The Form is the Substance: Classification of Genres in Text

Nigel Dewdney
U.S. Department of Defense
njdewdn@afterlife.ncsc.mil

Carol VanEss−Dykema
U.S. Department of Defense
cjvanes@afterlife.ncsc.mil

Richard MacMillan
MITRE Corp.
macmilla@mitre.org

Abstract

Categorization of text in IR has traditionally focused on topic. As use of the Internet and e−mail increases, categorization has become a key area of research as users demand methods of prioritizing documents. This work investigates text classification by format style, i.e. "genre", and demonstrates, by complementing topic classification, that it can significantly improve retrieval of information. The paper compares use of presentation features to word features, and the combination thereof, using Naïve Bayes, C4.5 and SVM classifiers. Results show use of combined feature sets with SVM yields 92% classification accuracy in sorting seven genres.

1 Introduction

This paper firstly defines genre, explains the rationale for automatic genre classification, and reviews some previously published work relevant to this problem. It describes the features chosen to be extracted from documents for input to a classification system. The paper next describes data used, experiments carried out, and the results obtained. Finally the paper discusses the results and suggests ways for the research to progress.

1.1 Defining Genre

The genre of a document is defined here as a label which denotes a set of conventions in the way in which information is presented. These conventions cover both formatting and style of language used. Examples of genres include "Newswire", "Classified Advertisements", and "Radio Broadcast News Transcript". The format of the text and the style of language used within a genre is usually consistent even though the topics of different documents may vary greatly. Note that text classifications such as "Sport" or "Politics" are not considered as genres here since these are broad topic areas.

1.2 Why Genre?

Many people are experiencing the growth in the volume of electronic text: Sources include news services, online journals, and e−mail. Few people have time to scan every text source of potential interest to them and not all sources are of equal interest to everyone.

The continuing expansion of the Internet makes it increasingly hard to find information relevant to the user’s needs. Search engines go some way to solving this problem, but often the results are dominated by hits that do not match the user’s requirements. Many search engines, such as Yahoo, provide a hierarchical classification of sites which organize web sites by the type of information and/or services they provide. However, the hierarchies only cover a fraction of the Web and are largely hand built. An automatic method of building site categories, in conjunction with topic identification, would speed the hierarchy construction and allow more frequent updates.

The authors believe that a classifier can be trained to distinguish different document classes, or genres, such as advertisements or jokes from news stories, for example. It can be trained to help identify the proportion of user−relevant texts, which can often be very small. If a user is searching for "stories of the god Jupiter" then news articles and scientific papers would less likely be of interest than classical fiction. Note that sorting by genre differs from "Information filtering" as the latter carries out text selection based on content (Oard, 1994). The belief is that for users who often find irrelevant texts classified as relevant, and thus

Query	**"house"**	**"house" & ("sale" | "rent")**	**"house" + genre "ads"**	**G(ads) "house" & ("sale" | "rent")**
Recall	80%	53%	75%	47%
Precision	2%	19%	43%	35%
F1 Metric	4	28	55	40

Table 1: Comparison of example queries with and without use of genre tag in a marked corpus

making finding the relevant texts time consuming, a genre filter can function with a high degree of accuracy and lessen misclassification of text.

The junk e−mail problem is also well known. The variety in types of e−mail is just as large as that found in regular mail: from personal communications through to unsolicited junk mail. The user must wade through the e−mail queue or risk insufficient storage space. Users could hand−craft rules to filter junk mail but these require constant refinement (Sahami, 1997). "Spam" filters are becoming popular though these just seek to separate unsolicited and solicited mail. A genre sorter could serve as a prioritization tool for helping the user to organize the e−mail in his or her queue beyond just "solicited" and "unsolicited" with a higher degree of accuracy.

The following example illustrates the potential of genre identification by showing improvement in topic based information retrieval. Note: This is a single example, so figures have no significance.

Consider looking for information about houses currently for sale. In a 9,705 document corpus, detailed later, there are 20 such documents. (Each document is tagged with one of seven genres.) If one attempts to retrieve these documents with the naïve query "house" one finds 693 documents, 16 of which contain the information required. If one runs the same query but specifies the genre "ads", then 35 documents are retrieved, 15 of which contain relevant information. A more specific boolean query works better than the naïve keyword (unsurprisingly), but adding the genre specification still improves the result.

Four documents offered houses for sale but did not mention "house", one document offered a house for sale but was not tagged "ads".

The goal, then, is to develop a system that can automatically categorize text into genres, where genres are defined as classes whose members follow certain conventions of form.

1.3 Related Work

The idea of genre classification is not new. Kessler et. al. ('97), for example, proposed genre "facets". They note that genre does not become salient until faced with large hetergoneous search spaces, but only report an experiment using insignificant quantities of data. Stamatatos et. al. (2000) used discriminant analysis on the frequencies of commonly occurring words. They also improved results by including frequencies of eight punctuation marks. Four identified genres from the Wall Street Journal formed the corpus, but only 40 samples per genre were used. Both works dispensed with the more complex features proposed by Karlgren and Cutting (1994) which showed promising results. They report 27% error in distinguishing four genres in 500 samples from the Brown corpus.

Illouz et. al. (2000) report successful use of coarse level Part−of−Speech features in distinguishing section types from Le Monde. Their work also showed that fine grain Part−of−Speech distinctions did not make for good features in genre classification.

Sahami et. al. (1997) report on methods for automating the construction of e−mail filters, sorting into two categories: "junk" and "non−junk". Using probabilistic learning methods in conjunction with a differential misclassification cost to produce filters, they used selected words as their feature set. They also augmented these by some domain specific features, such as number of recipients and time of posting. This yielded improved results, but no results on use of domain specific features alone are presented.

Pannu and Sycara (1996) describe a reusable agent that learns a model of the user's research interests for filtering Conference Announcements and Requests for Proposals from the Web. They used Information Retrieval (IR) and Artificial Neural Network (ANN) techniques. The features used were a fixed number of keywords selected from terms sorted by TF*IDF (term frequency x inverse document frequency). They report results of 85% and 70% accuracy in classifying a document’s relevance using IR techniques and ANN techniques respectively.

May (1997) describes a system that automatically classifies e−mail messages in the HUMANIST electronic discussion group into one of four categories. He based the classification of a message on string matching using predefined phrase sets, selected manually, for each message type. Results within categories were mixed but overall May reports a 45.9% accuracy in tagging.

Cohen (1996) reports on two methods for classification in sorting personal e−mail messages. He uses a TF−IDF weighting method, and a new method for learning sets of "keyword−spotting rules" based on the RIPPER rule learning algorithm. Both methods obtained error rates of below 5%. However, only extraction of one category from the stream is considered.

2 Document features

This work investigates the use of two different feature sets: a set based on words (traditional), and a set of features that reflect the way in which the text is presented.

2.1 Word features

Traditionally the document features used have been the words therein. Text classification and clustering experiments have focussed on "bag−of−words" techniques: the features used are individual words usually weighted by some scheme such as TF*IDF. Feature selection techniques, such as thresholding term weights, are used to reduce feature vector size. The Information Gain algorithm is employed for this work.

Information Gain is frequently employed in machine learning to reduce computational complexity. For document classification, it measures the number of bits of information gained, with respect to deciding the class to which a document belongs, by each word’s frequency of occurrence in the document (Mitchell, 1997. Yang, 1997). Maximum−entropy characterization of information gain [2] is used for this paper as formulated for document classification by Yang et. al. (1997). Yang’s formulation is appropriate because it treats features as objects whose values can be measured numerically, rather than as a finite set of predefined discrete values:

Let $\{g_i\}_{\{i=1..m\}}$ be the set of target genres, let w be a word, and let k be a token (i.e., an occurrence of a word) in the document corpus. Denote by $P(w)$ the probability that a randomly chosen token k is an occurrence of the word w. Let g(k) be the genre of the document in which k occurs, $P(g_i)$ be the probability that a randomly chosen k has $g(k) = g_i$, and $P(g_i \mid w)$ be the conditional probability that a randomly chosen k has $g(k) = g_i$, given that k is an occurrence of the word w. Denote by $|S|$ the number of members in any set S. An overstrike denotes the negation of a condition. The information gain of word w is defined to be:

$$
\begin{aligned}
IG(w) = \Sigma_{i=1..m} [-(& P(g_i) \log P(g_i) + P(\overline{g}_i) \log P(\overline{g}_i)) \\
& + P(w) P(g_i \mid w) \log P(g_i \mid w) \\
& + P(\overline{w}) P(g_i \mid \overline{w}) \log P(g_i \mid \overline{w}) \\
& + P(w) P(\overline{g}_i \mid w) \log P(\overline{g}_i \mid w) \\
& + P(\overline{w}) P(\overline{g}_i \mid \overline{w}) \log P(\overline{g}_i \mid \overline{w})] \quad \text{(eqn 1)}
\end{aligned}
$$

where:

$P(w)$ = *(no. of occurrences of w in corpus) / (total no. of all word−occurrences in corpus)*

$P(\overline{w})$ = *(no. occurrences of words other than w in corpus) / (total no. of all word−occurrences in corpus)*

$P(g_i) = (|\{x \mid g(x) = g_i\}|)$ / *(total no. of all word−occurrences in corpus)*

$P(\overline{g}_i)=(|\{x \mid g(x) \neq g_i\}|)$ / *(total no. of word−occurrences in corpus)*

Note: The experiments, here, employ the information gain algorithm just once over the whole corpus and apply a threshold so that no document would result in a zero vector. This gives an "ideal" feature vector but does not investigate the performance of the feature selection algorithm.

2.2 Presentation features

There are many more features present in documents than words. These vary from linguistic features such as prevalence of adjectives, use of tense, and sentence complexity, to layout features such as line−spacing, tabulation, and non alpha−numeric characters. Space limitations preclude a detailed description of all features used in the set which comprises eighty nine such features, but an outline description is given below.

The feature extractor developed for this work employs a ’rough and ready’ Part−of−Speech tagger based on the Porter Stemming Algorithm (Porter, 1980). The algorithm analyzes word morphology and decides, where possible, if the word is a noun, verb, adjective, or adverb. However, it does not reduce the original word to its root. It is augmented by tables of closed−class words which include words normally considered stop−words. These tables allow the identification of pronouns, conjunctions, and articles. Simple hand−crafted rules combine the tables and the morphological analysis to aid verb tense identification. The program, therefore, would not compare well with any modern Part−of−Speech tagger, but accuracy should not be too important provided word tagging is consistent.

The tenses identified for verbs are restricted to past, present and future. However, the program also calculates the proportion of transitions in verb tense from one identified verb to the next. For example, if a verb was idenfied as past tense and the next one idenfied as being present tense, a "past−to−present" change would be recorded.

Frequencies of different closed−class word sets are calculated during the analysis. By word sets, here, we mean such things as days of the week, months of the year, signs of the zodiac etc. Some of these sets are general, others are specific to genre. While a term such as ’leo’ might not, by itself, be a particularly good discriminator for a horoscope genre, the fact that it is an astrological sign and *appears with other terms deemed astrological* may well be.

The mean and variance of sentence length, and the mean and variance of word length are measured. Sentence length, word length and syllable estimates are combined to give measures of sentence complexity. Mean word length divided by the mean sentence length, and the Flesch metric (Flesch, 1974) are also calculated.

The remainder of the presentation feature set comprise punctuation character usage, the use of upper and lower case characters, the amount of whitespace, and combinations of characters such as ":−)" often referred to as "smilies". Use of streams of punctuation marks to act as a section break in the text are also identified. Indentation, line−spacing, and tabulation are also measured.

All features are normalized over the document length and scaled to lie in the range [0,1] as this range is suitable for the SVM−light and C4.5 classifiers. If the feature is a count rather than a proportion the inverse of the count (minimum value one) is taken. Feature extraction results in a vector which is used by a classifier either in training or in the testing of a model.

3 Evaluation system

Both word based features and presentation features could be calculated from samples and their use compared in classification experiments. Experiments used these feature values with three different classifiers as it was thought that different classifiers might work better with one or other of the feature sets.

3.1 Data Set

Jaime Carbonell and Fang Liu of Carnegie Mellon University (CMU) supplied the data used in the experiments. The corpus is comprised of seven genres and is summarized in Table 2. The genres "Television News" and "Radio News" were predominantly produced by transcription systems and contain errors. (Whether these two classes are truly distinct genres is, perhaps, debatable.)

Genre	No. of Samples
Advertisement	1091
Bulletin Board	998
Frequently Asked Questions	1062
Message Board	1106
Radio News	2000
Reuters Newswire	2000
Television News	1448
TOTAL	9705

Table 2: Breakdown of CMU genre corpus

3.2 The classifiers

The three different classifier types used were: Naïve Bayes, C4.5 decision tree, and a Support Vector Machine. There are several methods for employing Bayes’ equation; the formulation used here is outlined below.

Bayes formula yields the conditional probability of a random variable X having value x, given that another random variable Y has value y. In adapting Bayes conditional probability formula to document classification, this work followed the treatment of Mitchell (1997) pp.174−184. A document is a series of tokens denoted $K(d)$. G denotes a set of genres. The probability that the genre of document d, $g(d)$, is $g_i \varepsilon G$, given that d is an arbitrary token series KS, is written:

$$P(g(d)=g_i|K(d)=KS) =$$
$$P(g(d)=g_i)P(K(d)=KS|g(d)=g_i) /$$
$$\Sigma_{gi\varepsilon G}[P(g(d)=g_i)P(K(d)=KS \mid g(d)=g_i)] \quad \text{(eqn.2)}$$

Classification only requires the most likely genre. The denominator in the above equation is constant so only the numerator needs to be considered. Using only those words with high Information Gain, WS, according to eqn. 1, a document d's words $W(d) \cap WS$ is often null, i.e. zero probability for all cases. Following Mitchell's smoothing method to prevent this, the most likely genre for a document d is the genre g_m such that $m=argmax_i$ of eqn. 2. The numerator value is calculated from:

$$P(g(d)=g_{i)} = |D(g_i)| / |C| \qquad \text{(eqn.3)}$$

$$P(K(d)=KS \mid g(d)=g_i) = \Pi_{w \ \varepsilon \ W(d)}(|\{k \ \varepsilon \ K(g_i) \mid W(k)=w \ \varepsilon \ WS\}| / |K(g_i)|)^{|(w,K(d))|}) \qquad \text{(eqn.4)}$$

where:

C is the document corpus and $D(g_i)$ are documents tagged as being of genre g_i.

The second classifier used was C4.5 (Quinlan 1993). This decision tree classifier uses the Information Gain algorithm to rank features. A tree is constructed where at each node is a decision by which the data are split into two groups using the feature with the most information gain among features not yet considered. Leaves are points at which a classification is made. The tree is then pruned by replacing a sub−tree with a leaf if the expected error is reduced. This alleviates over−fitting and reduces the complexity of the tree. The pruned tree is the resultant classifier for use on new data. During the classification phase for documents under test the rules at each node are applied to the corresponding document feature value to select the next node rule to apply. The document is classified when a leaf is reached. The heuristics used in the process are tunable, but we chose to use C4.5 with default settings.

The third classifier used was the Support Vector Machine (SVM) (Burges, 1998. Christianini, 2000) classifier which has received significant interest in recent years (Osuna, 1997 Joachims, 1998). The version of SVM used was SVM−light by Thorsten Joachims (Universitat Dortmund) (SVM−light webpages). This classifier has many tunable parameters and the vector space may have a function applied to it. Initial experiments had some trouble in getting models using linear vector space to converge in reasonable time. Using a radial basis function seemed to alleviate this problem. In all other respects experiments used SVM with default settings.[1] SVM−light builds binary models. In the case where multiple classes are present, as in our experiments, a model must be produced for each class. The classifier outputs real values rather than binary decisions. Each item is compared against each model and classified according to a winner−takes−all rule.

4 Experiments

The experiments detailed here were run under the ten−fold cross−validation method. This splits the data up into training and test sets in a 90%/10% proportion. Experiments are repeated ten times with the split being made in a round−robin fashion. In this way all of the data is used both in training and testing but not within the same cycle. Recorded here are Recall, Precision and F1 where recall is the number of correct classifications divided by number of documents, precision is the number of correct classifications divided by the number of classifications made, and F1=2 * (Precision * Recall) / (Precision + Recall). Note that in experiments where a classification is required (i.e. No "unknown" class) Recall, Precision and F1 are all equal.

323 word features were selected by analysing the Information Gain over the *whole* corpus. The selected words, therefore, are in some sense an "ideal" feature set. The length of the vector was chosen such that no zero vector would result.[2] Word counts were measured in each document. When using Naïve Bayes these counts were multiplied by the log probabilities calculated.

[1] Duplicate sub−vectors across class caused problems using SVM. Doubling the sub−vector length for the optimisation phase (using the −q parameter in SMV−light) increased training time but avoided the problem.

[2] The version of SVM light used consistently failed to converge when encountering duplicate vectors across classes. This situation easily arises if documents containing no words present in the feature set are present.

Genre	Ads	Bulletin	F.A.Q.	Message	Radio	Reuters	TV	Unknown
Ads	911	50	13	32	6	−	5	73
Bulletin Board	78	804	6	24	4	1	14	59
F.A.Q.	13	9	907	34	7	−	14	76
Message Board	20	17	27	875	25	3	25	108
Radio	10	7	8	22	1543	19	205	186
Reuters	−	2	−	3	22	1934	18	31
TV	6	10	12	23	234	10	952	193

Table 3: Misclassification Matrix for SVM using 323 word frequencies from CMU dataset

Genre	Ads	Bulletin	F.A.Q.	Message	Radio	Reuters	TV	Unknown
Ads	823	47	23	6	6	5	1	179
Bulletin Board	85	715	41	1	10	1		137
F.A.Q.	13	36	756	14	4	3		234
Message Board	7	1	6	1045	2	4	1	34
Radio	4	4	3		1640	5	277	67
Reuters	4	1	1	1	5	1966		22
TV					272		1147	21

Table 4: Misclassification Matrix using SVM with presentation features from CMU dataset

Genre	Ads	Bulletin	F.A.Q.	Message	Radio	Reuters	TV	Unknown
Ads	830	18	7	7	1	−	1	226
Bulletin Board	49	608	8	−	2	−	−	323
F.A.Q.	5	3	836	7	6	−	−	203
Message Board	2	1	3	996	4	2	−	92
Radio	−	2	−	−	1624	−	219	155
Reuters	1	−	−	−	−	1952	−	47
TV	−	−	−	−	87	−	1294	59

Table 5: Misclassification Matrix using SVM with presentation + word features from CMU dataset

For SVM and C4.5 logs of the counts were taken and divided by logs of the total word count in the corresponding document. (Smoothing carried out by adding 1 to both numerator and denominator prior to taking logs.) A misclassification matrix example is shown in Table 3. The true genre tag is indicated by row and the classifier's decision is listed by column. The number of correct classifications appear on the diagonal of the table, and the numbers of misclassifications are shown in the remaining column cells.

SVM and C4.5 experimnets used presentation feature values directly. Using the Naïve Bayes classifier requires that feature value ranges be defined because Bayesian classifiers usually work with features that are either present or absent. However, Mitchell's formulation generalizes to real−valued functions: The conditional probability that the genre of document d is g_i, given that feature f is in d, $v(f,d)$, is a real value V, given by:

$$V = v(f,d)[\Sigma_{d \varepsilon D(gi)} v(f,d)] / |D(gi)| \qquad \text{(eqn.5)}$$

where:

$D(g_i)$ is the subset of the training corpus tagged as being of genre g_i.

Table 4 shows an example misclassification matrix resulting from using presentation feature vectors.

Misclassifications made using word frequency features were different from those made using presentation features. The question arises as to whether combining the two feature sets would show improvement. To test this, each document had its word frequency and presentation feature vectors combined. An

Classifier	Word frequency features	Presentation features	Combined features
Naïve Bayes	77.8% +/– 1.6%	64.0% +/– 1.2%	83.1% +/– 1.5%
C4.5 Decision Tree	79.8% +/– 1.1%	85.3% +/– 1.0%	87.8% +/– 1.1%
SVM	85.4% +/– 0.9%	87.1% +/– 1.0%	92.1% +/– 0.8%

Table 6: Average recall in 10−fold cross validation genre identification experiment; forced decision

	Classifier	Word frequency ftrs.	Presentation ftrs.	Combined ftrs.
Recall	Naïve Bayes	76.7% +/– 1.4%	33.9% +/– 1.4%	82.4% +/– 1.5%
	SVM	81.8% +/– 0.6%	83.6% +/– 0.9%	84.0% +/– 2.5%
Precision	Naïve Bayes	80.4% +/– 1.3%	78.8% +/– 1.4%	84.1% +/– 1.4%
	SVM	88.4% +/– 1.3%	90.1% +/– 2.4%	94.9% +/– 0.7%
F1 Metric	Naïve Bayes	78.5% +/– 1.4%	47.4% +/– 1.5%	83.2% +/– 1.4%
	SVM	85.0% +/– 0.4%	86.7% +/– 0.9%	89.1% +/– 1.5%

Table 7: Mean recall in 10−fold cross validation experiment; positive identification threshold applied

example misclassification matrix resulting from using combined feature set vectors is shown in Table 5.

4.1 Results

In the first set of experiments each classifier's "best guess" at a document's genre was taken as the class. The results are shown in Table 6. Each cell shows average recall with an error margin quoted at one standard deviation.

The C4.5 classifier is not able to be configured to allow an unknown classification. It is possible for Naïve Bayes and SVM to be so configured, since their results give a numeric value for the most likely classification. A document is deemed "unknown" if it scores less than 0.5 using Naïve Bayes, or negatively using SVM. Recall and precision figures can be calculated in using this scheme. The results are shown in Table 7. (Mean value with one standard deviation error margin.)

4.2 Discussion

Use of the presentation feature set yields a significant advatage over use of word frequencies except when using Naïve Bayes. This shows that presentations features alone, when used with a suitable classifier, are pertinent to classifying by genre without the need for word features often used. The advantage in combining feature values is seen consistently over the three classifiers and is true even for the Naïve Bayes classifier which does poorly with presentation features alone. Applying a threshold to the classifier output score, to allow an "unknown" class, increases precision at some expense in recall as should be expected.

When the Naïve Bayes classifier used the 89 presentation features, the results were considerably less accurate than when it used the 323 word features. It seems likely that at least three factors were involved. Firstly, Naïve Bayes assumes feature independence. While this is not true for words, it has been the experience of the IR community that word dependencies are small enough for documents to be treated as "bags−of−words". Some features in the presentation feature set, however, are far from independent; e.g., the proportions of parts of speech identified are explicitly linked. Secondly, the Bayesian formulation in Section 3.2 has an implicit assumption of monotonicity. If word *w* occurs in document *d* a total of $n=|(w,K(d))|$ times, the quantity $|(w,K(d))|$, being an exponent in the formula, gives word *w* more weight when it occurs more times. Let genre g_j be the genre most strongly associated with *w* in the training corpus. The more times *w* occurs in *d*, the more likely the Naïve Bayes classifier is to classify *d* as genre g_j; and this effect is monotonic. But monotonicity does not always hold, even with words; and it fails to hold even more often in the presentation features. For example, in our training data, the word "said" occurs seldom in the genres: Message−board, Bulletin−board, and

FAQ. It occurs very frequently in Reuters−newswire, and with intermediate frequency in Radio and TV transcripts. Thirdly, the word−feature−counts are integer−valued, while the presentation features have continuous values. The effect of assigning ranges of continuous values to discrete value bins, in the use of presentation features with Naïve Bayes, is not yet clear. There has yet to be performed an error analysis to quantify the reason(s) for the observed accuracy decrease.

The large increase in precision with only a small decrease in recall when allowing an "unknown" class suggests that a more relaxed threshold could be applied to SVM or Naïve Bayes output.

5 Conclusion

Experiments have demonstrated success in creating genre models for automatic recognition of multiple genres using a corpus of sufficient size to draw some conclusions. These experiments have shown that linguistic and format features alone can be used successfully in sorting documents into different genres, and that performance is at least as good as use of word based features providing a suitable classifier is used. Rather than presentation features aiding discimination by words it seems selected word features assist discrimination by presentation features as the best results are obtained using a combination. The results suggest that automatic genre identification could be used in applications, such as Information Retrieval, for better performance. For example, this technique could improve IR performance against tasks such as the TREC web track (TREC webpages). Future work will investigate the effects of having an increased number of genres and increased corpus size.

6 References

Biber, D. (1995) "Variation Across Speech and Writing", Cambridge University Press, New York.

Berger, A., S.Della Pietra, V.Della Pietra, ˆ(1996) "A Maximum Entropy Approach to Natural Language Processing", Computational Linguistics, vol. 22 No. 1, pp39−71.

Burges, C. (1998) "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, No. 2.

Cherney, L. (1999) "Conversation and Community: Chat in a Virtual World", CSLI Publications, Stanford University.

Cohen, W.W. (1996) "Learning Rules that Classify E−Mail", AAAI Spring Symposium on Machine Learning in Information Access.

Cristianini, N., J.Shaw−Taylor, (2000) "Introduction to Support Vector Machines", Cambridge University Press, .

Flesch, R. (1974) "The Art of Readable Writing", Harper and Row, New York.

Illouz, G., B.Habert, H.Folch, S.Heiden, S.Fleury, P.Lafon, S.Prevost (2000) "TyPTex: Genric features for Text Profiler", Proc. RIAO 2000, Paris, France.

Joachims, T. (1998) "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", European Conference on Machine Learning.

Karlgren, J., and D Cutting (1994) "Recognizing Text Genres with Simple Metrics Using Discriminant Analysis". In Proc. of the 15 th International conference on Computational Linguistings (COLING'94)

Kessler, B., G Nunberg, H. Schutze (1997) "Automatic Detection of Text Genre". In Proc. if 35 th Annual Meeting of the ACL and 8th Conference of ECACL.

May, A. (1997) "Automatic Classification of E−Mail Messages by Message Type", JASIS, vol 48 No.1, pp32−39.

Mitchell, T.M. (1997) "Machine Learning", McGraw−Hill, Boston, Massachusetts.

Oard, D.W., N. De Claris, B.J.Dorr, C.Faloutsos, (1994) "On Automatic Filtering of Multilingual Texts", Proceedings of IEEE International Conference on Sytems, Man, and Cybernetics, pp1645−1650, Vsan Antonio Texas.

Osuna, E. R.Freund, and F.Girosi, (1997) "Training support vector machines: an application to face recognition", IEEE Conference on Computer Vision and Pattern Recognition.

Pannu, Anandeep, Sycara. (1996) "Learning Text Filtering Preferences", Symposium on Machine Learning and Information Processing, AAAI Symposium Series, Stanford, Ca., March'96.

Porter, M.E. (1980) "An Algorithm for Suffix Stripping", Program Vol. 14, No. 3, pp318−327, July'80.

Quinlan, J.R. (1993) "C4.5: Programs for Machine Learning", Morgan Kaufmann, California.

Sahami, M., S.Dumais, D.Heckerman, E.Horvitz, (1998) "A Bayesian Approach to Filtering Junk E−Mail" AAAI Workshop Technical Report WS−98−05.

Stamatatos, E., N. Fakotakis, and G. Kokkinakis (2000) "Text Genre Detection Using Common Word Frequencies". In the Proc. of the 18 th International Confernece on Computational Linguistics (COLING2000)

SVM−light webpages URL: http://ais.gmd.de/~thorsten/svm_light/

TREC webpages. URL: http://trec.nist.gov

Yang, Y., J.Pedersen, (1997) "A Comparative Study on Feature Selection in Text Categorization", Proceedings of the 1997 International Conference on Machine Learning (ICML), pp.412−420.