# Taming Wild Phrases

C.H.A. Koster and M. Seutter

Dept. Comp. Sci.
University of Nijmegen
The Netherlands,
{kees, marcs}@cs.kun.nl

**ECIR'03**

**Abstract.** In this paper the suitability of different document representations for automatic document classification is compared, investigating a whole range of representations between bag-of-words and bag-of-phrases. We look at some of their statistical properties, and determine for each representation the optimal choice of classification parameters and the effect of Term Selection.
Phrases are represented by an abstraction called Head/Modifier pairs. Rather than just throwing phrases and keywords together, we start with pure HM pairs and gradually add more keywords to the document representation. We use the classification on keywords as the baseline, which we compare with the contribution of the pure HM pairs to classification accuracy, and the incremental contributions from heads and modifiers. Finally, we measure the accuracy achieved with all words and all HM pairs combined, which turns out to be only marginally above the baseline.
We conclude that even the most careful term selection cannot overcome the differences in Document Frequency between phrases and words, and propose the use of term clustering to make phrases more cooperative.

**Keywords**: syntactic phrases, Head/Modifier pairs, term selection, text categorization.

## 1 Introduction

*Anyhow, you've been warned and I will not be blamed*
*If your Wild Strawberry cannot be tamed..*
– Shel Silverstein, "A Light in the Attic", Harper & Row, 1981

Over the last decades, many researchers in Information Retrieval have tried to combine keywords with phrases, extracted from documents by linguistic or statistical techniques, in order to raise the accuracy of Retrieval (for an overview see [Strzalkowski, 1999]). Little is known about the best way to combine phrases and words in one language model, but the common approach followed in query-based Information Retrieval is to *add* the phrases to the words rather than to *replace* the words by phrases.

Just adding phrases (or collocations) as terms besides keywords has led to disappointingly small improvements [Fagan, 1988][Lewis and Croft, 1990]. This is commonly attributed to the fact that phrases have a distribution over documents which is very different from that of (key)words. Moreover it is obvious that a (composed) phrase is statistically correlated with its components, which may violate assumptions of statistical independence. At any rate, the improvements gained by using more precise terms (phrases) may well be offset by a loss in recall.

In this paper, we compare the suitability of different document representations for automatic document classification, investigating a whole range of representations between bag-of-words and bag-of-phrases. Being aware of the fact that different representations may need different classification parameters, we shall first determine the optimal tuning and Term Selection parameters for each representation.

Text categorization is a wonderful area for performing experiments in Information Retrieval: given the availability of large labeled corpora and the high performance of modern classification engines, it is a simple matter to measure the way in which the Accuracy achieved depends on the parameter settings and the choice of document representation. In traditional query-based Information Retrieval, doing such controlled experiments is much harder and costlier.

### 1.1 HM pairs

We are investigating the effect of using linguistically motivated terms (phrases) in Information Retrieval, particularly in Text Categorization. Following many earlier authors (e.g. [Fagan, 1988][Lewis and Croft, 1990][Strzalkowski, 1992] [Ruge, 1992][Evans and Lefferts, 1994][Lin, 1995]), these will be represented by *Head/Modifier pairs* (HM pairs) of the form

```
[ head, modifier]
```

where the head and the modifier are (possibly empty) strings of words, usually one word. A *pure HM pair* is one where head and modifier are not empty. There may also be HM pairs with an empty modifier (only a single head).

As an example, the phrase "new walking shoes" will first be transduced to the HM tree `[[shoes, walking], new]` and then unnested to one of the following:

- pure HM pairs

  `[shoes,walking] [shoes,new]`
- HM pairs including single heads

  `[shoes] [shoes,walking] [shoes,new]`
- idem plus single modifiers

  `[shoes] [shoes,walking] [shoes,new] [walking] [new]`

In distinction to many other researchers, we shall represent not only the Noun Phrase and its elements by HM pairs, but we shall also express the subject relation (as a Noun/Verb pair) and the object relation (a Verb/Noun pair). The effectiveness of these different representations will be compared in section 5.

### 1.2 The EP4IR parser/transducer

For generating the HM pairs, we made use of the EP4IR parser/transducer described in [Koster and Verbruggen, 2002], which is available under the GPL/LGPL license. It is generated from the EP4IR grammar and lexicon by means of the AGFL system[1].

Being especially directed towards IR applications, the EP4IR grammar does not set out to give a linguistically impeccable "account" of all English sentences, but it describes mainly the Noun Phrase (NP), including its adjuncts, and the various forms of the Verb Phrase (VP), consisting of the application of a certain verbal part to certain noun phrases (NP's) which occur as its complements. These phrases are transduced into HM pairs, in the process performing certain *syntactic and morphological normalizations*: elements of the phrase are selected, reordered and grouped. Furthermore, NP's not covered by a VP are also extracted.

The transformations are purely syntactic, i.e. they take no other information into account than the grammar, the lexicon and the input. In some cases this may result in linguistically suspect interpretations.

Precision and Recall of the EP4IR version used in the experiments are barely satisfactory, between .6 and .7 according to measurements. In interpreting the following experiments it should be kept in mind that the linguistic resources are not perfect – phrases are missed or mangled. But rather than waiting for perfect resources, we started experimenting with the available ones.

### 1.3 About the classification engine

The classification engine used in the experiments is the Linguistic Classification System LCS, developed for the PEKING project [2]. It implements two classification algorithms, Winnow [Dagan et al, 1997] and Rocchio [Rocchio, 1971], with a number of Term Selection algorithms and automatic Threshold Selection.

## 2 Statistics of phrases

According to the literature, the improvement in precision and/or recall obtained by using phrases as terms in retrieval and classification has repeatedly been found disappointing. There is a common feeling that "the statistics of phrases are wrong". In this section we shall compare the statistics of words and HM pairs in various ways.

A moment's thought gives support to the idea that the statistical distribution of HM pairs (pairs of keywords) is definitely different from that of the keywords themselves: according to a well-known folklore law in corpus linguistics, in any sufficiently long text, the number of words occurring precisely once (*hapaxes*) is about 40%; therefore the expected percentage of random pairs of words occurring precisely once is $1 - (1 - 0.4)^2 = 64\%$.

[1] `www.cs.kun.nl/agfl`

[2] `http://www.cs.kun.nl/peking`

### 2.1 The corpus

Our corpus, EPO1A, is a mono-classified corpus with 16 classes totalling $16 \times 1000$ abstracts of patent applications in English from the European Patent Office, see [Krier and Zaccà, 2001][Koster et al, 2001], with an average length of 143 words. From this corpus we used 4 subsets of 4000 documents each, chosen at random, in a four-fold cross-validation (training on each of the subsets while using the union of the other three as test set). The reasons for taking this unusual 25/75 split are:

- there is an abundance of labeled documents (1000/category) so that a 25% subset as train set is enough to reach stable classification
- 4-fold cross-validation is a reasonable compromise with efficiency
- testing is much faster than training and the large size of the test set reduces the variance.

The documents have been only lightly pre-processed: de-capitalization and elimination of certain characters from the keywords, but no lemmatization. For the HM pair representation they were completely parsed and the resulting trees unnested, also without lemmatization.

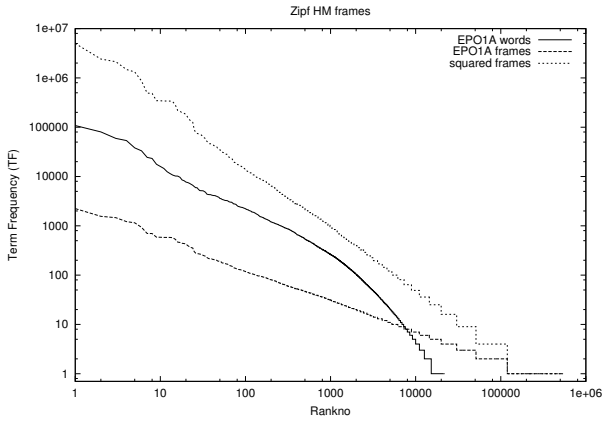
As a measure of the Accuracy, we take the micro-averaged F1-value (a harmonic average between Precision and Recall, combining the hits and misses of all categories before averaging). In the EPO1A corpus, all classes have the same size, so that the differences between macro- and micro-averaging are small.

### 2.2 Playing Zipf

Using the EP4IR parser/transducer, each of the EPO1A documents was parsed and transduced to a bag of unnested HM pairs. We omit the phrases with an empty modifier (i.e., consisting of only one word) to avoid all overlap with the bag of words. This provides us with bag of pairs representation of the same 16000 documents, with the following statistics:

| corpus id | total terms | different terms | total size (bytes) |
|---|---|---|---|
| EPO1A words | 2004011 | 21921 | 12069192 |
| EPO1A pairs | 921466 | 541642 | 20302454 |

It appears that the total number of HM pairs is about half the number of words, but there are 25 times as many *different* HM pairs. The average word frequency in EPO1A is about 9, the average HM pair frequency is about 2. The statistics of words and phrases are definitely different, which may well explain the bad experiences reported in literature. This also becomes clear from a comparison of the Zipf curves (frequency of words, pairs and, for comparison, squared frequency of pairs):

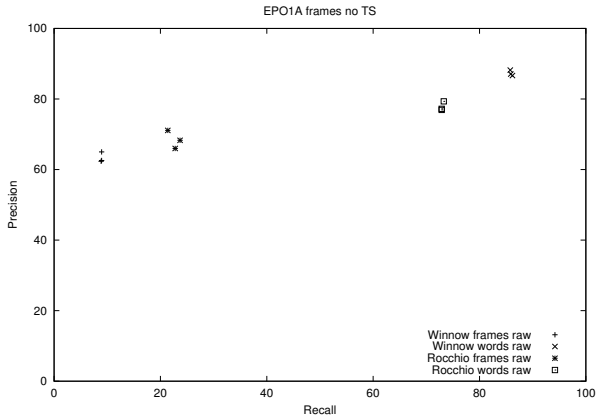



According to Zipf's law, on a log-log scale the relation between the frequency of a word in a corpus and its rank (ordering the words by frequency) looks like a straight line.

Compared to the graph for words, the graph for pairs is ramrod straight and much less steep. Representing a document by HM pairs gives an enormous number of low-frequency terms, among which the significant terms are cunningly hidden. Intuitively, some of the HM pairs must be much more indicative of a particular class than the keywords out of which they are composed. But the space of HM pairs is much larger than that of keywords, and therefore much more sparse.

### 2.3 The trouble with phrases

We first compare the effect of the two classification algorithms (Winnow and Rocchio) on phrases (represented as pure HM pairs) and keywords, using standard (default) parameters for the classification algorithms and without performing any term selection at all. We show three individual results (different 25% train sets) for each combination.



In comparison to the classification on phrases, words give not only a much higher Recall but even a higher precision. The naive use of phrases always leads to disappointing results. We must be doing something wrong.

## 3 Improving the statistics

The classification algorithms are subject to noise, because their work is based on term statistics, including very many irrelevant terms, and because of the imperfect labeling of the documents. When eliminating irrelevant terms by Term Selection, we expect not only increased performance but also increased Accuracy (see [Yiming and Pedersen, 1997] [Peters and Koster, 2002]).

In order to get the best Accuracy out of different document representations, we may also have to adapt some classification parameters to the representation. On the basis of extensive experiments, we found three parameter settings to be crucial:

- the Rocchio parameters Beta and Gamma
- the Winnow parameters for the Thick Threshold heuristic
- the choice of Term Selection and in particular the number of terms per class.

We found that by an optimal choice of these parameters, the Accuracy is remarkably improved, even for the baseline (keyword representation). In the following section we shall do the same analysis for the other representations. It will turn out that the optimal values are in fact not strongly dependent on the representation, but that they differ quite a lot from their usual values in literature.

### 3.1 Winnow and its parameters

The Balanced Winnow algorithm [Grove et al, 2001][Dagan et al, 1997] is a child of the Perceptron. For every class $c$ and for every term $t$ two weights $W_t^+$ and $W_t^-$ are kept. The score of a document $d$ for a class $c$ is computed as

$$SCORE(c,d) = \sum_{t \in d} (W_{t,c}^+ - W_{t,c}^-) \times s(t,d)$$

where $s(t,d)$ is the (ltc normalized) strength of the term $t$ in $d$. A document $d$ belongs to a class $c$ if $SCORE(c,d) > \theta$, where the threshold $\theta$ is usually taken to be 1.

Winnow learns multiplicatively, driven by mistakes, one document at a time: When a train document belonging to some class $c$ scores below $\theta$, the weights of its terms $t$ in $W_t^+$ are multiplied by a constant $\alpha > 1$ and those in $W_t^-$ multiplied by $\beta < 1$; and conversely for documents *not* belonging to $c$ which score above $\theta$. The default values for the Winnow parameters are (following [Dagan et al, 1997]) $\alpha = 1.1$ and $\beta = 0.9$.

### 3.2 Rocchio and its parameters

The Rocchio algorithm [Rocchio, 1971][Cohen and Singer, 1999] computes for each class $c$ a weight for each feature (another word for term) by

$$w(t,c) = max(0, \frac{\beta}{|D_c|} \sum_{d \in D_c} s(t,d) - \frac{\gamma}{|\overline{D}_c|} \sum_{d \in \overline{D}_c} s(t,d))$$

where [3]

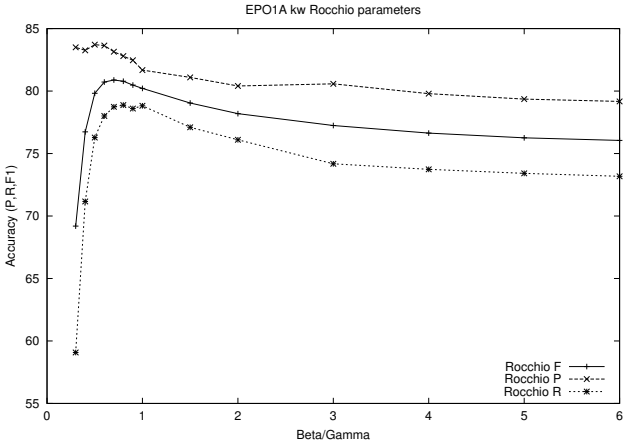
- $s(t,d)$ is the normalized strength of the term $t$ in the document $d$
- $D_c$ is the set of documents classified as $c$ and $\overline{D}_c$ the set of non-c documents.

The score of a document for a class is the inproduct of the weights of its features times their strength, and a document is assigned to class $c$ if its score for $c$ exceeds a class threshold which is computed from the train set (as is done for Winnow).

### 3.3 Tuning Rocchio

The Rocchio parameters $\beta$ and $\gamma$ control the relative contribution of the positive and negative examples to the weight vector; standard values in literature are $\beta = 16$ and $\gamma = 4$ [Cohen and Singer, 1999][Caropreso et al, 2000]. These values are rather puzzling, because only the *ratio* between $\beta$ and $\gamma$ is important for the outcome. We may fix one of the parameters arbitrarily at one without loosing generality.

In order to tune Rocchio to the base line (keywords), we determine experimentally its Accuracy as a function of the parameter $\beta$, keeping $\gamma = 1$ ($\beta = 4$ amounts to the traditional choice, [Arampatzis et al., 2000a] proposed $\beta = \gamma = 1$).




[3] In the experiments we used a variant of Rocchio in which the maximazation to 0 is not performed, thus allowing negative term contributions. Recently we found that the version with maximization can be tuned to even higher Accuracy.

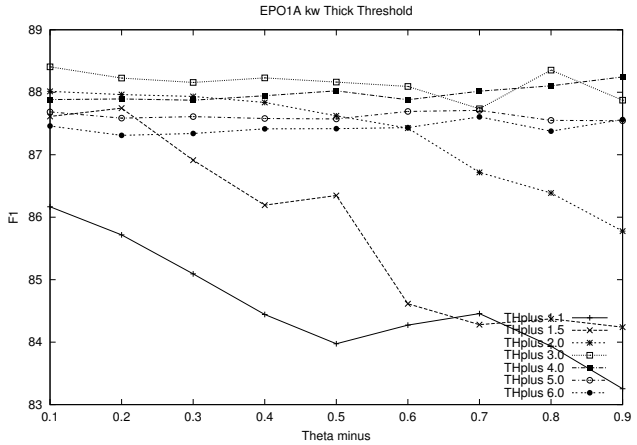
The graph shows that the optimum for $\gamma = 1$ is not at $\beta = 4$ but at $\beta = .7$, which means that negative contributions are favoured! Taking the optimal choice of $\beta$ raises the F1-value from .76 to .81, which is an important improvement.

### 3.4 Tuning Winnow

Phrases do not reoccur as often as words, because there are more of them; that is our main problem. The appearance of a phrase in a document may indicate that it belongs to a certain class, but its absence does not say much. Somehow, we must reward its presence stronger than we deplore its absence.

In the Winnow algorithm, this can be achieved by means of the *thick threshold* heuristic. In training, we try to force the score of relevant documents up above $\theta^+ > 1.0$ (rather than just 1) and irrelevant documents below $\theta^- < 1.0$. This resembles the "query zoning" heuristic for Rocchio, in the sense that documents on the borderline between classes get extra attention.

According to [Dagan et al, 1997] the optimal values for these Thick Threshold parameters are 1.1 and 0.9, respectively (just like the Winnow $\alpha$ and $\beta$). The following graphs show the effect of the thickness of the threshold for (key)words, our baseline.




The F1-value fluctuates wildly, but by and by an increase of $\theta^+$ improves the Accuracy, and [3.0,0.4] raises the F1-value over [1.1,0.9] by more than 2 points.

As was the case for Rocchio (see 3.3), it turns out that a non-traditional choice of parameters for Winnow may lead to a large improvement of the Accuracy. Obviously, the textbook values are far from optimal for every dataset!

### 3.5 Changing the statistics by Term Selection

Term selection is based on certain statistics of the terms, in particular their distribution over (documents belonging to) the various classes. We expect the classification process to be more accurate when two kinds of terms are eliminated [Peters and Koster, 2002]:

1. stiff terms – terms distributed evenly across documents of all categories, therefore occurring frequently. The traditional stop list is an attempt to eliminate on linguistic grounds the most frequent stiff terms.
2. noisy terms – terms distributed unreliably within a category and between classes. These often have a small frequency, but there are very many of them, causing *dispersion* of the document scores.

A good Term Selection criterion will remove both.

At the optimal values for the Rocchio and Winnow parameters, we apply Simplified $\chi^2$ (SX) as a local (i.e. category-dependent) Term Selection criterion in order to find the optimal number of terms per category (i.e. the number or rather range that maximizes Accuracy).

For our baseline, the keyword representation, term selection does not improve the Accuracy further, because the optimal choice of parameters apparently has the effect of removing most of the noisy terms. But for other representations we found that suitable term selection definitely raised the Accuracy (see later).

### 3.6 Summarizing the baseline

Here we summarize the best results obtained in classifying EPO1A using keywords.

| algorithm | method | max F1 value | parameter value |
|---|---|---|---|
| Winnow | raw | .83 | $\theta^+ = 1.1, \theta^- = 0.9$ |
| Winnow | tuned | .88 | $\theta^+ = 3.0, \theta^- = 0.4$ |
| Winnow | TSel | .88 | 1400-2000 terms/class |
| Rocchio | raw | .75 | $\beta$=4, $\gamma$=1 |
| Rocchio | tuned | .81 | $\beta = .7$, $\gamma$=1 |
| Rocchio | TSel | .81 | 100-1000 terms/class |

At the optimal parameter values, Term Selection makes hardly any improvement to the Accuracy, but the number of terms per class can be quite low without losing Accuracy.

## 4 Adding phrases to words

All authorities (e.g. [Fagan, 1988][Lewis and Croft, 1990][Strzalkowski, 1999]) agree that phrases should be used *besides* keywords, not *instead of* keywords. This is based on experience made with some form of phrases in query-based retrieval, where adding more precise terms to a query may always be beneficial. In document classification, the classification engine has to choose a subset of the terms available, not only for reasons of efficiency but also to optimize the Accuracy of the classification. It has to choose from a plethora of possible terms the most discriminative ones. What happens when we add to the tens of thousands of keywords many hundred thousands of phrases, all clamouring for attention? Is the Term Selection mechanism capable of coping with this riot? Will we need

many more terms per category? Or will the (low-frequency) phrases simply be ignored with respect to the much more frequent keywords?

After combining each document with the phrases extracted from it, we have repeated the experiment described above, and the results can be summarized as follows:

| algorithm | method | max F1 value | parameter value |
|---|---|---|---|
| Winnow | raw | .82 | $\theta^+ = 1.1, \theta^- = 0.9$ |
| Winnow | tuned | .88 | $\theta^+ = 3.0, \theta^- = 0.4$ |
| Winnow | TSel | .88 | 1400+ terms/class |
| Rocchio | raw | .78 | $\beta$=4, $\gamma$=1 |
| Rocchio | tuned | .82 | $\beta$ = .7, $\gamma$=1 |
| Rocchio | TSel | .83 | 100-1000 terms/class |

Rocchio performs a little bit better than for keywords alone, but the optimal choice of $\beta$ is the same. Winnow is not improved. Again, Term Selection makes no appreciable difference.

Inspection of the generated classifier shows that on the average only one HM pair is included among the top 40 terms, confirming our fear that the HM pairs are overwhelmed by the much more frequently occurring keywords.

## 5 Phrases instead of words

The easiest way to liberate the phrases from the aggressive keywords is to dispense with the keywords altogether and to use phrases instead of keywords. It also seems likely that only some well-chosen subset of the keywords should be used, in order to achieve optimal precision and recall.

In this section we shall compare the properties of a wide spectrum of text representations, ranging from pure HM pairs (bag-of-phrases representation) to all (key)words (bag-of-words representation).

The baseline which we want to exceed is remarkably high, due to the good statistical properties of the words in the text, even without lemmatization.

### 5.1 Pure HM pairs

Starting at the extreme end, we investigate the effect of using only "pure" HM pairs, with a nonempty modifier, corresponding to the composed phrases and the traditional collocations.

**Optimal choice of parameters** As was the case with the keyword representation, we have first determined the optimal Winnow and Rocchio parameters. At the traditional value of those parameters, the Accuracy was much lower than for the keyword representation. In particular, the Recall is much less than the Precision. Tuning the Winnow and Rocchio parameters in order to adapt them

to this different representation greatly improves the Accuracy (see the table in 5.1), but the optimal parameters values are practically the same as for keywords.

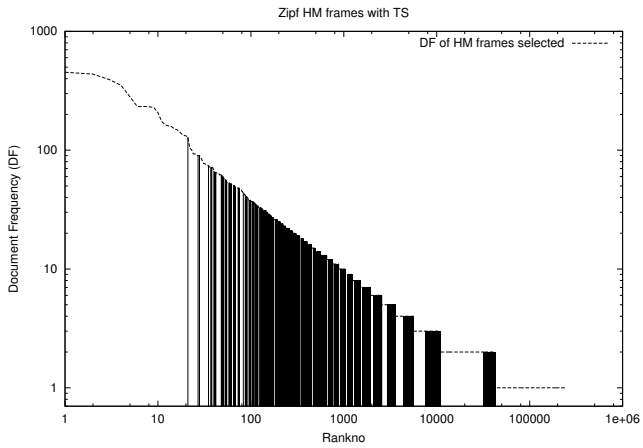
Term Selection has no positive effect for Rocchio, but Winnow with SX manages to raise the Accuracy by another 7%.

Using all 16000 documents as the train set (so that all terms are included) for Winnow with optimal parameters and Term Selection (first eliminating the hapaxes (MinTF=2) and then selecting 3000 terms per category), the following results show the effect of term selection on the number of terms:

```
541537 different terms in train set
119773 global terms in train set
30289 final terms in train set
```

It is clear that about 421000 of the terms are hapaxes. Of the remaining HM pairs about 90000 are eliminated by term selection. Only 30289 terms (instead of $16 \times 3000$) remain, because there is much overlap in the terms selected for the 16 classes.

The following graph shows for a typical example the selected terms with their Document Frequencies (DF); both at the high end and at the low end terms have been eliminated.




**Summary HM** Using only the "pure HM pairs, the Accuracy falls far short of the baseline, even with the best choice of parameters:

| algorithm | method | max F1 value | parameter value |
|---|---|---|---|
| Winnow | raw | .37 | $\theta^+ = 1.1, \theta^- = 0.9$ |
| Winnow | tuned | .56 | $\theta^+ = 3.0, \theta^- = 0.1$ |
| Winnow | TSel | .63 | 3000-4000 terms/class |
| Rocchio | raw | .40 | $\beta$=4, $\gamma$=1 |
| Rocchio | tuned | .57 | $\beta$ = .7, $\gamma$=1 |
| Rocchio | TSel | .57 | no term selection |

### 5.2 Phrases plus single heads

Not all phrases are composed. A non-composed phrase will have an empty modifier, which excludes it from the pure HM pairs. In this section we therefore add the single heads taken from the HM pairs as terms.

```
[shoes] [shoes,walking] [shoes,new]
```

Due to the prominent semantical role of heads, they are very promising classification terms, but we expect them to be less precise than pure HM pairs (since the modifiers were added to them to increase precision), and to have higher Term and Document Frequencies so that they might overwhelm the pure HM pairs.

Again we optimize the parameters and determine the Accuracy as a function of the number of terms per category, selected with the SX criterion. The effect of Term Selection is small, the optimum number of terms per class is in a broad band at a high number of terms, much higher than for keywords. It appears that many infrequent terms have to be combined in order to achieve good Precision and Recall.

Training and testing on seen documents again, we now obtain:

```
573222 different terms in train set
137451 global terms in train set
26546 final terms in train set
```

There are 32000 more terms to begin with, of which 14000 are eliminated as hapaxes. At the bottom line, fewer terms are selected. Inspection of the generated classifiers shows that only 4 HM pairs are included among the top 40 terms.

**Summary HM+H** Adding the heads to the pure HM pairs greatly improves the Accuracy, provided the parameters are well-chosen. The additional effect of Term Selection is small.

| algorithm | method | max F1 value | parameter value |
|---|---|---|---|
| Winnow | raw | .54 | $\theta^+ = 1.1, \theta^- = 0.9$ |
| Winnow | tuned | .77 | $\theta^+ = 3.0, \theta^- = 0.4$ |
| Winnow | TSel | .79 | 3000-7000 terms/class |
| Rocchio | raw | .55 | $\beta$=4, $\gamma$=1 |
| Rocchio | tuned | .715 | $\beta$ = .7, $\gamma$=1 |
| Rocchio | TSel | .72 | 5000-7000 terms/class |

### 5.3 Phrases plus heads and modifiers

In the next representation, we include besides the pure HM pairs and their heads also their modifiers, which we also expect to be be important keywords.

Using this representation, the example tree `[[shoes,walking],new]` is now unnested to

```
[shoes] [shoes,walking] [shoes,new] [walking] [new]
```

We expect this addition to be a mixed blessing: again the number of different terms is increased, and the heads and modifiers are certainly not statistically independent from the HM pair from which they are derived.

The effect of Term Selection on the number of terms is:

```
633412 different terms in train set
162758 global terms in train set
24650 final terms in train set
```

The modifiers add about 60000 modifiers new terms, of which only 25000 are not eliminated as hapaxes. The number of terms after term selection is reduced by about 2000, mostly replacing a number of HM pairs by one word. Indeed, the top 40 terms now contain only one HM pair.

**Summary HM+H+M** The Accuracy is now much improved by tuning and Term selection, and nearly as good as for keywords alone ...

| algorithm | method | max F1 value | parameter value |
|---|---|---|---|
| Winnow | raw | .59 | $\theta^+ = 1.1, \theta^- = 0.9$ |
| Winnow | tuned | .85 | $\theta^+ = 3.0, \theta^- = 0.5$ |
| Winnow | TSel | .855 | 3000-7000 terms/class |
| Rocchio | raw | .715 | $\beta$=4, $\gamma$=1 |
| Rocchio | tuned | .78 | $\beta$ = .7, $\gamma$=1 |
| Rocchio | TSel | .79 | 2000-7000 terms/class |

But what we have achieved looks more like a linguistic form of term selection than like the best way to use phrases as terms.

### 5.4 Conclusion

It is clear that we have not succeeded in domesticating the wild phrases. The experiments described here did not yield a document representation based on Head/Modifier pairs which gives better classification Accuracy than the traditional keywords, but it did give many surprizes.

The first surprise is that the optimal setting of the Winnow and Rocchio parameters are so far from the values given in literature. Our main result is that the choice of parameters is crucial. By themselves the optimal parameter settings for Winnow and Rocchio differ little from one representation to another, at least for the EPO1A corpus, but the parameters value quoted in literature are far from optimal.

The use of an appropriate Term Selection (Simplified ChiSquare) adds some further Accuracy (7% in the case of pure HM pairs and 0-2% with keywords added), which shows that Term Selection is an important issue when using HM pairs.

The Winnow algorithm again clearly outperforms the Rocchio algorithm, although before tuning and Term Selection Rocchio behaves slightly better than Winnow.

Compared to the use of all keywords as a baseline, adding phrases helps very little, because their Document Frequencies are so low that they get very little weight. The various ways to use phrases *instead of* keywords all give less Accuracy than the baseline. Even with the best Term Selection, pure HM pairs give the lowest Accuracy. Adding the single heads improves it, and adding the modifiers even more, closely approaching the baseline but still below it (but with this last representation very few HM pairs are actually selected).

This may mean that at least some of the best classification terms are not heads or modifiers, as found by the syntax analysis. Maybe they are not verbs, nouns or adjectives and therefore do not appear in the HM pairs.

The quality of the linguistic resources is another concern: the limited Precision and Recall of the HM pair extraction (presently between 60 and 70%) causes the system to miss about one third of them. The free resources used here need more work.

Text Categorization is still an area where Statistics wins over Linguistics. It profits less from the use of phrases than traditional Query-based Retrieval, because the latter involves human formulation of queries.

### 5.5 Outlook

*An expert is a man who has made all the mistakes, which can be made, in a very narrow field.*

– Niels Henrik David Bohr (1885-1962).

In spite of this, we are convinced that the use of phrases in Text Categorization merits further research.

Intuitively, a document yields very many highly precise phrases with a very low Document Frequency. We can try to improve term conflation by lemmatization and syntactical or semantical normalizations. Furthermore, many of the phrases are statistically and linguistically related, or at least not independent, as is the case for two HM pairs with the same head. We may perform some form of Term Clustering [Lewis and Croft, 1990] or fuzzy matching [Koster et al., 1999] in order to conflate terms that are not independent.

But most urgently, new language models capturing some of the richness of phrase structure must be found. In the present experiment, we have used HM pairs as monolithic terms, disregarding their internal structure. There is additional information to be found in the co-occurrence of heads with different modifiers, in particular when generalizing from HM pairs to complete HM trees of varying depth.

## References

[Arampatzis et al., 2000a] Avi Arampatzis, Jean Beney, C.H.A. Koster, Th.P. van der Weide, KUN on the TREC-9 Filtering Track: Incrementality, Decay, and Thresh-

old Optimization for Adaptive Filtering Systems. The Ninth Text REtrieval Conference (TREC-9), Gaithersburg, Maryland, November 13-16, 2000.

[Caropreso et al, 2000] M.F. Caropreso, S. Matwin and F. Sebastiani (2001), A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization, In: A. G. Chin (Ed.), *Text Databases and Document Management: Theory and Practice*, Idea Group Publishing, Hershey, US, pp. 78–102.

[Cohen and Singer, 1999] W.W. Cohen and Y. Singer (1999), Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems 13*, 1, 100-111.

[Dagan et al, 1997] I. Dagan, Y. Karov, D. Roth (1997), Mistake-Driven Learning in Text Categorization. In: *Proceedings of the Second Conference on Empirical Methods in NLP*, pp. 55-63.

[Evans and Lefferts, 1994] D. Evans and R.G. Lefferts (1994), Design and evaluation of the CLARIT-TREC-2 system. Proceedings TREC-2, NIST Special Publication 500-215, pp. 137-150.

[Fagan, 1988] J.L. Fagan (1988), *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods*, PhD Thesis, Cornell University.

[Grove et al, 2001] A. Grove, N. Littlestone, and D. Schuurmans (2001), General convergence results for linear discriminant updates. Machine Learning 43(3), pp. 173-210.

[Koster et al., 1999] C.H.A. Koster, C. Derksen, D. van de Ende and J. Potjer, Normalization and matching in the DORO system. Proceedings of IRSG'99, 10pp.

[Koster et al, 2001] C.H.A. Koster, M. Seutter and J. Beney (2001), Classifying Patent Applications with Winnow, Proceedings Benelearn 2001, Antwerpen, 8pp.

[Koster and Verbruggen, 2002] C.H.A. Koster and E. Verbruggen (2002), The AGFL Grammar Work Lab, Proceedings FREENIX/Usenix 2002, pp 13-18.

[Krier and Zaccà, 2001] M. Krier and F. Zaccà (2002), Automatic Categorisation Applications at the European Patent Office, World Patent Information 24, pp. 187-196, Elsevier Science Ltd.

[Lewis and Croft, 1990] Term Clustering of Syntactic Phrases (1990), Proceedings SIGIR 90, pp. 385-404.

[Lin, 1995] D. Lin (1995), A dependency-based method for evaluating broad-coverage parsers. *Proceedings IJCAI-95*, pp. 1420-1425.

[Peters and Koster, 2002] C. Peters and C.H.A. Koster (2002), Uncertainty-based Noise Reduction and Term Selection, Proceedings ECIR 2002, Springer LNCS 2291, pp 248-267.

[Rocchio, 1971] J.J. Rocchio (1971), Relevance feedback in Information Retrieval, In: Salton, G. (ed.), *The Smart Retrieval system - experiments in automatic document processing*, Prentice - Hall, Englewood Cliffs, NJ, pp 313-323.

[Ruge, 1992] G. Ruge (1992), Experiments on Linguistically Based Term Associations, Information Processing & management, 28(3), pp. 317-332.

[Strzalkowski, 1992] T. Strzalkowski (1992), TTP: A Fast and Robust Parser for Natural Language, In: Proceedings COLING '92, pp 198-204.

[Strzalkowski, 1999] T. Strzalkowski, editor (1999), *Natural Language Information Retrieval*, Kluwer Academic Publishers, ISBN 0-7923-5685-3.

[Yiming and Pedersen, 1997] Y. Yiming and J.P. Pedersen (1997), A Comparative Study on Feature Selection in Text Categorization. In: ICML 97, pp. 412-420.