Statistical Identification of Key Phrases for Text Classification

Frans Coenen, Paul Leng, Robert Sanderson, and Yanbo J. Wang

Department of Computer Science, The University of Liverpool,
Ashton Building, Ashton Street, Liverpool L69 3BX, United Kingdom
{frans,phl,azaroth,jwang}@csc.liv.ac.uk

Abstract. Algorithms for text classification generally involve two stages, the first of which aims to identify textual elements (words and/or phrases) that may be relevant to the classification process. This stage often involves an analysis of the text that is both language-specific and possibly domain-specific, and may also be computationally costly. In this paper we examine a number of alternative keyword-generation methods and phrase-construction strategies that identify key words and phrases by simple, language-independent statistical properties. We present results that demonstrate that these methods can produce good classification accuracy, with the best results being obtained using a phrase-based approach.

Keywords: Text Classification, Text Preprocessing.

1 Introduction

The increasing volume and availability of electronic documents, especially those available on-line, has stimulated interest in methods of *text classification* (TC). TC algorithms typically make use of a classifier developed from analysis of a training set of documents that have been previously classified manually. The training process usually involves two stages: first, a *preprocessing* stage to identify relevant textual characteristics in the documents of the training set, and second, a learning stage in which these characteristics are associated with class labels. We are in this paper especially concerned with TC methods that use this second stage to develop *Classification Rules* by a process of Classification Association Rule Mining (CARM).

CARM methods, and other related rule-based classification systems, require the initial preprocessing stage to identify textual components (words or phrases) that can be used in the construction of classification rules of the form $A \rightarrow c$, where A is the conjunction of a set of these components and c is a class label. Much current work on document preprocessing makes use of techniques tailored to either the language in which the documents to be classified are written (e.g. English, Spanish, Chinese, etc.) or the particular domain that the documents describe (e.g. *medline* abstracts, Biological texts, etc.). Knowledge of the language used allows the application of techniques such as natural language parsing

P. Perner (Ed.): MLDM 2007, LNAI 4571, pp. 838–853, 2007.

and stemming, and the use of stop and synonym lists. Knowledge of the domain allows the application of specialised dictionaries and lexicons or the use of sophisticated ontology structures. These approaches can produce very accurate classifiers, but are costly to implement, in terms of human resources, as they are not generally applicable, and the techniques involved may also be relatively costly in computational terms. These reasons motivate a search for methods that will identify relevant words and phrases by statistical techniques, without the need for deep linguistic analysis or domain-specific knowledge.

In this paper we examine a number of such methods for identifying key phrases (and words) in document sets to support TC. The methods all begin by using language-independent statistical methods to identify *significant* words in the document set: i.e. words that are likely to be relevant to the classification task. We investigate a number of strategies for constructing phrases, all of which make use only of simple textual analysis using significant words derived in this way. Eight different methods of generating the significant words are considered, coupled with four phrase formulation algorithms. We compare the phrase-generation methods with results obtained from simpler "bag of words" approaches. Our results demonstrate that the shallow linguistic analysis employed in our preprocessing is nevertheless sufficient to produce good classification accuracy, and that even simple phrase-construction approaches can improve on single-word methods.

The rest of this paper is organised as follows. In section 2 we describe the background and some related work relevant to this study. Section 3 outlines the CARM algorithm and the data sets that we have used to evaluate the various preprocessing strategies. Section 4 describes the methods we use for identification of significant words, and section 5 the phrase-construction algorithms. In section 6 we present experimental results, and in section 7 our conclusions.

2 Previous Work

Text for TC purposes is usually represented using the vector space model, where each document is represented as a single numeric vector d, and d is a subset of some vocabulary V. The vocabulary V is a representation of the set of textual components that are used to characterise documents. Two broad approaches are used to define this: the *bag of words* and the *bag of phrases* approaches.

In the bag of words approach each document is represented by the set of words that is used in the document. Information on the ordering of words within documents as well as the structure of the documents is lost. The vectors representing documents may comprise either (a) Word identification numbers (the *binary representation*), or (b) Words weighted according to the frequency with which they appear in the document (the *term-weighted representation*). The problem with the approach is how to select a limited, computationally manageable, subset of words from the entire set represented in the document base. Methods include the use of stop and synonym lists and stemming, or the use of a domain-dependent set of key words or named entities. These are all options that make use of knowledge of the language in which the documents in the document set are written, an approach which, for reasons discussed above, we wish to avoid.

In the bag of phrases approach each element in a document vector represents a phrase describing an ordered combination of words appearing in sequence, either contiguously or with some maximum word gap. A variety of techniques exist for identifying phrases in documents, most of which again make use of some kind of meta knowledge (either the application domain or the language used in the document set). For example Sharma and Raman in [10] propose a phrase-based text representation for web document management using rule-based Natural Language Processing (NLP) and a Context Free Grammar (CFG). In [4] Katrenko makes an evaluation of the phrase-based representation.

In [6] and [8] a sequence of experiments is described comparing the bag of keywords approach with the bag of phrases approach in the context of text categorisation. The expectation was that the phrase based approach would work better than the keyword approach, because a phrase carries more information; however the reverse was discovered. In [9] a number of reasons for this are given:

1. Phrases have inferior statistical properties.
2. Phrases have lower frequency of occurrence than keywords.
3. The bag of phrases includes many redundant and/or noise phrases.

We hypothesise that these drawbacks can be overcome by the use of appropriate classification algorithms. It is clear that phrases will be found in fewer documents than corresponding key words, but conversely we expect them to have a greater discriminating power. To take advantage of this, we require algorithms that will identify classification rules with relatively low applicability as well as very common ones. To avoid problems of noise, conversely, we require the ability to discard rules that fall below defined thresholds of validity. These requirements point us to the use of CARM algorithms to construct classification rules using the identified words and/or phrases. CARM approaches are based on methods of Association Rule Mining that rely on the examination of large data sets to identify even scarce rules without overfitting. A number of studies (e.g. [1], [7], etc.) have demonstrated that, for many classification problems, CARM approaches can lead to better classification accuracy than other methods. Earlier work by the authors [2] [3], employing a CARM algorithm, TFPC, showed that appropriate selection of thresholds led to high classification accuracy in a wide range of cases. In the present work we seek to apply this algorithm to the TC problem, and to identify parameter values to optimise its accuracy.

3 Experimental Organisation

All experiments described in this paper were undertaken using the authors' TFPC algorithm [2] [3]. TFPC (Total From Partial Classification) is a CARM algorithm that constructs a classifier by identifying Classification Association Rules (CARs) from a set of previously-classified cases. A CAR is a special case of an Association Rule for which the consequent is a class-label. As is the case for association rules in general, CARs can be characterised by their *support* (the relative frequency with which the rule is found to apply), and *confidence* (the ratio of their support to the frequency of the antecedent of the rule). An appropriate selection of threshold values for support and confidence is used to define a set of rules from which the classifier is constructed. The unusual feature of TFPC is that, when the algorithm finds a general rule that meets its threshold requirements, it does not search for any more specific rule whose antecedent is a superset of this. This heuristic makes TFPC less prone to overfitting than other CARM methods that follow an "overfit and prune" strategy, while still enabling the identification of low-support rules. These characteristics make TFPC a realistic choice for TC in potentially noisy environments.

The experimental analysis was undertaken using a subset of the Usenet collection, a set of documents compiled by Lang [5] from 20 different newsgroups, often referred to as the "20 Newsgroup" collection. There are exactly 1,000 documents per group (class) with the exception of one class that contains only 997. For our experiments the collection was split into two data sets covering 10 classes each: NGA.D10000.C10 and NGB.D9997.C10, and the analysis was undertaken using NGA.D10000.C10.

4 Phrase Identification

The phrase identification approach we employed proceeds as follows, for each document in the training set:

1. Remove *common* words, i.e. words that are unlikely to contribute to a characterisation of the document.
2. Remove *rare* words, i.e. words that are unlikely to lead to generally applicable classification rules.
3. From the remaining words select those *significant* words that serve to differentiate between classes.
4. Generate significant phrases from the identified significant words and associated words.

4.1 Noise Word Identification

We define words as continuous sequences of alphabetic characters delimited by non-alphabetic characters, e.g. punctuation marks, white space and numbers. Some non-alphabetic characters (',', '.', ':', ';', '!' and '?'), referred to as *stop marks*, play a role in the identification of phrases (more on this later). All other non-alphabetic characters are ignored.

Common and rare words are collectively considered to be *noise* words. These can be identified by their *support* value, i.e. the percentage of documents in the training set in which the word appears. Common words are words with a support value above a user defined Upper Noise Threshold (UNT), which we refer to as Upper Noise Words (UNW). Rare words are those with a support value below a user defined Lower Noise Threshold (LNT), and are thus referred to as Lower Noise Words (LNW).

The UNT must of course exceed the LNT value, and the distance between the two values determines the number of identified non-noise words and consequently, if indirectly, the number of identified phrases. A phrase, in the context of the TFPC algorithm, represents a possible attribute of a document which may be a component of the antecedent of rules. Some statistics for the NGA.D10000.C10 set, using LNT = 1% and UNT = 50% are presented in Table 1. It can be seen that the majority of words occur in less than 1% of documents, so LNT must be set at a low value so as not to miss any potential significant words. Relatively few words are common, appearing in over 50% of the documents.

Table 1. Statistics for 20 Newsgroup data sets A and B using LNT = 1% and UNT = 50%

Data Set	**# words**	**# LNW**	**# UNW**	**% LNW**	**% UNW**
NGA.D10000.C10	49,605	47,981	21	96.73	0.04
NGB.D9997.C10	47,973	46,223	22	96.35	0.05

Tables 2 and 3 list the most common words (support greater than 40%) in the two 20 Newsgroup sets. Figures in parentheses indicate the number of documents where the word appears; recall that there are 10,000 and 9,997 documents in the two sets respectively. Note that NGB.D9997.C10 set contains the additional common word “but”.

Table 2. Number of common words (UNT = 40%) in NGA.D10000.C10

a (7,666)	and (7,330)	are (4,519)	be (4,741)	for (6,367)	have (5,135)
i (6,803)	in (7,369)	is (6,677)	it (5,861)	not (4,565)	of (7,234)
on (5,075)	re (5,848)	that (6,012)	the (8,203)	this (5,045)	to (7,682)
with (4,911)	writes (4,581)	you (5,015)			

Table 3. Number of common words (UNT = 40%) in NGB.D9997.C10

a (7,837)	and (7,409)	are (4,807)	be (5,258)	but (4,633)	for (6,401)
have (5,366)	i (6,854)	in (7,579)	is (6,860)	it (6,169)	not (4,849)
of (7,546)	on (5,508)	re (6,267)	that (6,515)	the (8,427)	this (5,333)
to (7,905)	with (4,873)	writes (4,704)	you (5,013)		

4.2 Significant Word Identification

The desired set of *significant* words is drawn from an ordered list of *potential significant* words. A potential significant word is a non-noise word whose *contribution* value exceeds some user specified threshold G. The contribution value of a word is a measure of the extent to which the word serves to differentiate between classes and can be calculated in a number of ways. For the study presented here we considered two methods: (a) Using support counts only, and (b) Term weighting.

Contribution from support counts only is obtained using the following identify:

$$\text{Contribution } G_{wi} \text{ of word } w \text{ with respect to class } i = \frac{S_{wi} \times D}{S_w \times S_i}$$

Where D is the total number of documents in the training set, S_i is the number of documents that are labelled as class i, S_{wi} is the number of documents in class i that contain word w, and S_w is the total number of documents that contain word w. The ratio $\frac{S_w}{D}$ describes the overall frequency of occurrence of word w in the document set. If the ratio $\frac{S_{wi}}{S_i}$ is greater than this, then the contribution value G_{wi} will be greater than 1, indicating that w may be a significant word for class i. In practice, of course, even words with little significance may have contribution values slightly greater than 1, so to indicate a significant contribution we require G_{wi} to exceed some threshold value $G > 1$. The maximum value of the contribution can reach is $\frac{D}{S_i}$, obtained when $\frac{S_{wi}}{S_w} = 1$, indicating that w occurs only in class i. In the case of the NGA.D10000.C10 set, we have ten classes of exactly 1,000 documents each, so the maximum contribution value is 10. The algorithm for calculating contribution values using support counts is given in Table 4.

Table 4. Algorithm for calculating contribution using support counts

```
$G \leftarrow$ significance threshold
$w \leftarrow$ the given word
$C \leftarrow$ set of available classes
$D \leftarrow$ total number of documents
$S_w \leftarrow$ number of documents that contain $w$
for each $c_i$ in $C$ from $i = 1$ to $|C|$ {
    $S_i \leftarrow$ number of documents labelled as in class $c_i$
    $S_{wi} \leftarrow$ number of documents in $c_i$ that contain $w$
    $S_{Li} \leftarrow \frac{S_{wi}}{S_i}$
    contribution $\leftarrow \frac{S_{Li} \times D}{S_w}$
    if (contribution $> G$) then $w$ is a significant word
    }
```

We apply a similar approach when term weighting is used. TF-IDF (Term Frequency - Inverse Document Frequency) [11] is a well established term weighting technique. Our variation of this is defined as follows:

$$\text{Contribution } G_{wi} \text{ of word } w \text{ with respect to class } i = \frac{TF_{wi} \times N}{TF_w \times N_i}$$

Where TF_{wi} is the total number of occurrences of w in documents in class i, N is the total number of words in the document set, N_i is the total number of words contained in documents labeled as class i, and TF_w is the total number of occurrences of the word w in the document set. The ratio $\frac{TF_w}{N}$ defines the overall term frequency of w in the document set; if the corresponding ratio $\frac{TF_{wi}}{N_i}$ is significantly greater than this, then a contribution value G_{wi} greater than 1 will indicate a potential significant word. The algorithm for calculating contribution values using term weighting is given in Table 5.

Table 5. Algorithm for calculating contribution using term weighting

```
$G \leftarrow$ significance threshold
$w \leftarrow$ the given word
$C \leftarrow$ set of available classes
$N \leftarrow$ total number of words in the document base
$T_w \leftarrow$ total number of occurrences of word $w$
for each $c_i$ in $C$ from $i = 1$ to $|C|$ {
    $T_{wi} \leftarrow$ total number of occurrences of word $w$ in $c_i$
    $N_i \leftarrow$ total number of words in $c_i$
    contribution $\leftarrow \frac{T_{wi} \times N}{T_w \times N_i}$
    if (contribution $> G$) then $w$ is a significant word
    }
```

Thus we have two options for calculating the contribution of a word, using support counts or using term weightings. We place those whose contribution exceeds the threshold G into a potential significant words list ordered according to contribution value. This list may include words that are significant for more than one class, or we may choose to include only those non-noise words with contribution greater than G with respect to one class only (i.e. *uniques*).

From the potential significant words list we select the final list of significant words from which we generate phrases. We have examined two strategies for doing this. The first method, which simply selects the first (most significant) K words from the ordered list, may result in an unequal distribution between classes. In the second approach we select the top $\frac{K}{|C|}$ for each class (where $|C|$ is the number of available classes), so as to include an equal number of significant words for each class. Thus, in summary, we have:

- Two possible contribution selection strategies (support count and term weighting).
- Two potential significant word list construction strategies (include all words with appropriate level of contribution, or alternatively only unique words).
- Two significant word selection strategies (top K or top $\frac{K}{|C|}$ for each class).

These possibilities define eight different methods for the identification of significant words. Tables 6 and 7 illustrate some consequences of these options. Table 6 gives the distribution of significant words per class for the NGA.D10000.C10 set using the "support count, all words and top K strategy" with UNT = 7%, LNT = 0.2%, $G = 3$. Note that the number of significant words per class is not balanced, with the general "forsale" class having the least number of significant words and the more specific "mideast" class the most. Table 7 shows the 10 most significant words for each class using the same strategy and thresholds. The value shown in parentheses is the contribution of the word to that class in each case. Recall that using the support count strategy the highest possible contribution value for the NGA.D10000.C10 set is 10, obtained when the word is unique to a certain class. In the "forsale" category quite poor contribution values are found, while the "mideast" category has many high contribution words.

Table 6. Number of significant words in NGA.D10000.C10 using the "support count, all words and top K strategy" with UNT = 7.0%, LNT = 0.2%, G = 3

Class Label	**# Sig. Words**	**Class Label**	**# Sig. Words**
comp.windows.x	384	rec.motorcycles	247
talk.religion.misc	357	sci.electronics	219
alt.atheism	346	misc.forsale	127
sci.med	381	talk.politics.mideast	1,091
comp.sys.ibm.pc.hardware	175	rec.sport.baseball	360

Table 7. Top 10 significant words per class for NGA.D10000.C10 using the "support count, all words and top K strategy" with UNT = 7.0%, LNT = 0.2%, G = 3

windows.x	motorcycles	religion	electronics	atheism
colormap(10)	behanna(10)	ceccarelli(10)	circuits(9.8)	inimitable(10)
contrib(10)	biker(10)	kendig(10)	detectors(9.6)	mozumder(10)
imake(10)	bikers(10)	rosicrucian(10)	surges(9.5)	tammy(10)
makefile(10)	bikes(10)	atf(9.5)	ic(9.3)	wingate(10)
mehl(10)	cages(10)	mormons(9.5)	volt(9.3)	rushdie(9.8)
mwm(10)	countersteering(10)	batf(9.3)	volts(9.2)	beauchaine(9.7)
olwn(10)	ducati(10)	davidians(9.2)	ir(9.2)	benedikt(9.4)
openlook(10)	fxwg(10)	abortions(9.0)	voltage(9.2)	queens(9.4)
openwindows(10)	glide(10)	feds(8.9)	circuit(8.9)	atheists(9.3)
pixmap(10)	harley(10)	fbi(8.8)	detector(8.9)	sank(9.1)
forsale	med	mideast	hardware	baseball
cod(10)	albicans(10)	aggression(10)	nanao(10)	alomar(10)
forsale(9.8)	antibiotic(10)	anatolia(10)	dma(9.4)	astros(10)
comics(9.5)	antibiotics(10)	andi(10)	vlb(9.4)	baerga(10)
obo(9.0)	candida(10)	ankara(10)	irq(9.3)	baseman(10)
sale(8.8)	diagnosed(10)	apartheid(10)	soundblaster(9.0)	batter(10)
postage(8.6)	dyer(10)	appressian(10)	eisa(8.8)	batters(10)
shipping(8.6)	fda(10)	arabs(10)	isa(8.8)	batting(10)
mint(8.4)	homeopathy(10)	argic(10)	bios(8.7)	bullpen(10)
cassette(8.2)	infections(10)	armenia(10)	jumpers(8.7)	cardinals(10)
panasonic(7.6)	inflammation(10)	armenian(10)	adaptec(8.7)	catcher(10)

5 Phrase Identification

Whichever of the methods described above is selected, we define four different categories of word:

1. **Upper Noise Words (UNW):** Words whose support is above a user defined Upper Noise Threshold (UNT).
2. **Lower Noise Words (LNW):** Words whose support is below a user defined Lower Noise Threshold (LNT).

Table 8. Phrase generation strategies

Delimiters	**Contents**	**Label**
Stop marks and noise words	Sequence of one or more significant words and ordinary words	DelSNcontGO
	Sequence of one or more significant words and ordinary words replaced by "wild cards"	DelSNcontGW
Stop marks and ordinary words	Sequence of one or more significant words and noise words	DelSOcontGN
	Sequence of one or more significant words and noise words replaced by "wild cards"	DelSOcontGW

Table 9. Example of significant word identification process using a document from the NGA.D10000.C10 data set

@Class rec.motorcycles
paint jobs in the uk
can anyone recommend a good place for reasonably
priced bike paint jobs, preferably but not
essentially in the london area.
thanks
john somename.
–
acme technologies ltd xy house,
147 somewherex road

(a) *Example document from NGA.D10000C10 data set in its unprocessed form*

paint jobs in the uk can anyone recommend a good place for reasonably priced bike paint jobs # preferably but not essentially in the london area # thanks john somename # acme technologies ltd xy house # somewherex road

(b) *Document with stop marks indicated and non-alphabetic characters removed*

paint jobs $\widehat{\textit{in}}$ $\widehat{\textit{the}}$ *uk* $\widehat{\textit{can}}$ $\widehat{\textit{anyone}}$ *recommend a* $\widehat{\textit{good}}$ *place* $\widehat{\textit{for}}$ *reasonably priced* $\widetilde{\textit{bike}}$ *paint jobs # pre*$\widetilde{\textit{ferab}}$*ly* $\widehat{\textit{but}}$ $\widehat{\textit{not}}$ *essentially* $\widehat{\textit{in}}$ $\widehat{\textit{the}}$ *london area #* $\widehat{\textit{thanks}}$ $\overline{\textit{john}}$ $\overline{\textit{somename}}$ *# acme* $\widetilde{\textit{technologies}}$ *ltd* $\overline{\textit{xy}}$ *house #* $\overline{\textit{somewherex}}$ $\widetilde{\textit{road}}$

(c) *Document with* $\overline{\textit{lower}}$, $\widehat{\textit{upper}}$ *and* $\widetilde{\textit{significant}}$ *words marked (all other words are ordinary words)*

3. **Significant Words (SW):** Selected key words that serve to distinguish between classes.
4. **Ordinary Words (OW):** Other non-noise words that were not selected as significant words.

We also identify two groups of categories of words:

1. **Non-Noise Words (NNW):** The union of significant and ordinary words.
2. **Noise Words (NW):** The union of upper and lower noise words.

These categories are all used to describe the construction of phrases. We have investigated four different simple schemes for creating phrases, defined in terms of rules describing the content of phrases and the way in which a phrase is delimited. In all cases, we require a phrase to include at least one significant word. In addition to this, Table 8 shows the four different algorithms used for the experiments described here.

An example illustrates the consequences of each method. In Table 9a we show a document taken from the NGA.D10000.C10 data set (with some proper names changed for ethical reasons). Note that the first line is the class label and plays no part in the phrase generation process. The first stage in preprocessing replaces all stop marks by a # character and removes all other non-alphabetic characters (Table 9b). In Table 9c the document is shown "marked up" after the significant word identification has been completed. Significant words are shown using "wide-tilde" ($\widetilde{abc...}$), upper noise words use "wide-hat" ($\widehat{abc...}$), and lower noise words use "over-line" ($\overline{abc...}$).

In Table 10 we show the phrases used to represent the example document from Table 9 using each of the four different phrase identification algorithms. Where appropriate "wild card" words are indicated by a '?' symbol. Note that a phrase can comprise any number of words, unlike *word-gram* approaches where words are a fixed length. The phrase identified in a document become its *attributes* in the classification process.

Table 10. Example phrases (attributes) generated for example document given in Table 9 using the four advocated phrase identification strategies

Phrase Identification Algorithm	**Example of Phrase Representation (Attributes)**
DelSNcontGO	$\{\{\widetilde{road}\}, \{\widetilde{preferably}\}, \{reasonably\ priced\ \widetilde{bike}\ paint\ jobs\}, \{acme\ \widetilde{technologies}\ ltd\}\}$
DelSNcontGW	$\{\{\widetilde{road}\}, \{\widetilde{preferably}\}, \{?\ ?\ \widetilde{bike}\ ?\ ?\}, \{?\ \widetilde{technologies}\ ?\}\}$
DelSOcontGN	$\{\{\overline{somewherex}\ \widetilde{road}\}, \{\widetilde{preferably}\ \widehat{but}\ \widehat{not}\}, \{\widetilde{bike}\}, \{\widetilde{technologies}\}\}$
DelSOcontGW	$\{\{?\ \widetilde{road}\}, \{\widetilde{preferably}\ ?\ ?\}, \{\widetilde{bike}\}, \{\widetilde{technologies}\}\}$

6 Experimental Results

Experiments conducted using the NGA.D10000.C10 data set investigated all combinations of the eight different proposed significant word generation strategies with the four proposed different phrase generation approaches. We also investigated the effect of using the generated significant words on their own as a "bag of keywords" representation. The suite of experiments described in this section used the first 9/10th (9,000 documents) as the training set, and the last

1/10th (1,000 documents) as the test set. We used the TFPC algorithm to carry out the classification process. For all the results presented here, the following thresholds were used: support = 0.1%, confidence = 35.0%, UNT = 7.0%, LNT = 0.2%, $G = 3$, and maximum number of significant words threshold of 1,500. These parameters produced a word distribution that is shown in Table 11. As would be expected the number of potential significant words is less when only unique words (unique to a single class) are selected. Note also that using word frequency to calculate the contribution of words leads to fewer significant words being generated than is the case when using the "word support calculation" which considers only the number of documents in which a word is encountered.

Table 11. Number of potential significant words calculated per strategy (NGA.D10000.C10)

Number of Noise Words above UNT	208			
Number of Noise Words below LNT	43,681			
Number of Ordinary Words	4,207			
Number of Significant Words	1,500			
Number of Words	49,596			
	Word Frequency		**Word Support**	
	Unique	**All**	**Unique**	**All**
Number of Potential Significant Words	2,911	3,609	3,188	3,687

Table 12. Number of attributes (phrases) generated (NGA.D10000.C10)

	Word Frequency				**Word Support**			
	Unique		**All**		**Unique**		**All**	
	Dist	**Top K**	**Dist**	**Top K**	**Dist**	**Top K**	**Dist**	**Top K**
DelSNcontGO	27,551	27,903	26,973	27,020	26,658	25,834	26,335	25,507
DelSNcontGW	11,888	12,474	12,118	13,657	11,970	11,876	11,819	11,591
DelSOcontGN	64,474	63,134	60,561	61,162	59,453	58,083	59,017	57,224
DelSOcontGW	32,913	34,079	32,549	35,090	32,000	32,360	31,542	31,629
Keywords	1,510	1,510	1,510	1,510	1,510	1,510	1,510	1,510

Table 12 shows the number of attributes generated using all the different combinations of the proposed significant word generation and phrase generation strategies, including the case where the significant words alone were used as attributes (the "keyword" strategy). In all cases, the algorithms use as attributes the selected words or phrases, and the ten target classes. Thus, for the keyword strategy the number of attributes is the maximum number of significant words (1,500) plus the number of classes (10). In other experiments, we examined the effect on the keyword strategy of removing the upper limit, allowing up to 4,000 significant words to be used as attributes, but this led to reduced accuracy, suggesting that a limit on the number of words used is necessary to avoid including words whose contribution may be spurious.

Table 13. Classification accuracy (NGA.D10000.C10)

	Word Frequency				Word Support			
	Unique		All		Unique		All	
	Dist	Top K	Dist	Top K	Dist	Top K	Dist	Top K
DelSNcontGO	75.9	73.6	**77.3**	72.4	76.4	73.2	**77.4**	74.5
DelSNcontGW	75.1	71.6	**76.2**	68.5	74.9	71.3	**75.8**	72.3
DelSOcontGN								
DelSOcontGW			**70.9**		70.4	66.0	**71.2**	68.9
Keywords	75.1	73.9	**75.8**	71.2	74.4	72.2	**75.6**	73.7

In the DelSNcontGO and DelSNcontGW algorithms, stop and noise words are used as delimiters. As the results demonstrate, this leads to many fewer phrases being identified than is the case for the other two phrase generation strategies, which use stop words and ordinary words as delimiters. For DelSOcontGN (and to a lesser extent DelSOcontGW) the number of attributes generated usually exceeded the TFPC maximum of 2^{15} (32,767) attributes. This was because these algorithms allow the inclusion of noise words in phrases. Because there are many more noise words (43,889) than ordinary words (4,207), the number of possible combinations for phrases far exceeds the number obtained using the two DelSN strategies. Further experiments which attempted to reduce the number of phrases produced by adjusting the LNT, UNT and G thresholds did not lead to good results, and led us to abandon the DelSOcontGN and DelSOcontGW strategies.

Variations within the DelSN strategies were less extreme. DelSNcontGW produces fewer attributes than DelSNcontGO because phrases that are distinct in DelSNcontGO are collapsed into a single phrase in DelSNcontGW. Intuitively it might seem that identifying more attributes (phrases) would improve the quality of representation and lead to better classification accuracy. In other experiments we increased the number of attributes produced by the DelSNcontGO and DelSNcontGW strategies by increasing the limit on the number of significant words generated. However, as was the case with the keywords strategy, this did not lead to any better accuracies, presumably because the additional significant words included some that are unhelpful or spurious.

Table 13 shows the percentage classification accuracy results obtained using the different strategies. Because too many phrases were generated using DelSOcontGN and, in some cases, DelSOcontGW for the TFPC algorithm to operate, the results were incomplete for these algorithms, but, as can be seen, results obtained for DelSOcontGW were invariably poorer than for other strategies. In the other cases, it is apparent that better results were always obtained when significant words were distributed equally between classes (columns headed "Dist", noted as "top $\frac{K}{|C|}$" in section 4.2) rather than selecting only the K (1,500) most significant words. Best results were obtained with this policy using a potential significant word list made up all words with a contribution above the G threshold (columns headed "All"), rather than when using only those that were unique to one class. Overall, DelSNcontGO performed slightly better than DelSNcontGW,

Table 14. Number of empty documents in the training set (NGA.D10000.C10)

	Word Frequency				Word Support			
	Unique		All		Unique		All	
	Dist	Top K	Dist	Top K	Dist	Top K	Dist	Top K
DelSNcontGO	190	258	251	299	229	238	224	370
DelSNcontGW	190	226	251	299	229	147	224	370
DelSOcontGN								
DelSOcontGW			251		229	411	224	370
Keywords	190	226	251	299	229	411	224	370

and both phrase-generation strategies outperformed the Keywords-only algorithm. The contribution calculation mechanism used did not appear to make a significant difference to these results.

Table 14 shows the number of “empty” training set documents found in the different cases: that is, documents in which no significant attributes were identified. These represent between 2% and 5% of the total training set. Perhaps more importantly, any such documents in the test set will necessarily be assigned to the default classification. Although no obvious relationship between the frequency of empty documents and classification accuracy is apparent from these results, further investigation of this group of documents may provide further insight into the operation of the proposed strategies.

Table 15 shows execution times in seconds for the various algorithms, including both time to generate rules and time to classify the test set. The key words only approach is faster than DelSNcontGO because many fewer attributes are considered, so TFPC generates fewer frequent sets and rules. However, DelSNcontGW is fastest as the use of wild card leads to faster phrase matching.

Table 15. Execution times (NGA.D10000.C10)

	Word Frequency				Word Support			
	Unique		All		Unique		All	
	Dist	Top K	Dist	Top K	Dist	Top K	Dist	Top K
DelSNcontGO	244	250	253	242	250	248	328	235
DelSNcontGW	155	148	145	158	157	194	145	224
DelSOcontGN								
DelSOcontGW			370		326	281	278	314
Keywords	183	176	282	287	261	262	235	220

A further set of experiments were conducted to investigate the effects of adjusting the various thresholds. The first of these analysed the effect of changing G. The G value (contribution or siGnificance threshold) defines the minimum contribution that a potential significant word must have. The size of the potential significant word list thus increases with a corresponding decrease in G; conversely, we expect the quality of the words in the list to increase with G.

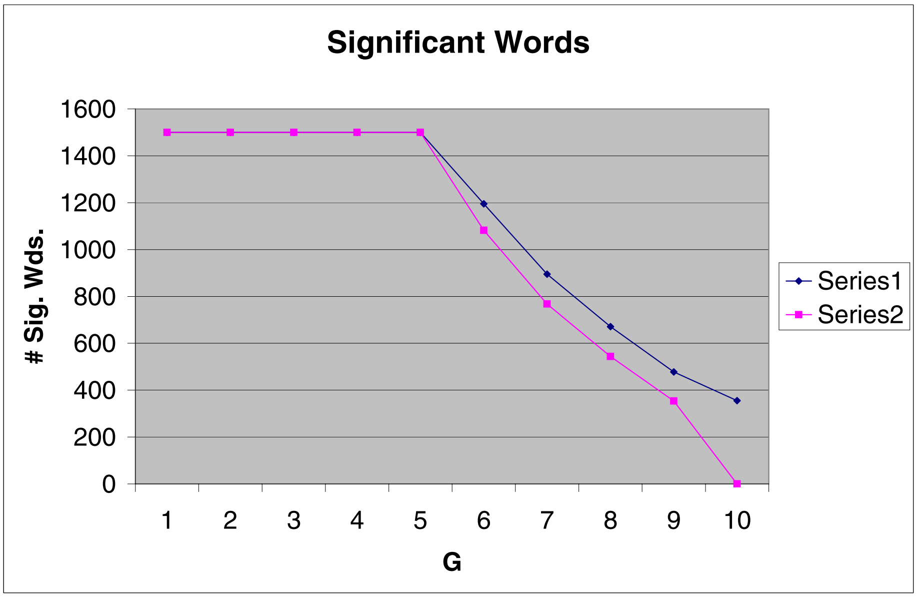

Fig. 1. Relationship between G value and number of significant words identified for NGA.D10000.C10, UNT = 7.0%, LNT = 0.2%, and K = 1,500. Series 1 = word frequency contribution calculation, Series 2 = word support contribution calculation

Figure 1 shows the effect on the number of selected significant words with changes in G, when UNT = 7.0%, LNT = 0.2%, and K = 1,500. The figure shows that there is little effect until the value of G reaches a point at which the size of the potential significant words list drops below K, when the number of selected significant words falls rapidly and a corresponding fall in accuracy is also experienced. The drop is slightly less severe using word frequency contribution calculation compared with support count contribution calculation.

In other experiments, varying the support and confidence thresholds had similar effects to those experienced generally in Association Rule Mining. Relatively low support and confidence thresholds are required because of the high variability of text documents, so as not to miss any significant frequent item sets or useful if imprecise rules. Generally we found that a support threshold corresponding to 10 documents produced best results, with a confidence threshold of 35.0%. We also undertook a number of experiments with the LNT and UNT thresholds. Best results were obtained using low values for both (such as those used in the above experiments).

7 Conclusion

In this paper we have described a number of different strategies for identifying phrases in document sets to be used in a "bag of phrases" representation for text classification. Phrases are generated using four different schemes to combine noise, ordinary and significant words. In all eight methods were used to identify significant words, leading overall to 32 different phrase generation strategies that were investigated, as well as 8 keyword only identification strategies.

The main findings of the experiments were:

1. Best results were obtained from a strategy that made use of words that were significant in one or more classes, rather than only those that were unique to one class, coupled with a selection strategy that produced an equal distribution between classes.
2. The most successful phrase based strategy outperformed classification using only keywords.

From the experiments described above we observe that a small subset of the documents to be classified were represented by an empty vector, i.e. they were not represented by any phrases/key words. This suggests that there remain possibilities to improve the strategies considered, which will be the subject of further investigation planned by the authors.

References

1. Ali, K., Manganaris, S., Srikant, R.: Partial classification using association rules. In: Heckerman, D., Mannila, H., Pregibon, D., Uthurusamy, R. (eds.) Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), pp. 115–118. AAAI press, Stanford (1997)
2. Coenen, F., Leng, P., Zhang, L.: Threshold tuning for improved classification association rule mining. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 216–225. Springer, Heidelberg (2005)
3. Coenen, F., Leng, P.: The effect of threshold values on association rule based classification accuracy. Journal of Data and Knowledge Engineering 60(2), 345–360 (2007)
4. Katrenko, S.: Textual data categorization: Back to the phrase-based representation. In: Proceedings of the Second IEEE International Conference on Intelligent Systems, vol. 3, pp. 64–67 (2004)
5. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the 12th International Conference on Machine Learning (ICML-95), pp. 331–339. Morgan Kaufmann, San Francisco (1995)
6. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Belkin, N.J., Ingwersen, P., Pejtersen, A.M. (eds.) Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-92), pp. 37–50. ACM Press, New York (1992)
7. Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: Cercone, N., Lin, T.Y., Wu, X. (eds.) Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM-01), pp. 369–376. IEEE Computer Society Press, Los Alamitos (2001)

8. Scott, S., Matwin, S.: Feature engineering for text classification. In: Bratko, I., Dzeroski, S. (eds.) Proceedings of the 16th International Conference on Machine Learning (ICML-99), pp. 279–388 (1999)
9. Sebastiani, F.: Machine learning in automated text categorization. ACM Computer Surveys 34(1), 1–47 (2002)
10. Sharma, R., Raman, S.: Phrase-based text representation for managing the web documents. In: Proceedings of the 2003 International Symposium on Information Technology (ITCC-03), pp. 165–169 (2003)
11. Sparck Jones, K.: Exhaustivity and specificity. Journal of Documentation 28, 11–21 (1972) (reprinted in 2004, 60, pp. 493–502)