Experiments with Sentence Classification

Anthony Khoo, Yuval Marom and David Albrecht
Faculty of Information Technology, Monash University
Clayton, VICTORIA 3800, AUSTRALIA
nthony_ak@yahoo.com, {yuvalm,dwa}@csse.monash.edu.au

Abstract

We present a set of experiments involving sentence classification, addressing issues of representation and feature selection, and we compare our findings with similar results from work on the more general text classification task. The domain of our investigation is an email-based help-desk corpus. Our investigations compare the use of various popular classification algorithms with various popular feature selection methods. The results highlight similarities between sentence and text classification, such as the superiority of Support Vector Machines, as well as differences, such as a lesser extent of the usefulness of features selection on sentence classification, and a detrimental effect of common preprocessing techniques (stop-word removal and lemmatization).

1 Introduction

Classification tasks applied to textual data have been receiving increasing attention due to the explosion in digital presentation and storage of textual information, such as web pages, emails, publications, and discussion forums. The bulk of the research concerns the classification of complete documents, such as spam detection in emails (Drucker et al., 1999), and the classification of news articles (Yang and Pedersen, 1997; Joachims, 1998). These kinds of tasks are widely known as text classification (TC). A text document is best characterized by the words and terms it contains, and consequently the representation of textual data is often of a very high dimensionality. Thus, an important aspect of TC is feature selection (Yang and Pedersen, 1997; Forman, 2003).

There are numerous examples of textual documents whose content conveys communication between multiple parties. In such documents, it may be useful to classify individual sentences that express communicative acts, either to obtain a more meaningful description of the documents, or simply to extract meaningful components, such as action items or opinions. The computational linguistics community devotes considerable research into speech and dialogue acts, and has developed a markup convention for coding both spoken and written language (Core and Allen, 1997, for example). The classifications we use for sentences are inspired by such conventions.

Although there are existing implementations of sentence classification (SC) (Zhou et al., 2004; Wang et al., 2005; McKnight and Srinivasan, 2003), including ones where sentences convey communicative acts (Cohen et al., 2004; Corston-Oliver et al., 2004; Ivanovic, 2005), comparatively little attention has been given to SC in general. In particular, there are no empirical demonstrations of the effect of feature selection in SC tasks, to the best of our knowledge.

This paper presents a study into sentence classification, with particular emphasis on representational issues of extracting features from sentences, and applying feature selection (FS) methods. We experiment with various widely accepted FS methods and classification algorithms, and relate our findings to results from TC reported in the literature. Note that we do not offer any new methods in this paper. Rather, we offer some insight into the characteristics of SC and what distinguishes it from the more general TC, and this insight is driven by empirical findings. We believe that sen-

Proceedings of the 2006 Australasian Language Technology Workshop (ALTW2006), pages 18–25.

Sentence Class	Frequency	Percentage	Sentence Class	Frequency	Percentage
APOLOGY	23	1.5%	SALUTATION	129	8.7%
INSTRUCTION	126	8.5%	SIGNATURE	32	2.2%
INSTRUCTION-ITEM	94	6.3%	SPECIFICATION	41	2.8%
OTHERS	22	1.5%	STATEMENT	423	28.5%
QUESTION	24	1.6%	SUGGESTION	55	3.7%
REQUEST	146	9.8%	THANKING	228	15.3%
RESPONSE-ACK	63	4.2%	URL	80	5.4%

Table 1: Sentence class distribution.

tence classification is emerging as an important task to investigate, due to the increasing interest in detecting intentional units at a sub-document level.

The rest of the paper is organized as follows. In the next section we present our domain of investigation. In Section 3 we discuss the experiments that we carried out, and we conclude the paper in Section 4.

2 Domain

Our corpus consists of 160 email dialogues between customers and operators at Hewlett-Packard's help-desk. These deal with a variety of issues, including requests for technical assistance, inquiries about products, and queries about how to return faulty products or parts. As an initial step in our study, we decided to focus only on the response emails, as they contain well-formed grammatical sentences, as opposed to the customers' emails. In future work we intend to extend our study to include both types of emails. The response emails contain 1486 sentences overall, which we have divided into the classes shown in Table 1. The classes are inspired by the SWBD-DAMSL tag set (Jurafsky et al., 1997), an adaptation of the Dialog Act Markup in Several Layers (DAMSL) annotation scheme (Core and Allen, 1997) for switchboard conversations. For example, RESPONSE-ACK refers to an acknowledgement by the operator of receiving the customer's request: *Your email was submitted to the HP eServices Commercial Support group*; INSTRUCTION-ITEM is similar to INSTRUCTION but appears as part of a list of instructions.

We can see from Table 1 that there is a high distribution skew, where some classes are very small. This means that many of the classes have very few positive examples to learn from. We will see various implications of this high skew in our investigation (Section 3).

When annotating the sentences, problems arose when a sentence was of compound form, which consisted of multiple independent clauses connected by conjunctions, like *"and"*, *"but"*, and *"or"*. For example, the sentence *"Please send us the error message and we will be able to help."*. The two clauses could be labeled as REQUEST and STATEMENT respectively. As our study considered only one tag per sentence, the annotators were asked to consider the most dominant clause to tag the sentence as a whole. Another tricky problem when tagging the sentences dealt with the complex sentences, which contained one independent clause and one or more dependent clauses, for example *"If you see any error message, please forward it to us"*. The first clause is a dependent clause, while the second one is an independent clause. To solve this problem, the annotators were asked to consider only the independent clause to determine which tag to use. Despite these difficulties, we obtained a high inter-tagger agreement, measured with the widely used Kappa statistic (Carletta, 1996) as 0.85. We had three annotators, and we considered only the sentences on which at least two of the annotators agreed. This was the case in all but 21 of the sentences.

3 Experiments

Our experiments involve three classification algorithms, Naive Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM). The evaluation platform is the machine learning software toolkit WEKA (Witten and Frank, 2005). For the SVM, the multi-class task is implemented as a series of binary classification tasks. We employ a stratified 10-fold validation procedure, where the labelled sentences are randomly allocated to training and testing data splits.

A standard measure for classification performance is classification accuracy. However, for datasets with skewed distribution this measure can be misleading, and so instead we have used the F_1 measure, derived from precision and recall (Salton and McGill, 1983), as follows. The precision of a class i is defined as

$$\text{Precision} = \frac{\text{\# sentences correctly classified into class i}}{\text{\# of sentences classified into class i}}$$

and the recall of class i is defined as

$$\text{Recall} = \frac{\text{\# sentences correctly classified into class i}}{\text{\# of sentences that are truly in class i}}$$

and then F_1, the harmonic mean between precision and recall, is defined as

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Once the F_1 measure is calculated for all the classes, we average it to get an overall indication of performance, and also look at the standard deviation as an indication of consistency. The average can be computed in two different methods to reflect the importance of small classes. The first method, called *macro averaging*, gives an equal weight to each class. The second, called *micro averaging*, gives proportional weight according to the proportion of the classes in the dataset. For classes with only a few positive training data, it is generally more difficult to achieve good classification, and their poor performance will have a larger effect on the overall performance when the macro average is used. The choice between the two measures depends on the relative preference that an experimenter places on the smaller classes. Since the classes in our corpus have unbalanced distributions (Table 1) we consider both alternatives and discuss their differences.

3.1 Experiments with representation

Before looking at feature selection, we investigate different techniques for extracting features from sentences. Finding a useful representation for textual data can be very challenging, and the success of classification hinges on this crucial step. Many different techniques have been suggested for text classification, and we have investigated the most common ones.

3.1.1 Representation techniques

Bag-of-words (BoW). Each distinct word in the text corresponds to a feature, and the text is transformed to a vector of N weights ($< w_1, w_2, \ldots, w_N >$), where N is the total number of distinct words in the entire corpus, and w_k is the weight of the k^{th} word in the vector. Information about sentence order, word order and the structure of the text and sentence are discarded. The BoW representation is widely used due to its simplicity and computational efficiency (Cardoso-Cachopo and Oliveira, 2003). There are various methods for setting the weights, for example, solely taking into account the presence of the word, or also considering the frequency of the word. Since we are dealing with sentences that are usually quite short, we do not believe that the frequency of each word conveys any meaning. This is in contrast to typical text classification tasks. We use a binary word-presence representation, indicating whether a word is present or absent from the sentence.[1]

Stop-word removal. Generally, the first step to reduce the feature space is to remove the stop-words (connective words, such as *"of"*, *"the"*, *"in"*). These words are very common words and are conjectured in TC to provide no information to the classifier. Stop-word removal is said to be used in almost all text classification experiments (Scott and Matwin, 1999).

Tokenization. This involves separating any symbols from the numbers or alphabets. For example, the word *"(manual12.txt)"* is separated into five tokens, *"("*, *"manual12"*, *"."*, *"txt"* and *")"*, all considered as features. Without tokenization, a word that is coupled with different symbols may lose its discriminative power because the BoW treats each coupling as a distinct feature. Similarly, the symbols lose any discriminative power.

Lemmatization. The process of mapping words into their base form. For example, the words *"installed"*, *"installs"* and *"installing"* are mapped to *"install"*. This mapping makes the bag-of-words approach treat words of different forms as a single feature, hence reducing the total number of features. This mapping can increase the discriminative power of a word if that word appears in a particular sentence class but in different forms.

Grouping. This involves grouping certain types of words into a single feature. For instance, all words that are valid numbers, like *"1"*, *"444"* and

[1] We have also attempted a bigram representation, however, our results so far are inconclusive and require further investigation.

Representation	Num Features	Measure	NB	DT	SVM
Basic	2710	micro-F_1 ave	0.481	0.791	0.853
		macro-F_1 ave	0.246	0.693	0.790
Best	1622	micro-F_1 ave	0.666	0.829	0.883
		macro-F_1 ave	0.435	0.803	0.866

Table 2: Classification performance using different representations.

"9834", are grouped to represent a single feature. Grouping was also applied to email addresses, phone numbers, URLs, and serial numbers. As opposed to the other techniques mentioned above, grouping is a domain-specific preprocessing step.

3.1.2 Results.

We begin our investigation by looking at the most basic representation, involving a binary BoW without any further processing. The first row in Table 2 shows the results obtained with this basic setup. The second column shows the number of features resulting from this representation, the third column shows the performance measure, and the last three columns show the results for the three classifiers. We see that SVM outperforms the other classifiers on both measures. We also inspected the standard deviations of the macro averages and observed that SVM is the most consistent (0.037 compared to 0.084 and 0.117 for DT and NB, respectively). This means the SVM's performance is most consistent across the different classes. These results are in line with observations reported in the literature on the superiority of SVMs in classification tasks involving text, where the dimensionality is high. We will return to this issue in the next sub-section when we discuss feature selection. We can also see from Table 2 that the micro F_1 average consistently reports a better performance than the macro F_1 average. This is expected due to the existence of very small classes: their performance tends to be poorer, but their influence on the micro average is proportional to their size, as opposed to the macro average which takes equal weights.

We have experimented with different combinations of the various representation techniques mentioned above (Anthony, 2006). The best one turned out to be one that uses tokenization and grouping, and its results are shown in the second row of Table 2. We can see that it results in a significant reduction in the number of features (approximately 40%). Further, it provides a consistent improvement in all performance measures for all classifiers, with the exception of NB, for which the standard deviation is slightly increased. We see that the more significant improvements are reported by the macro F_1 average, which suggests that the smaller classes are particularly benefiting from this representation. For example, serial numbers occur often in SPECIFICATION class. If grouping was not used, serial numbers often appear in different variation, making them distinct from each other. Grouping makes them appear as a single more predictive feature. To test this further, the SPECIFICATION class was examined with and without grouping. Its classification performance improved from 0.64 (no grouping) to 0.8 (with grouping) with SVM as the classifier. An example of the effect of tokenization can be observed for the QUESTION class, which improved largely because of the question mark symbol '?' being detected as a feature after the tokenization process. Notice that there is a similar increase in performance for NB when considering either the micro or macro average. That is, NB has a more general preference to the second representation, and we conjecture that this is due to the fact that it does not deal well with many features, because of the strong assumption it makes about the independence of features.

The surprising results from our investigations are that two of the most common preprocessing techniques, stop-word removal and lemmatization, proved to be harmful to performance. Lemmatization can harm classification when certain classes rely on the raw form of certain words. For example, the INSTRUCTION class often has verbs in imperative form, for example, "install the driver", but these same verbs can appear in a different form in other classes, for example the SUGGESTION sentence "I would try installing the driver", or the QUESTION sentence "Have you installed the driver?". Stop-words can also carry crucial information about the structure of the sentence, for example, "what", "how", and "please". In fact, often the words in our stop-list appeared in the top list of words produced by the feature selection methods. We conclude that unlike text classification tasks, where each item to be classified is rich with textual information, sentence classification involves small textual units that contain valuable cues that are often lost when techniques such as lemmatization and stop-word removal are employed.

3.2 Experiments with feature selection

Since there can be thousands or even tens of thousands of distinct words in the entire email corpus, the feature space can be very large, as we have seen in our baseline experiments (Table 2). This means that the computational load on a classification algorithm can be very high. Thus feature selection (FS) is desirable for reducing this load. However, it has been demonstrated in text classification tasks that FS can in fact improve classification performance as well (Yang and Pedersen, 1997).

We investigate four FS methods that have been shown to be competitive in text classification (Yang and Pedersen, 1997; Forman, 2003; Gabrilovich and Markovitch, 2004), but have not been investigated in sentence classification.

3.2.1 Feature selection algorithms

Chi-squared (χ^2). Measures the lack of statistical independence between a feature and a class (Seki and Mostafa, 2005). If the independence is high, then the feature is considered not predictive for the class. For each word, χ^2 is computed for each class, and the maximum score is taken as the χ^2 statistic for that word.

Information Gain (IG). Measures the entropy when the feature is present versus the entropy when the feature is absent (Forman, 2003). It is quite similar to χ^2 in a sense that it considers the usefulness of a feature not only from its presence, but also from its absence in each class.

Bi-Normal Separation (BNS). This is a relatively new FS method (Forman, 2003). It measures the separation along a Standard Normal Distribution of two thresholds that specify the prevalence rate of the feature in the positive class versus the negative class. It has been shown to be as competitive as χ^2 and IG (Forman, 2003; Gabrilovich and Markovitch, 2004), and superior when there is a large class skew, as there is in our corpus.

Sentence Frequency (SF). This is a baseline FS method, which simply removes features that are infrequent. The sentence frequency of a word is the number of sentences in which the word appears. Thus this method is much cheaper computationally than the others, but has been shown to be as competitive when at least 10% of the words are kept (Yang and Pedersen, 1997).

3.2.2 Results.

We evaluate the various FS methods by inspecting the performance of the classifiers when trained with increasing number of features, where we retain the top features as determined by each FS method. Figure 1(a) shows the results obtained with the χ^2 method, reported using the macro F_1 average, where the error bars correspond to the 95% confidence intervals of these averages. We can see from the figure that SVM and DT are far less sensitive to feature selection than NB. As conjectured in the previous sub-section, NB does not deal well with many features, and indeed we can see here that it performs poorly and inconsistently when many of the features are retained. As we filter out more features, its performance starts to improve and become more consistent. In contrast, the SVM seems to prefer more features: its performance degrades slightly if less than 300 features are retained (although it still outperforms the other classifiers), and levels out when at least 300 features are used. As well as having an overall better performance than the other two classifiers, it also has the smallest variability, indicating a more consistent and robust behaviour. SVMs have been shown in text classification to be more robust to many features (Joachims, 1998).

When comparing the FS methods against each other, it seems their performance is not significantly distinguishable. Figure 1(b) shows the performance of the four methods for the NB classifier. We see that when at least 300 features are retained, the performances of the FS methods are indistinguishable, with IG and χ^2 slightly superior. When less than 300 features are retained, the performance of the SF method deteriorates compared to the others. This means that if we only want very few features to be retained, a frequency-based method is not advisable. This is due to the fact that we have small classes in our corpus, whose cue words are therefore infrequent, and therefore we need to select features more carefully. However, if we can afford to use many features, then this

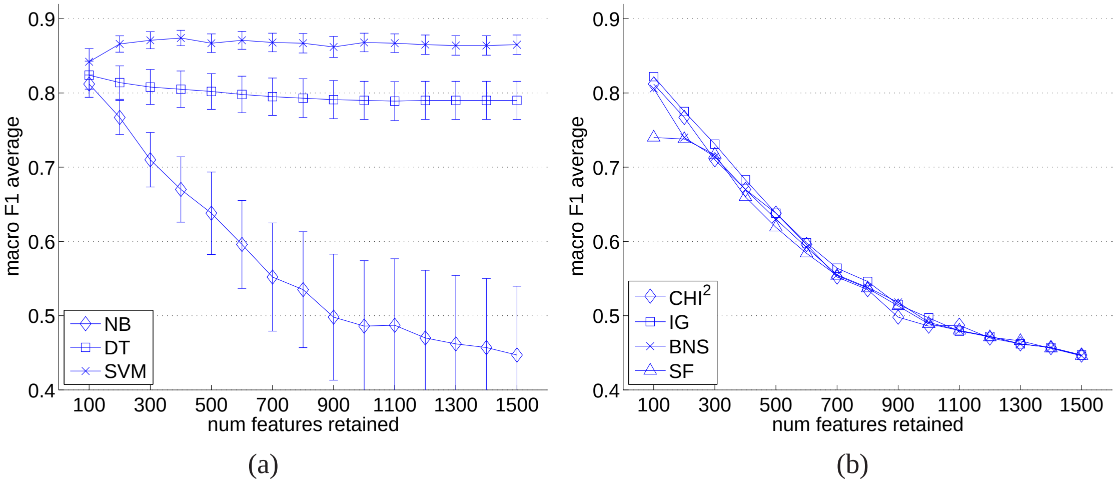

Figure 1: Results from feature selection: (a) the effect of the χ^2 method on different classifiers, (b) the effect of different feature selection methods on the Naive Bayes classifier.

simple method is adequate. We have observed the pattern seen in Figure 1(b) also with the other classifiers, and with the micro F_1 average (Anthony, 2006).

Our observations are in line with those from text classification experiments: the four FS methods perform similarly, except when only a small proportion is retained, when the simple frequency-based method performs worse. However, we expected the BNS method to outperform the others given that we are dealing with classes with a high distributional skew. We offer two explanations for this result. First, the size of our corpus is smaller than the one used in the text classification experiments involving skewed datasets (Forman, 2003) (these experiments use established benchmark datasets consisting of large sets of labelled text documents, but there are no such datasets with labelled sentences). Our smaller corpus therefore results in a substantially fewer number of features (1622 using the our "best" representation in Table 2 compared with approximately 5000 in the text classification experiments). Thus, it is possible that the effect of BNS can only be observed when a more substantial number of features is presented to the selection algorithm. The second explanation is that sentences are less textually rich than documents, with fewer irrelevant and noisy features. They might not rely on feature selection to the extent that text classification tasks do. Indeed, our results show that as long as we retain a small proportion of the features, a simple FS method suffices. Therefore, the effect of BNS cannot be observed.

3.3 Class-by-class analysis

So far we have presented average performances of the classifiers and FS methods. It is also interesting to look at the performance individually for each class. Table 3 shows how well each class was predicted by each classifier, using the macro F_1 average and standard deviation in brackets. The standard deviation was calculated over 10 cross-validation folds. These results are obtained with the "best" representation in Table 2, and with the χ^2 feature selection method retaining the top 300 features.

We can see that a few classes have F_1 of above 0.9, indicating that they were highly predictable. Some of these classes have obvious cue words to distinguish them from other classes. For instance, *"inconvenience"*, *"sorry"*, *"apologize"* and *"apology"* to discriminate APOLOGY class, *"?"* to discriminate QUESTION, *"please"* to discriminate REQUEST and *"thank"* to discriminate THANKING.

It is more interesting to look at the less predictable classes, such as INSTRUCTION, INSTRUCTION-ITEM, SUGGESTION and SPECIFICATION. They are also the sentence classes that are considered more useful to know than some others, like THANKING, SALUTATION and so on. For instance, by knowing which sentences are instructions in the emails, they can be extracted into a to-do list of the email recipient. We have inspected the classification confusion matrix to better understand the less predictable classes. We saw that INSTRUCTION, INSTRUCTION-ITEM,

Sentence Class	**NB**	**DT**	**SVM**
Apology	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Instruction	0.593 (0.109)	0.619 (0.146)	0.675 (0.126)
Instruction-item	0.718 (0.097)	0.582 (0.141)	0.743 (0.127)
Others	0.117 (0.249)	0.411 (0.283)	0.559 (0.282)
Question	0.413 (0.450)	1.000 (0.000)	1.000 (0.000)
Request	0.896 (0.042)	0.930 (0.046)	0.940 (0.047)
Response-ack	0.931 (0.061)	0.902 (0.037)	0.942 (0.057)
Salutation	0.908 (0.029)	0.972 (0.028)	0.981 (0.020)
Signature	0.370 (0.362)	0.960 (0.064)	0.986 (0.045)
Specification	0.672 (0.211)	0.520 (0.218)	0.829 (0.151)
Statement	0.837 (0.042)	0.843 (0.040)	0.880 (0.035)
Suggestion	0.619 (0.206)	0.605 (0.196)	0.673 (0.213)
Thanking	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Url	0.870 (0.071)	0.970 (0.041)	0.988 (0.025)

Table 3: Class-by-class performance

SUGGESTION and STATEMENT were often misclassified as one another. This means that there were not enough distinguishing features to clearly separate these classes. The highest confusion was between INSTRUCTION and STATEMENT, and indeed, sentences of the form *"the driver must be installed before the device will work"* can be interpreted as both an instruction and a general statement. This suggests that the usage of some of these sentence classes may need to be revised.

4 Conclusions

We have presented a set of experiments involving sentence classification. While the successful deployment of classification algorithms for sentences has been demonstrated previously, this kind of classification has received far less attention than the one involving complete documents. In particular, the usefulness of feature selection for sentence classification has not been investigated, to the best of our knowledge.

There are many types of documents where individual sentences carry important information regarding communicative acts between parties. In our experiments this corresponds to email responses to technical help-desk inquiries. However, there are many more examples of such documents, including different kinds of emails (both personal and professional), newsgroup and forum discussions, on-line chat, and instant messaging. Therefore, sentence classification is a useful task that deserves more investigation. In particular, such investigations need to relate results to the more well established ones from text classification experiments, and thus highlight the significant differences between these two tasks.

Our results confirm some observations made from text classification. The SVM classification algorithm generally outperforms other common ones, and is largely insensitive to feature selection. Further, the effect of non-trivial feature selection algorithms is mainly observed when an aggressive selection is required. When a less aggressive selection is acceptable (that is, retaining more features), a simple and computationally cheap frequency-based selection is adequate. Our results also show some important differences between text and sentence classification. Sentences are much smaller than documents, and less rich with textual information. This means that in pruning the feature space one needs to be very careful not to eliminate strong discriminative features, especially when there is a large class distribution skew. We saw that lemmatization and stopword removal proved detrimental, whereas they have been demonstrated to provide a useful dimensionality reduction in text classification. This difference between sentences and documents may also be responsible for obscuring the effect of a particular feature selection method (BNS), which has been demonstrated to outperform others when there is a large distribution skew. We conclude from these observations that while feature selection is useful for reducing the dimensionality of the classification task and even improving the performance of some classifiers, the extent of its usefulness is not as large as in text classification.

Acknowledgments

This research was supported in part by grant LP0347470 from the Australian Research Council and by an endowment from Hewlett-Packard. The authors also thank Hewlett-Packard for the extensive help-desk data, Ingrid Zukerman, Peter

Tischer and anonymous reviewers for their valuable feedback, and Edward Ivanovic for providing a prototype for our tagging software.

References

Anthony. 2006. Sentence classifier for helpdesk emails. Honours Thesis, Clayton School of Information Technology, Monash University.

Ana Cardoso-Cachopo and Arlindo Limede Oliveira. 2003. An empirical comparison of text categorization methods. In *String Processing and Information Retrieval, 10th International Symposium*, pages 183–196, Brazil, October.

Jean Carletta. 1996. Assessing agreement on classification tasks: The Kappa statistic. *Computational Linguistics*, 22(2):249–254.

William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to classify email into "Speech Acts". In *Proceedings of Empirical Methods in Natural Language Processing*, pages 309–316, Barcelona, Spain, July. Association for Computational Linguistics.

Mark G. Core and James F. Allen. 1997. Coding dialogues with the DAMSL annotation scheme. In David Traum, editor, *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35, Menlo Park, California. American Association for Artificial Intelligence.

Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 43–50, Barcelona, Spain, July.

Harris Drucker, Donghui Wu, and Vladimir N. Vapnik. 1999. Support Vector Machines for spam categorization. *IEEE Transactions on Neural Network*, 10(5):1048–1054.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3(7–8):1289–1305. Cambridge, MA, MIT Press.

Evgeniy Gabrilovich and Shaul Markovitch. 2004. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In *Proceedings of the 21st International Conference on Machine Learning*, pages 321–328, Alberta, Canada.

Edward Ivanovic. 2005. Dialogue act tagging for instant messaging chat sessions. In *Proceedings of the ACL Student Research Workshop*, pages 79–84, Ann Arbor, Michigan.

Thorsten Joachims. 1998. Text categorization with Support Vector Machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142.

Daniel Jurafsky, Rebecca Bates, Noah Coccaro, Rachel Martin, Marie Meteer, Klaus Ries, Elizabeth Shriberg, Andreas Stolcke, Paul Taylor, and Carol Van Ess-Dykema. 1997. Automatic detection of discourse structure for speech recognition and understanding. In *Proceedings of the IEEE Workshop on Speech Recognition and Understanding*, pages 88–95, Santa Barbara, CA, December.

Larry McKnight and Padmini Srinivasan. 2003. Categorization of sentence types in medical abstracts. In *Proceedings of the American Medical Informatics Association Annual Symposium*, pages 440–444, Washington D.C.

Gerard Salton and Michael J. McGill. 1983. *An Introduction to Modern Information Retrieval*. McGraw-Hill.

Sam Scott and Stan Matwin. 1999. Feature engineering for text classification. In *Proceedings of ICML-99: The 16th International Conference on Machine Learning*, pages 379–388, Slovenia.

Kazuhiro Seki and Javed Mostafa. 2005. An application of text categorization methods to gene ontology annotation. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 138–145, Salvador, Brazil.

Chao Wang, Jie Lu, and Guangquan Zhang. 2005. A semantic classification approach for online product reviews. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 276–279, Compiegne, France.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Fransisco, 2nd edition.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97: The 14th International Conference on Machine Learning*, pages 412–420, Nashville, US.

Liang Zhou, Miruna Ticrea, and Eduard Hovy. 2004. Multi-document biography summarization. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 434–441, Barcelona, Spain.