutter. Taming wild phrases. In F. Sebastiani, editor, *Proceedings of ECIR'03, 25th European Conference on Information Retrieval*, pages 161–176, Pisa, IT, 2003. Springer Verlag.

D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In N. J. Belkin, P. Ingwersen, and A. M. Pejtersen, editors, *Proceedings of SIGIR'92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK, 1992. ACM Press, New York, US.

H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C.J.C.H. Watkins. Text classification using string kernels. In *Advances in Neural Information Processing Systems (NIPS)*, pages 563–569, 2000.

D. Mladenić and M. Grobelnik. Word sequences as features in text-learning. In *Proceedings of ERK'98, the Seventh Electrotechnical and Computer Science Conference*, pages 145–148, Ljubljana, SL, 1998.

B. Raskutti, H. Ferrá, and A. Kowalczyk. Second order features for maximising text classification performance. In L. De Raedt and P. A. Flach, editors, *Proceedings of ECML'01, 12th European Conference on Machine Learning*, pages 419–430, Freiburg, DE, 2001. Springer Verlag, Heidelberg, DE.

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

R. E. Schapire and Y. Singer. BOOSTEXTER: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

S. Scott and S. Matwin. Feature engineering for text classification. In I. Bratko and S. Dzeroski, editors, *Proceedings of ICML'99, 16th International Conference on Machine Learning*, pages 379–388, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.

C. M. Tan, Y. F. Wang, and C. D. Lee. The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4):529–546, 2002.

V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons Inc., New York, 1998.

S. M. Weiss, C. Apté, F. J. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp. Maximizing text-mining performance. *IEEE Intelligent Systems*, 14(4):63–69, 1999.

D. Zhang and W. S. Lee. Question classification using support vector machines. In J. Callan, G. Cormack, C. Clarke, D. Hawking, and A. Smeaton, editors, *Proceedings of SIGIR'03, 26th ACM International Conference on Research and Development in Information Retrieval*, pages 26–32, Toronto, CA, 2003. ACM Press, New York, US.