Mining E-mail Authorship[*]

Olivier de Vel
Information Technology Division
Defence Science and Technology Organisation
P.O. Box 1500
Salisbury 5108, Australia
olivier.devel@dsto.defence.gov.au

ABSTRACT

In this paper we report an investigation into the learning of authorship identification or categorisation for the case of e-mail documents. We use various e-mail document features such as structural characteristics and linguistic evidence together with the Support Vector Machine as the learning algorithm. Experiments on a number of e-mail documents give promising results with some e-mail document features and author categories giving better categorisation performance results.

1. INTRODUCTION

With the rapid growth in computer technology, many industries and governments have become dependent on the use of electronic mail (e-mail) as an expedient and economical form of communication over the Internet and intranets. E-mail is used in many different situations as, for example, in the exchange and broadcasting of messages, documents and for conducting electronic commerce. Unfortunately, it can also be misused for the distribution of unsolicited and/or inappropriate messages and documents. Examples of misuse include the distribution of unsolicited junk mail (commonly referred to as "spamming"), unauthorised conveyancing of sensitive information, mailing of offensive or threatening material etc.. In some misuse cases the sender will attempt to hide his/her true identity in order to avoid detection. For example, the sender's address can be forged or routed through an anonymous mail server, or the e-mail's contents and header information may have been modified in an attempt to hide the true identity of the sender. The ability to provide empirical evidence and identify the original author of e-mail misuse is an important factor in the successful prosecution of an offending user.

As a result of this growing e-mail misuse problem, efficient automated methods for analysing the content of e-mail messages and identifying or categorising the authors of these messages are becoming imperative. The principal objectives are to classify an ensemble of e-mails as belonging to a particular author and, if possible, obtain a set of characteristics that remain relatively constant for a large number of e-mails written by the author. Identifying such characteristics highlights the inherent difficulties facing authorship categorisation since we expect that the writing characteristics of an author to evolve in time and change in different contexts. For example, the composition of formal e-mails will differ from informal ones (changes in vocabulary etc.). Even in the context of informal e-mails there could be several composition styles (e.g., for personal relations and for work relations). However, humans are creatures of habit and have certain personal traits which tend to persist. All humans have unique (or near-unique) patterns of behaviour, biometric attributes, and so on. We therefore conjecture that certain characteristics pertaining to language, composition and writing, such as particular syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage (eg, converting the letter "f" to "ph", or the excessive use of digits and/or upper-case letters), stylistic and sub-stylistic features will remain relatively constant. The identification and learning of these characteristics are the principal challenges in authorship categorisation. Another difficulty with authorship categorisation is whether it can be performed with a sufficiently high accuracy for the results to be presented as legal argument.

Authorship categorisation can be effected using various approaches. Firstly, the simplest method is to use domain experts to identify new e-mail documents and allocate them to well-defined author categories. This can be time-consuming and expensive and, perhaps most limiting, provides no continuous measure of the degree of confidence with which the allocation was made. Secondly, the domain expert can establish a set of fixed rules which can be used to classify new e-mail documents. Unfortunately, in many cases, the rule-set can be large and unwieldy, typically difficult to update, and unable to adapt to changes in document content or author characteristics. Finally, categorisation can be undertaken automatically by inductively learning the classifiers from training example documents. This approach should, hopefully, generalise well to new, unseen e-mail documents and has the advantage that it should able to adapt to a measure of drift in the characteristics of authors and create a more accurate profile of each author.

[*] *KDD-2000 Workshop on Text Mining*, August 20, 2000, Boston.

A closely related area of authorship categorisation is text categorisation, which attempts to categorise a set of text documents based on its content-type. Text categorisation provides support for a wide variety of activities in information mining and information management. It has found applications in document filtering and can be used to support document retrieval by generating the categories required in document retrieval. Many methods that automatically learn rules have been proposed for text categorisation. Most of these techniques are based on the "bag–of–words" or word vector space representation [16] in which each feature in the text document corresponds to a single word and then use a learning approach such as decision trees [1], neural networks [12], Bayesian probabilistic approaches [11][22], or support vector machines [8] to classify the text document. Work in e-mail text classification has also been undertaken by some researchers in the context of automated e-mail document filtering and filing. Cohen [3] learns rule sets based on a small number of keywords in the e-mail. Sahami *et al* [14] focuses on the more specific problem of filtering junk e-mail using a Naive Bayesian classifier and incorporating domain knowledge using manually constructed domain-specific attributes such as phrasal features and various non-textual features.

In this paper we investigate methods for learning authorship classifiers from e-mail documents. We incorporate various document features such as structural characteristics and linguistic evidence in the learning algorithm. We first introduce the field of authorship categorisation in Section 2 and briefly outline the Support Vector Machines learning algorithm in Section 3. We present the experimental methodology and database of e-mail documents used in the experiments in Section 4. Validation of the method is then undertaken by presenting results of categorisation performance in Section 5. Finally, we conclude with some general observations and present future directions for the work in Section 6.

2. AUTHORSHIP CATEGORISATION

Formally, authorship categorisation is the task of determining the author of a piece of work. In particular, we are interested in categorising textual work usually based on other text samples produced by the same author. We assume that only one author is responsible for producing the text – contributions by, or text modified by, multiple authors are not considered here.

Authorship categorisation is a subset of the more general problem called "authorship analysis" [7]. Authorship analysis includes other distinct fields such as author characterisation and similarity detection. Authorship characterisation determines the author profile or characteristics of the author that produced a piece of work. Example characteristics include educational and cultural backgrounds. Similarity detection calculates the degree of similarity between two or more pieces of work without necessarily identifying the authors. Similarity is used extensively in the context of plagiarism detection which involves the complete or partial replication of a piece of work with or without permission of the original author. We note, however, that authorship categorisation and author characterisation are different from plagiarism detection. Plagiarism detection attempts to detect the similarity between two substantially different pieces of work but is unable to determine if they were produced by the same author.

Authorship analysis has been used in a small but diverse number of application areas. Examples include identifying authors in literature, in program code, and in forensic analysis for criminal cases. We briefly outline the work undertaken in each one of these areas.

Perhaps the most extensive and comprehensive application of authorship analysis is in literature and in published articles. Several studies attempting to resolve Shakespeare's works date back many years (see, for example, [4]). In one of these studies, attempts were made to show that Shakespeare was a hoax and that the real author was Edward de Vere, the Earl of Oxford [6]. Specific author features such as unusual diction, frequency of certain words, choice of rhymes, and habits of hyphenation have been used as tests for author attribution. An important kind of evidence that can be used to establish authorship is that of linguistic evidence, that is, distinctive language habits that are sufficiently unique to identify the author. It is thought that such linguistic evidence is generated dynamically and subconsciously when language is created, similar to the case of the generation of utterances during speech composition and production [4]. Language patterns or sub-stylistic features are generated beyond an author's conscious control. An example of such features is short, all-purpose words (called "function words") such as "the", "if", "to" etc. whose frequency or relative frequency of usage is unaffected by the subject matter. Therefore, a combination of sub-stylistic features may be sufficient to uniquely authenticate an author. Bosch *et al* used a small set of function words for the the classification of two authors involved in the authorship of the Federalist Papers articles [2].

Program code authorship has been researched by some workers in the context of software theft and plagiarism, software author tracking and intrusion detection. For example, software author tracking enables the identification of the author of a particular code fragment from a large set of programmers working on a software project. This can be useful for the purpose of identifying authors for software upgrade and maintenance. The authorship of a computer virus or trojan horse can be identified in a similar manner [17]. By examining peculiar characteristics or metrics of programming style it is possible to identify the author of a section of program code [13], in a similar way that linguistic evidence can be used for categorising the authors of free text. Program metrics such as typographical characteristics (eg, use of lower and upper case characters, multiplicity of program statements per line, etc.), stylistic metrics (eg, length of variable names, preference for `while` or `for` loops, etc.), programming structure metrics (eg, placement of comments, use of debugging symbols, etc.) have been employed [10][9][15].

The forensic analysis of text, or forensic linguistics, attempts to match text to authors for the purpose of a criminal investigation. The textual analysis of the Unabomber manifesto is a well-known example of the use of forensic linguistics. In this particular case, the manifesto and the suspect bomber used a set of similar characteristics, such as a distinctive vocabulary, irregular hyphenations etc. [4].

E-mail documents have several characteristics which make authorship categorisation challenging compared with longer, formal text documents such as literary works or published articles (such as the Federalist Papers). Firstly, the composition style used in formulating an e-mail document is often different from normal text documents written by the same author. That is, an author profile derived from normal text documents (eg, publications) may not necessarily be the same as that obtained from an e-mail document. For example, e-mail documents are generally brief and to the point, can be punctuated with a larger number of grammatical errors etc. Indeed, the authoring composition style attributed to e-mails is often a combination of formal writing and speech transcript. Secondly, the author's composition style used in e-mails can vary depending upon the recipient and can evolve quite rapidly over time. Finally, e-mail documents have generally few sentences/paragraphs, thus making profiling based on traditional text document analysis techniques, such as the "bag–of–words" representation (eg, when using the Naive Bayes approach), more difficult. However, as stated previously, certain characteristics such as particular syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage, stylistic and sub-stylistic features will remain relatively constant for a given author. This provides the major motivation for the choice of attributes/features for the authorship categorisation of e-mails, as we shall discuss in Section 4.

3. SUPPORT VECTOR MACHINE CLASSIFIER

The fundamental concepts of Support Vector Machines (SVM) were developed by Vapnik [19]. The SVMs' concept is based on the idea of structural risk minimisation which minimises the generalisation error (i.e. true error on unseen examples) which is bounded by the sum of the training set error and a term which depends on the Vapnik-Chervonenkis (VC) dimension of the classifier and on the number of training examples. The use of a structural risk minimisation performance measure is in contrast with the empirical risk minimisation approach used by conventional classifiers. Conventional classifiers attempt to minimise the training set error which does not necessarily achieve a minimum generalisation error. Therefore, SVMs have theoretically a greater ability to generalise. For further reading, see [19].

Unlike many other learning algorithms, the number of free parameters used in the SVM depends on the margin that separates the data and does not depend on the number of input features. Thus the SVM does not require a reduction in the number of features in order to avoid the problem of over-fitting. This property is clearly an advantage in the context of high-dimensional applications, such as text document and authorship categorisation, as long as the data vectors are separable with a wide margin. Unfortunately, SVMs require the implementation of optimisation algorithms for the minimisation procedure which can be computationally expensive. A few researchers have applied SVMs to the problem of text document categorisation concluding that, in most cases, SVMs outperform conventional classifiers [22][8]. Drucker *et al* used SVMs for classifying e-mail text as spam or non-spam and compared it to boosting decision trees, Ripper and Rochio classification algorithms [5]. Bosch *et al* used a separating hyperplane based on a similar idea to that of a linearly separable SVM for determining the authorship of two authors of the formal articles published within the set of the Federalist Papers [2].

4. E-MAIL DOCUMENT CORPUS, ATTRIBUTES AND EXPERIMENTAL METHODOLOGY

The corpus of e-mail documents used in the experimental evaluation of authorship categorisation contained a total of 274 documents and five authors. It is not necessarily a trivial task to obtain a database of "useful" e-mail documents as a variety of triage issues need to be resolved, particularly the problem of privacy. Also, many e-mail documents are brief and/or have limited interest for identifying any author traits. In our database we have included private e-mails and have removed any e-mails with topics that contribute minimally to author traits (e.g., one-line e-mails dealing with technical issues).

The body of each e-mail document was pre-processed to remove (if present) any salutations, reply text and signatures. However, the existence, location and type of some of these are retained as inputs to the categoriser (see below). Attachments are excluded, though the e-mail body itself is used. A summary of the global e-mail document corpus statistics is shown in Table 1.

A number of attributes including structural features, patterns of vocabulary usage, stylistic and sub-stylistic features were extracted from each e-mail document. These are listed in Table 2.

We clarify how we derive some of the attributes shown in Table 2. Firstly, to calculate attribute index 13, the number of short words in each e-mail document (eg, "all", "both", "some" etc.) is obtained from a set of pre-defined short words with a maximum cardinality equal to 74. The total number of vocabulary words refers to the cardinality of the set of (distinct) words used in the e-mail body. Secondly, for attribute index 14, the number of words used once in the e-mail text document and which are 3 or more characters in length is extracted. Thirdly, the reply status (attribute index 12) indicates whether the e-mail contains a reply text. A reply text can be placed in any position in the e-mail document and each line is usually prefixed with a special character (e.g., ">"). Finally, the frequency of 24 all-purpose function words ("a", "all", "also", . . . , "to", "with") is obtained and used individually as separate features (attribute indices 15 to 38). The set of function words (which is a subset of the list short words) is used as a set of sub-stylistic features.

As Support Vector Machines only compute two-way categorisation, Q two-way classification models were generated, where Q is the number of author categories ($Q = 5$ for our e-mail document corpus), and each SVM categorisation was applied Q times. This produced Q two-way confusion matrices. The SVM classifier was trained on 192 documents (70% of the e-mail document set) and tested on the remaining (unseen) 82 documents.

To evaluate the categorisation performance on the e-mail document corpus, we use the accuracy, recall (R) and precision (P) performance measures commonly employed in the

Author Category	*Number of E-mail Documents*	*Document Size (Number of words)*	
		Minimum	*Maximum*
A	87	3	323
B	51	0	112
C	73	19	440
D	45	3	680
E	18	30	298

Table 1: Summary statistics of the e-mail document corpus used in the experiment.

Attribute	*Attribute Type*
1	Total number of words
2	Average length of words (number of characters)
3	Number of sentences
4	Average length of sentences
5	Number of lines
6	Number of blank lines
7	Average length of lines (number of characters)
8	Number of characters
9	Number of upper-case characters
10	Number of digits
11	Number of tabs
12	Reply status
13	Ratio of short words to total number of vocabulary words
14	Ratio of words used once to total number of vocabulary words
15–38	Function word attributes

Table 2: E-mail document body attributes.

		Predicted	*Category*
		+ve	*-ve*
Actual Category	*+ve*	**A**	**B**
	-ve	**C**	**D**

Table 3: Two-way confusion matrix, with predicted (assigned by the classifier) and actual (true) category numbers.

information retrieval and text categorisation literature (for a discussion of these measures see, for example, [20]). Given the two-way confusion matrix (Table 3) we can define precision, recall and accuracy as follows.

Precision (P) is defined as:

$$\begin{aligned} \mathrm{P} &= \frac{\text{Number of correct category assignments}}{\text{Total number of category assignments}} && (1) \\ &= \frac{\mathbf{A}}{\mathbf{A}+\mathbf{C}} && (2) \end{aligned}$$

recall (R) as:

$$\begin{aligned} \mathrm{R} &= \frac{\text{Number of correct category assignments}}{\text{Total number of correct category assignments}} && (3) \\ &= \frac{\mathbf{A}}{\mathbf{A}+\mathbf{B}} && (4) \end{aligned}$$

and accuracy (the more traditional measure of categorisation or classification performance) as:

$$\text{Accuracy} = 1 - \text{Error} = \frac{\mathbf{A}+\mathbf{D}}{\mathbf{A}+\mathbf{B}+\mathbf{C}+\mathbf{D}}$$

Accuracy suffers from the problem that, for categories with small sample document numbers, **A** and **D** can be small and the accuracy statistic will not be very sensitive to variations in **A** and/or **D**.

The precision statistic should be considered in conjunction with the recall performance measure when analysing classifier performance as there exists a trade-off between precision and recall. If only one statistic is used, some meaningless conclusions can be drawn (e.g., we can conclude that allocating every author to the positive class will provide a 100% recall but possibly give an extremely low and unrealistic value for the precision). To circumvent this, a combined statistic is normally used such as the F_β-measure:

$$F_\beta = \frac{(1+\beta^2)RP}{(R+\beta^2 P)}$$

where β is a parameter which determines the relative weight of the P and R statistics. Generally, we set $\beta = 1$ (i.e., P and R have equal weight):

$$F_1 = \frac{2RP}{(R+P)}$$

To obtain an overall performance figure over all binary categorisation tasks, a macro-averaged statistic is calculated [21]. Here, N_{AC} per-category confusion matrices (where N_{AC} is the total number of author categories) are computed and then averaged over all categories to produce the macro-averaged statistic, $F_1^{(M)}$:

$$F_1^{(M)} = \frac{\sum_{i=1}^{N_{AC}} F_{1,AC_i}}{N_{AC}}$$

where F_{1,AC_i} is the per-category F_1 statistic for category AC_i $(i = 1, 2, , \ldots N_{AC})$:

$$F_{1,AC_i} = \frac{2R_{AC_i}P_{AC_i}}{(R_{AC_i} + P_{AC_i})}$$

However, the macro-averaged statistic penalises author categories with few e-mail documents. To compensate for document frequency, the per-author category confusion matrices are "inversely" weighted by the author category frequency prior to summing each component. We call this the *weighted macro-averaged* F_1 *statistic*, $\overline{F}_1^{(M)}$.

$$\overline{F}_1^{(M)} = \frac{\sum_{i=1}^{N_{AC}}(1-w_{AC_i})F_{1,AC_i}}{\sum_{i=1}^{N_{AC}}(1-w_{AC_i})} = \frac{\sum_{i=1}^{N_{AC}}(1-w_{AC_i})F_{1,AC_i}}{N_{AC}-1}$$

where w_{AC_i} is the document frequency weight:

$$w_{AC_i} = \frac{N_{AC_i}}{\sum_{i=1}^{N_{AC}} N_{AC_i}}$$

and N_{AC_i} is the number of documents in author category AC_i.

The classifier used in the experiments was the Support Vector Machines classifier, *SVMlight* [18], developed by T. Joachims from the University of Dortmund. *SVMlight* is an implementation of Vapnik's Support Vector Machines. It scales well to a large number of sparse instance vectors as well as efficiently handling a large number of support vectors. In our experiments we used the linear kernel function and the "LOQO" optimiser for maximising the margin.

5. RESULTS AND DISCUSSION

We report our results presenting the per-category F_1 and macro-averaged F_1 statistics for the Support Vector Machines (SVM) classifier. The results are shown in Table 4. These were obtained by averaging over ten randomised classification runs (i.e., the classifier was run ten times, each time using randomly-sampled training and test sets by the process of resubstitution) and computing the average values for R_{AC_i}, P_{AC_i}.

The values computed for $F_1^{(M)}$ and $\overline{F}_1^{(M)}$ were 51.3% and 55.5%, respectively. As observed in Table 4, the per-category performance results for the R, P measures and F_1 statistic are, as we expect, lower than the accuracy results. In particular, author `E` has a relatively high accuracy with, however, a very low F_1 statistic, reflecting the combination of

Author Category, AC_i	Accuracy	R_{AC_i}	P_{AC_i}	F_{1,AC_i}
A	71.7	49.6	51.7	50.6
B	84.8	63.9	58.9	61.3
C	78.7	61.4	64.3	62.8
D	85.7	68.8	52.4	59.5
E	84.0	32.2	17.3	22.5

Table 4: Accuracy and per-category R_{AC_i}, P_{AC_i} and F_{1,AC_i} categorisation performance results (in %) for the different author categories.

poor precision and recall values for this class. This author category is characterised by having very few discriminating features. Better per-category R_{AC_i} and P_{AC_i} performance results can be observed for authors B, C and D. These author categories have certain characteristic features which enables better discrimination. Discriminating features for the different authors include the existence of tabs, number of lines in the e-mail documents and the function word attributes. Other features such as the "ratio of short words to total number of vocabulary words" or the "ratio of words used once to total number of vocabulary words" did not provide for any discrimination between the author categories. We suspect that other statistics of these features (e.g., the variance) may be more appropriate.

We also obtained values for the features for all e-mail documents for each author category to identify some of the ensemble author characteristics (see Table 5 for a subset of the features). We observe some significant in-between author category differences with some of the features, particularly with some of the function word attributes. This indicates that some features could potentially be strong discriminators of the author categories (e.g., ratio of the frequencies of the words "a" and "all") when larger document populations are considered.

6. CONCLUSIONS

In this paper we have investigated the learning of authorship categories from e-mail documents. We used various e-mail document features such as structural characteristics and linguistic patterns together with a Support Vector Machine learning algorithm. Experiments on a reduced number of documents gave promising results, though some author categories produced better categorisation performance results than other categories.

There are several limitations with the current approach. Firstly, the fact that some authors have a better categorisation performance than other authors indicates that more identifiable author traits need to be obtained. Secondly, the combination of features, particularly for features such as relative function word frequencies, should be considered. Thirdly, feature selection prior to categorisation should be undertaken to remove features that do not contribute to the categorisation performance. Finally, the number of author categories considered in our experiments at the moment is quite small. We are currently attempting to overcome these limitations.

7. REFERENCES

[1] C. Apte, F. Damerau, and S. Weiss. "Text mining with decision rules and decision trees". In *Workshop on Learning from text and the Web, Conference on Automated Learning and Discovery*, 1998.

[2] R. Bosch and J. Smith. "Separating hyperplanes and the authorship of the disputed federalist papers". *American Mathematical Monthly*, 105(7):601–608, 1998.

[3] W. Cohen. "Learning rules that classify e-mail". In *Proc. Machine Learning in Information Access: AAAI Spring Symposium (SS-96-05)*, pages 18–25, 1996.

[4] C. Crain. "The Bard's fingerprints". *Lingua Franca*, pages 29–39, 1998.

[5] H. Druker, D. Wu, and V. Vapnik. "Support vector machines for spam categorisation". *IEEE Trans. on Neural Networks*, 10:1048–1054, 1999.

[6] W. Elliot and R. Valenza. "Was the Earl of Oxford the true Shakespeare?". *Notes and Queries*, 38:501–506, 1991.

[7] A. Gray, P. Sallis, and S. MacDonell. "Software Forensics: Extending Authorship Analysis Techniques to Computer Programs". In *Proc. 3rd Biannual Conf. Int. Assoc. of Forensic Linguists (IAFL'97)*, pages 1–8, 1997.

[8] T. Joachims. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features". In *Proc. European Conf. Machine Learning (ECML'98)*, pages 137–142, 1998.

[9] I. Krsul. "Authorship analysis: Identifying the author of a program". Technical report, Department of Computer Science, Purdue University, 1994. Technical Report CSD-TR-94-030.

[10] I. Krsul and E. Spafford. "Authorship analysis: Identifying the author of a program". *Computers and Security*, 16:248–259, 1997.

[11] T. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

	Author Category (AC_i)				
Attribute Type	A	B	C	D	E
Feature 13	0.048	0.115	0.047	0.043	0.099
Feature 14	0.57	0.657	0.494	0.552	0.586
Ratio of features 15 and 16	4.30	37.0	12.5	5.07	7.0
Ratio of features 15 and 17	4.71	37.0	8.3	13.8	4.0
Ratio of features 34 and 36	1.66	1.51	1.73	1.54	11.8

Table 5: Summary statistics for all e-mail documents for each author category and for selected features/attributes. *Attribute Type* **is defined in Table 2, where features 15, 16, 17, 34 and 36 correspond to words "a", "all", "also", "the" and "to" respectively.**

[12] H. Ng, W. Goh, and K. Low. "Feature selection, Perceptron Learning, and a Usability Case Study for Text Categorization". In *Proc. 20th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR97)*, pages 67–73, 1997.

[13] P. Oman and C. Cook. "Programming style authorship analysis". In *Proc. 17th Annual ACM Computer Science Conference*, pages 320–326, 1989.

[14] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. "A Bayesian Approach to Filtering Junk E-Mail". In *Learning for Text Categorization Workshop: 15th National Conf. on AI. AAAI Technical Report WS-98-05*, pages 55–62, 1998.

[15] P. Sallis, S. MacDonell, G. MacLennan, A. Gray, and R. Kilgour. "Identified: Software Authorship Analysis with Case-Based Reasoning". In *Proc. Addendum Session Int. Conf. Neural Info. Processing and Intelligent Info. Systems*, pages 53–56, 1997.

[16] G. Salton and M. McGill. *Introduction to Modern Information Filtering.* McGraw-Hill, New York, 1983.

[17] E. Spafford and S. Weeber. "Software forensics: tracking code to its authors". *Computers and Security*, 12:585–595, 1993.

[18] University of Dortmund. *Support Vector Machine, SVMLight.* http://www-ai.cs.uni-dortmund.de/FORSCHUNG/VERFAHREN/ /SVM_LIGHT/svm_light.eng.html.

[19] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[20] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, San Francisco, 2000.

[21] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1(1/2):67–88, 1999.

[22] Y. Yang and X. Liu. "A Re-examination of Text Categorisation Methods". In *Proc. 22nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR99)*, pages 67–73, 1999.