

Statistical Phrases in Automated Text Categorization

Maria Fernanda Caropreso*

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Buenos Aires, Argentina
E-mail: mcaropre@dc.uba.ar

Stan Matwin

Dept. of Computer Science
University of Ottawa
K1N 6N5 Ottawa, Canada
E-mail: stan@site.uottawa.ca

Fabrizio Sebastiani

Istituto di Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche
56100 Pisa, Italy
E-mail: fabrizio@iei.pi.cnr.it

Abstract

In this work we investigate the usefulness of n -grams for document indexing in text categorization (TC). We call n -gram a set t_k of n word stems, and we say that t_k occurs in a document d_j when a sequence of words appears in d_j that, after stop word removal and stemming, consists exactly of the n stems in t_k , in some order. Previous researches have investigated the use of n -grams (or some variant of them) in the context of specific learning algorithms, and thus have not obtained general answers on their usefulness for TC. In this work we investigate the usefulness of n -grams in TC independently of any specific learning algorithm. We do so by applying feature selection to the pool of all α -grams ($\alpha \leq n$), and checking how many n -grams score high enough to be selected in the top σ α -grams. We report the results of our experiments, using several feature selection functions and varying values of σ , performed on the Reuters-21578 standard TC benchmark. We also report results of making actual use of the selected n -grams in the context of a linear classifier induced by means of the Rocchio method.

Categories and subject descriptors: H.3.3 [Information storage and retrieval]: Information search and retrieval - *Information filtering*; H.3.3 [Information storage and retrieval]: Systems and software - *Performance evaluation (efficiency and effectiveness)*; I.2.3 [Artificial Intelligence]: Learning - *Induction*

Terms: Algorithms, Experimentation, Theory

Keywords: Machine learning, text categorisation, text classification

* The work by this author was partially carried out while visiting the Department of Computer Science of the University of Ottawa, supported by Grant FOMEC376 from the Universidad de Buenos Aires.

1 Introduction

A key issue for *information retrieval* (IR) and all other content-based text management applications is *document indexing*, i.e. the task of automatically constructing an internal representation of a text d_j that (i) can be interpreted by the document management algorithms, and (ii) compactly captures the meaning of d_j . The choice of a representation format for text depends on what we consider to be (a) the meaningful textual units (the problem of *lexical semantics*) and (b) the meaningful natural language rules for the combination of the meanings that these units convey (the problem of *compositional semantics*). Traditionally, IR has concentrated on issue (a) and almost neglected issue (b), assuming that a good representation for a document d_j may be obtained by simply taking into account whether and how frequently a word t_k appears in d_j and in the document collection, thus disregarding the syntactic, semantic and pragmatic contexts of such occurrences. This has given rise to the so-called *bag of words* approach to indexing, according to which a text d_j is represented as a vector of weights $\langle w_{1j}, \dots, w_{rj} \rangle$, where r is the number of words that occur at least once in the document collection and $0 \leq w_{kj} \leq 1$ represents, loosely speaking, how much word t_k contributes to identifying the meaning of d_j . Weights $0 \leq w_{kj} \leq 1$ are computed according to the frequency of t_k in d_j and in the document collection. Variants of the bag of words approach are obtained by using *word stems* instead of words [9], or by disregarding frequency issues and simply using a binary assignment for w_{kj} based on the presence/absence of t_k in d_j (the *set of words* approach).

We will hereafter use the term *vector of features* to denote a vector of weighted words, or stems, or whatever characteristics of a document one might decide to use for the representation; accordingly, we will use variables t_1, t_2, \dots to denote features. Of course, the possible choices for what counts as a feature are limited by current text processing technology, i.e. by what can be extracted in a fully automated and scalable way from the text itself. That is, although in principle it would be best to identify features with the *concepts* the document deals with, or with the *problems* the document tackles, these pieces of knowledge are not within the reach of current knowledge extraction technology.

1.1 Phrase indexing in IR and TC

In the past a number of IR researchers have expressed their dissatisfaction with the bag (or set) of words approach, and have tried to use notions of what a feature is that are at the same time semantically richer and technically feasible. In particular, a number of authors have investigated *phrase indexing*, i.e. the use of *phrases*, in addition to individual words, as features. In a linguistic sense, a phrase is a textual unit usually larger than a word but smaller than a full sentence: examples of *noun phrases* are nuclear waste disposal, the dog that crossed the street, and Bill Clinton, while examples of *verb phrases* are playing ice hockey and went to school. Hereafter, we will use the term *syntactic phrase* to denote any phrase that is such according to a grammar of the language under consideration. Using syntactic phrases in indexing seems an interesting idea, in that

- phrases come closer than individual words or their stems to expressing structured concepts;
- phrases have a smaller degree of ambiguity than their constituent words, thanks to the *mutual disambiguation effect* of words. That is, while the two words hand and drill are both ambiguous (e.g. a hand of cards and shaking hands; oil drilling and a pronunciation drill), hand drill is not, since each of its two constituent words creates a context for the unambiguous interpretation of the other;
- by using phrases as index terms, a document that contains a phrase that occurs in the request would be ranked higher than a document that just contains its constituent words in unrelated contexts;
- current natural language processing technology allows the individuation of phrases to be performed with a good degree of robustness [29, 33].

Unfortunately, a number of researches that have investigated the usefulness of indexing with syntactic phrases in IR have obtained discouraging results (see Section 7). The likely reason for this is that, although indexing languages based on phrases have superior semantic qualities, they have inferior statistical qualities with respect to indexing languages based on single words [7, 18]. For instance, the phrase **nuclear waste disposal** definitely denotes an interesting, articulated concept, but unless it occurs frequently enough in the document collection it is unlikely to make an impact in terms of effectiveness. This situation is worsened by the fact that the same concept may be triggered by related but linguistically different units (such as **disposing of nuclear waste**, **Dispose of your nuclear waste!**, etc.), each of which is usually considered, from the standpoint of frequency, a different unit, unless the similarity of the underlying concepts is recognized.

Also, not every syntactic phrase denotes an interesting concept: **associate professor** does, but **tall professor** does not, and telling a phrase that does from one that does not is difficult (Kageura and Umno [15] call this “the termhood problem”).

Some researchers have attempted to find a way out of these problems by understanding the notion of phrase in a statistical sense, rather than syntactically. We will call *statistical phrase* any sequence of words that occur contiguously in a text, and do so in a statistically interesting way. Statistical phrases have a number of advantages over syntactic ones: a) they may be recognized by means of more robust and less computationally demanding algorithms; b) the effect of irrelevant syntactic variants can be factored out; and c) uninteresting phrases (e.g. **tall professor**) tend to be filtered out from interesting ones (e.g. **associate professor**). Of course, inherent in their statistical nature is the disadvantage of a non-null error rate: some phrases are not going to be recognized as such, and some non-phrases are instead going to be incorrectly recognized as phrases.

This work deals with assessing the value of statistical phrases for document indexing in the context of *text categorization* (TC), the activity of inductively learning to classify natural language texts with topical categories from a pre-specified set [28]. Previous researches have investigated the impact of statistical phrases on TC in the context of specific learning algorithms, and thus have not obtained general answers on their usefulness for TC *tout court*. In this work we want to analyze the problem in a learner-independent way, with the aim of obtaining an indication on the usefulness of statistical phrases for TC that be independent of the learning algorithm to be used. In order to do so, we extract phrases from a corpus of documents and assess their value not in a “direct” way (i.e. by running classification experiments on a test collection) but in an “indirect” way, i.e. by scoring the phrases by means of a number of different *feature evaluation functions* [32]. The extent to which phrases outplay (according to the computed scores) single words, and the increase that phrases bring about in the overall “score” of the resulting indexing language, will indicate the potential usefulness of phrases in TC.

Following this learner-independent study, we also perform a number of “direct” experiments by running the Rocchio classifier-learning algorithm on the phrase-based representations (we have started running a parallel experiment using the RIPPER system [3] but its results were not ready before submission time). These experiments are aimed at assessing whether the results from the “indirect” experiments are confirmed by field tests, i.e. whether an increase in the overall quality of the indexing language as measured by the above-mentioned criteria also results in an increase in classification effectiveness.

The paper is organized as follows. In Section 2 we briefly introduce the basic notions of text categorization. In Section 3 we define precisely our own notion of statistical phrase, that we will call *n-gram*¹. In Section 4 we describe our learner-independent method for the evaluation of *n*-grams. Section 5 describes the results we have obtained by applying this method on Reuters-21578, the standard benchmark of TC research. In Section 6 we discuss our “direct” experiments performed by applying the Rocchio classifier-inducing method to our *n*-gram representations, again using Reuters-21578. Section 7 describes some related work in phrase indexing in IR and TC. Section 8

¹We remark that the term “*n*-gram” is used in the text processing literature in two quite different senses. In the first sense it is used, as here, to indicate a *set of n words* that occur sequentially in a text. In the second sense it is used to indicate a *sequence of n characters* that occurs in a text, and that may be part of a word or of a sequence of two or more words occurring contiguously. The latter sense is typical of the literature on indexing noisy texts, such as those resulting from OCR, or texts in Asian languages, and will not be dealt with here.

concludes.

2 Text categorization

Text categorization (also known as *text classification*, or *topic spotting*) is the activity of automatically building, by means of machine learning (ML) techniques, *automatic text classifiers*, i.e. programs capable of labelling natural language texts with thematic categories from a predefined set $C = \{c_1, \dots, c_m\}$. A frequently used approach to building a text classifier for categories $C = \{c_1, \dots, c_m\}$ is that of building m independent classifiers, each capable of deciding whether a given document d_j should or should not be classified under category c_i , for $i \in \{1, \dots, m\}$ ². This process requires the availability of a corpus $Co = \{d'_1, \dots, d'_s\}$ of manually preclassified documents³, i.e. documents such that for all $i \in \{1, \dots, m\}$ and for all $j \in \{1, \dots, s\}$ it is known whether $d'_j \in c_i$ or not. A general inductive process (called the *learner*) automatically builds a classifier for category c_i by learning the characteristics of c_i from a *training set* $Tr = \{d'_1, \dots, d'_g\} \subset Co$ of documents. Once a classifier has been built, its effectiveness (i.e. its capability to take the right categorization decisions) may be tested by applying it to the *test set* $Te = \{d'_{g+1}, \dots, d'_s\} = Co - Tr$ and checking the degree of correspondence between the decisions of the automatic classifier and those encoded in the corpus.

2.1 Feature selection

Many classifier induction methods are computationally hard, and their computational cost is a function of the length of the vectors that represent the documents. It is thus of key importance to be able to work with vectors shorter than r (the number of words that occur at least once in the document collection), which is usually a number in the tens of thousands or more. For this, *feature selection* techniques are used to select, from the original set of r features, a subset of $r' \ll r$ features that are most useful for compactly representing the meaning of the documents; the value $\rho = \frac{r-r'}{r}$ is called the *reduction factor*. Usually, these techniques consist in scoring each feature by means of a *feature evaluation function* (FEF) and then selecting the r' features with the highest score. Often, feature selection is also beneficial in that it tends to reduce *overfitting*, i.e. the phenomenon by which a classifier tends to be better at classifying the data it has been trained on than at classifying other data.

Many functions, mostly from the tradition of decision or information theory, have been used as FEFs in TC [17, 22, 32]; some which are of interest to the present work are illustrated in Table 1. In the third column of this table, probabilities are interpreted on an event space of documents (e.g. $P(\bar{t}_k, c_i)$ indicates the probability that, for a random document x , feature t_k does not occur in x and x belongs to category c_i), and are estimated by counting occurrences in the training set. In Table 1 the χ^2 and *OR* functions are specified “locally” to a specific category c_i ; in order to assess the value of a feature t_k in a “global”, category-independent sense, either their weighted average $f_{avg}(t_k) = \sum_{i=1}^m P(c_i) f(t_k, c_i)$ or the maximum $f_{max}(t_k) = \max_{i=1}^m f(t_k, c_i)$ of their category-specific values are usually computed.

3 A definition of n -grams

We start by precisely characterizing what we mean by statistical phrases. The same definition has been used in a number of IR contexts (e.g. [2, 21]), but never in the case of TC (see Section 7 for a detailed discussion).

²In this paper we make the general assumption that a document d_j can in principle belong to zero, one or many of the categories in C ; this assumption is indeed verified in the Reuters-21578 benchmark we use for our experiments. All the techniques we discuss in this paper can be straightforwardly adapted to the other case in which each document belongs to exactly one category.

³In the following we use variables d_1, d_2, \dots to indicate generic documents and variables d'_1, d'_2, \dots to indicate preclassified documents.

Function	Denoted by	Mathematical form
<i>Document Frequency</i>	$DF(t_k)$	$P(t_k)$
<i>Information Gain</i>	$IG(t_k)$	$\sum_{t \in \{t_k, \bar{t}_k\}} \sum_{i=1}^m P(t, c_i) \log \frac{P(t, c_i)}{P(c_i) \cdot P(t)}$
<i>Chi-square</i>	$\chi^2(t_k, c_i)$	$\frac{g \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]^2}{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}$
<i>Odds Ratio</i>	$OR(t_k, c_i)$	$\frac{P(t_k c_i) \cdot (1 - P(t_k \bar{c}_i))}{(1 - P(t_k c_i)) \cdot P(t_k \bar{c}_i)}$

Table 1: Some feature evaluation functions used in the literature. In the $\chi^2(t_k, c_i)$ formula g is the cardinality of the training set.

Definition 1 A *1-gram* (or *unigram*) is a word stem. An *n-gram* is an alphabetically ordered sequence t_k of n unigrams. We say that an *n-gram* t_k occurs in a document d_j when a sequence of words appears in d_j that, after stop word removal and stemming, consists of a permutation of t_k .

For instance, *inform retriev* is a 2-gram (or *bigram*); among its possible occurrences in a text are the expressions

- (a) information retrieval
- (b) retrieval of information
- (c)* informative retrieval
- (d) retrieved information
- (e) retrieving information
- (f) retrieves information
- (g) Retrieve information!
- (h)* He informs the retriever

Note that, as evident from all these examples, stop word removal, stemming, and alphabetical ordering have the effect of factoring out from the notion of *n-gram* a number of morphological, syntactic, and semantic variations. As for morphosyntactic variations, note that *noun phrases* (expressions (a) to (d)), *verb phrases* (expressions (e) and (f)) and *full sentences* (expressions (g) and (h)) can all be occurrences of the same *n-gram*. As for semantic variations, note that noun phrases with different meanings, as is the case for (a) and (d), can also give rise to the same *n-gram*. Defining *n-grams* this way is based on the hypothesis that various syntactic expressions may convey the same concept, and is thus to be seen as a form of *conflation*. As for other types of conflation, the generalization we perform by means of *n-grams* has its problems too. In particular, *n-grams* as defined here suffer from

- *over-generalization*: this may be seen from the fact that examples (c) and (h) do not refer to the same concept as the other examples;
- *under-generalization*: this may be seen from the fact that an expression such as *retrieving interesting information* arguably refers to the same concept as example (e) but is not recognized as such.

Note also that, quite obviously, the mere contiguous occurrence of two words in a text does not guarantee that they refer to a complex concept. For instance, the text

What is recursion? It is what was illustrated in the dialogue *Little Harmonic Labyrinth*: nesting, and variations on nesting.

contains the bigrams *illustrat recur*, *dialog illustrat*, *dialog harmon*, *harmon labyrinth*, *labyrinth nest*, and *nest var*. Arguably, none of these conveys an articulated concept. It is then clear that the use of n -grams for indexing purposes is possible only in the presence of a method for filtering interesting n -grams from non-interesting ones.

Filtering is also necessary because the number of different n -grams that occur at least once in a collection is too high. In fact, while the number of α -gram occurrences increases linearly (for any occurrence of a α -gram there are 2 occurrences of a $(\alpha + 1)$ -gram), the number of *different* α -grams increases much more, since the average $(\alpha + 1)$ -gram occurs much less frequently than the average α -gram.

There are many possible filters, most of which are based on frequency considerations. This is not surprising, since we may expect an interesting bigram such as *inform retriev* to have different occurrence patters from an uninteresting, “occasional” bigram such as *illustrat recur*.

4 A classifier-independent evaluation of the usefulness of statistical phrases in text categorization

Our method of establishing the usefulness of n -grams for TC purposes consists in generating all α -grams (for $\alpha = 1, \dots, n$) that occur in a corpus of documents, score each of them by means of a FEF of the type discussed in Section 2.1, and rank them according to the score received. The proportion of n -grams that appear at the top of this ranked list will then constitute an indication of the potential usefulness of n -grams for text categorization.

In order to be more precise we introduce the notion of *penetration level* of n -grams.

Definition 2 *Let Tr be a training set of documents and r be the number of different unigrams that occur in it. We define the penetration level $\pi_\rho^f(n)$ of n -grams for FEF f at reduction factor ρ as the fraction of the $r' = r(1 - \rho)$ top (according to f) α -grams ($\alpha = 1, \dots, n$) of Tr for which $\alpha = n$.*

The purpose of this definition is best described by an example. Suppose that there are 10,000 different unigrams in our training set Tr . If we had to perform feature selection by applying a FEF f to these 10,000 unigrams with reduction factor .90, we would obtain the 1,000 unigrams that f considers the most valuable. Suppose that there are 120,000 different bigrams in Tr . In order to compute the penetration level $\pi_{.90}^f(2)$ we apply f to each of the 130,000 α -grams ($\alpha = 1, 2$) and check how many of the top 1,000 α -grams are actually bigrams. The higher $\pi_{.90}^f(2)$ is, the more valuable bigrams prospectively look, and the more worthwhile it looks to extract them. Or, at least, worthwhile according to our chosen FEF f and for the reduction factor ρ chosen. If we repeat the same experiment for different FEFs f_i and different reduction factors ρ_j , by averaging the results in some way we can get a fairly clear picture of how promising bigrams look for TC purposes, and we do so without invoking even a single learning algorithm, which means that our results are arguably going to be valid regardless of the specific learning algorithm chosen. This method is, of course, applicable for any value of n .

4.1 Pros and cons of this approach

Before moving to the discussion of the experimental results we have obtained, we should remark that this is not the only approach to the evaluation of n -grams for TC. A possible alternative approach consists in generating only a subset of prospectively good n -grams (i.e. n -grams selected according to a particular statistical filter [2, 4, 21] or heuristics [6, 11, 23]), using them in document indexing, and checking the difference in effectiveness that a given classifier exhibits with respect to the standard “bag of words” case.

This latter method has no doubt the advantage of a better computational efficiency; for instance, a heuristics according to which we generate all and only the n -grams that are composed of “valuable” unigrams and/or have certain frequency characteristics, allows to substantially reduce

the computation time needed to generate the n -grams, and completely avoids the computation time needed to score them. For many practical applications this may even be the only feasible method.

The drawback of this method, though, is that the experimental results thus obtained are going to be dependent (i) on the chosen heuristics and (ii) on the chosen classifier learning algorithm. The method we have chosen abstracts away from both aspects. While Point (ii) needs no further discussion, concerning Point (i) we want to emphasize that

1. our method relies not on generic heuristics, but on FEFs that are both well-studied and well-founded on statistical and information theory;
2. our method relies on the application of a whole range of FEFs, so as to obtain results that are not biased towards one or the other FEF.

In a sense, the real object of this work is *not* using n -grams in a particular TC application, and hence devising an efficient algorithm for extracting them. This work is more foundational in nature, as we want instead to assess whether, in principle, n -grams are prospectively interesting for TC applications so that it might be worth to devise such an algorithm. For this purpose, it is clear that we need to analyze *all* n -grams, and not just those that are generated by a selective heuristics. For the same reason, we need to perform this analysis in the most general possible way, that is, without reference to specific learning algorithms and with reference to the widest possible spectrum of FEFs.

5 “Indirect” experiments

We have performed a number of experiments in order to test the usefulness of n -grams for TC according to the above-mentioned learner-independent method. The experiments reported in this paper are limited to the case of $n = 2$, since the investigation of the $n \geq 3$ case might be worthwhile only after bigrams have unequivocally proven useful.

5.1 Experimental setting

For our experiments we have used the “Reuters-21578, Distribution 1.0” corpus, as it is currently the most widely used benchmark in text categorization research⁴. Reuters-21578 consists of a set of 12,902 news stories, partitioned (according to the “ModApté” split we have adopted) into a training set of 9,603 documents and a test set of 3,299 documents. The documents have an average length of 211 words (that become 117 after stop word removal) and are labelled by 118 categories; the average number of categories per document is 1.08, ranging from a minimum of 0 to a maximum of 16. The number of positive examples per category ranges from a minimum of 1 to a maximum of 3964. According to Definition 1, Reuters-21578 contains 17,439 unigrams and 250,059 bigrams, for a total of 267,498 “uni+bigrams”.

We have run our experiments on the set of 115 categories with at least 1 training example, rather than on other smaller, more commonly used subsets of it. The full set of 115 categories is “harder”, since it includes categories with very few positive instances for which inducing reliable classifiers is obviously a haphazard task⁵.

In all the experiments discussed in this paper, stop words have been removed using the stop list provided in [19, pages 117–118]. Punctuation has been removed and all letters have been converted to lowercase; no stemming and number removal have been performed.

⁴The Reuters-21578 corpus may be freely downloaded for experimentation purposes from <http://www.research.att.com/~lewis/reuters21578.html>

⁵See [13] for a discussion on why this is the “right” subset of Reuters-21578 categories to use.

5.2 Experimental results

Figure 1 displays the results of computing penetration levels for bigrams by applying the four FEFs described in Table 1 with varying reduction factors; χ^2 and OR have been tested in both their *avg* and *max* variants. We have chosen these FEFs as they have turned out to be the best performers in the thorough comparative experiments of [22, 32].

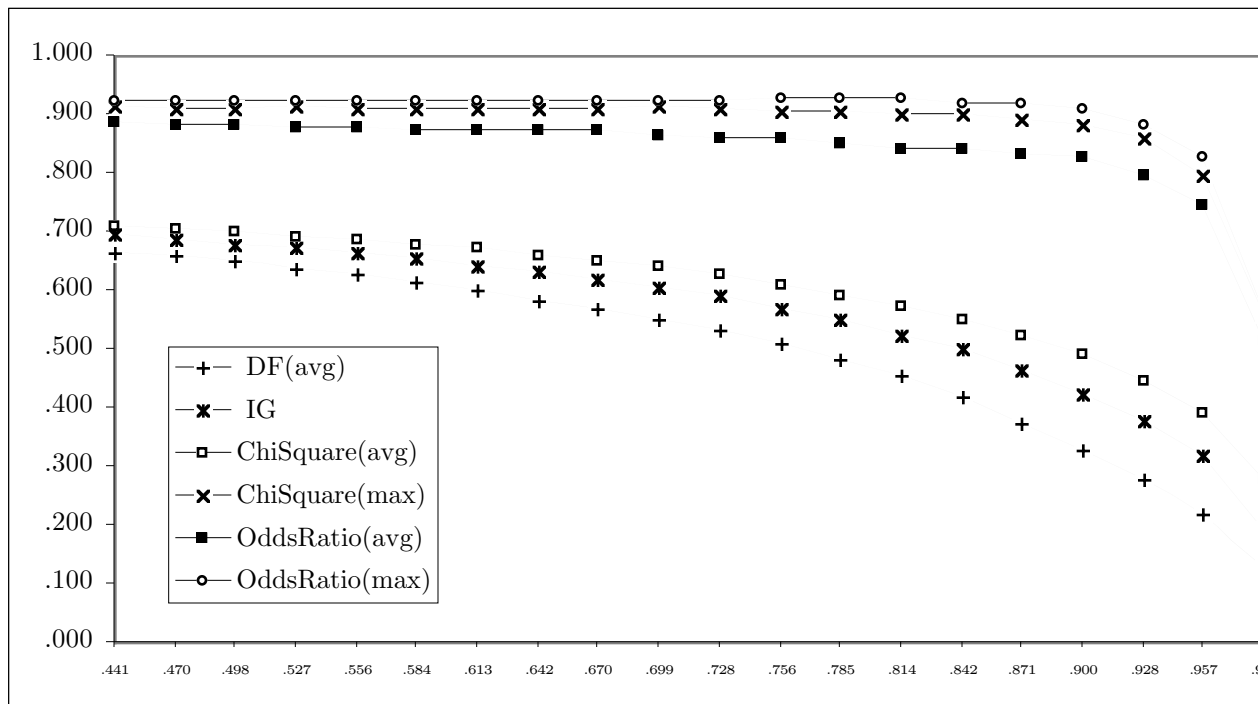


Figure 1: Penetration level for 2-grams computed for different FEFs at different reduction factors.

Table 2 lists, for a set of features φ_ρ^f selected by FEF f with reduction factor ρ ,

1. the values $AvgScore(\varphi_\rho^f)$, representing the average score attributed by f to a feature $t_k \in \varphi_\rho^f$;
2. the values $AvgDF(\varphi_\rho^f)$, representing the average number of documents in which a feature $t_k \in \varphi_\rho^f$ occurs;
3. the values $AvgCF(\varphi_\rho^f)$, representing the average number of categories that contain at least a document in which a feature $t_k \in \varphi_\rho^f$ occurs.

Each entry of Table 2 includes the value for the case of unigrams (upper row), for the case of uni+bigrams (lower row), and the increase obtained in switching from the former case to the latter. In order to correctly interpret the results, note that in each entry the values for the unigrams and the uni+bigrams cases are obtained by using *the same number of features*. For instance, to interpret the first row of results one should note that reducing the set of 17,439 unigrams by a $\rho = .70$ reduction factor yields 5,232 features, which is the same number of features obtained by reducing the set of 267,498 uni+bigrams by a $\rho = .9805$ reduction factor (this is why we will often speak the $.70/.9805$ reduction factor).

From Figure 1 it is immediately evident that, for all FEFs f , the penetration level $\pi_\rho^f(2)$ is a decreasing function of the reduction factor ρ . This is not surprising. In fact, suppose U_x and

FEF	ρ	$AvgScore(\varphi_\rho^f)$	$AvgDF(\varphi_\rho^f)$	$AvgCF(\varphi_\rho^f)$
<i>DF</i>	.70	80.393	80.393	14.751
<i>DF</i>	.9805	118.498 (+47.4%)	118.498 (+47.4%)	19.937 (+35.2%)
<i>DF</i>	.80	116.861	116.861	19.783
<i>DF</i>	.9870	164.113 (+40.4%)	164.113 (+40.4%)	24.247 (+22.6%)
<i>DF</i>	.90	214.195	214.195	30.096
<i>DF</i>	.9935	281.220 (+31.3%)	281.220 (+31.3%)	32.795 (+9.0%)
<i>IG</i>	.70	-2.907	80.037	14.653
<i>IG</i>	.9805	-2.903 (-0.1%)	114.478 (+43.0%)	18.873 (+28.8%)
<i>IG</i>	.80	-2.904	115.996	19.506
<i>IG</i>	.9870	-2.900 (-0.2%)	157.681 (+35.9%)	22.345 (+14.6%)
<i>IG</i>	.90	-2.898	210.065	28.886
<i>IG</i>	.9935	-2.893 (-0.2%)	264.287 (+25.8%)	29.073 (+0.6%)
χ_{avg}^2	.70	12.516	78.922	13.759
χ_{avg}^2	.9805	23.373 (+86.7%)	109.367 (+38.6%)	15.476 (+12.5%)
χ_{avg}^2	.80	17.521	113.975	18.161
χ_{avg}^2	.9870	31.339 (+78.9%)	150.118 (+31.7%)	18.646 (+2.7%)
χ_{avg}^2	.90	30.559	204.378	26.747
χ_{avg}^2	.9935	51.497 (+68.5%)	251.638 (+23.1%)	24.763 (-7.4%)
χ_{max}^2	.70	323.592	63.427	10.028
χ_{max}^2	.9805	1805.572 (+458.0%)	14.025 (-77.9%)	3.062 (-69.5%)
χ_{max}^2	.80	441.239	73.628	10.141
χ_{max}^2	.9870	2183.591 (+394.9%)	14.884 (-79.8%)	2.999 (-70.4%)
χ_{max}^2	.90	713.364	73.645	9.212
χ_{max}^2	.9935	2936.242 (+311.6%)	16.622 (-77.4%)	2.631 (-71.4%)
<i>OR_{avg}</i>	.70	2.980	22.825	3.321
<i>OR_{avg}</i>	.9805	8.257 (+177.0%)	16.919 (-25.9%)	1.961 (-40.9%)
<i>OR_{avg}</i>	.80	3.695	17.923	2.915
<i>OR_{avg}</i>	.9870	10.801 (+192.3%)	22.373 (+24.8%)	2.056 (-29.5%)
<i>OR_{avg}</i>	.90	5.506	24.107	2.990
<i>OR_{avg}</i>	.9935	17.721 (+221.9%)	34.618 (+43.6%)	2.052 (-31.4%)
<i>OR_{max}</i>	.70	411.681	18.826	6.498
<i>OR_{max}</i>	.9805	4003.307 (+872.4%)	5.339 (-71.6%)	2.799 (-56.9%)
<i>OR_{max}</i>	.80	575.073	15.278	5.823
<i>OR_{max}</i>	.9870	5217.889 (+907.3%)	5.113 (-66.5%)	2.464 (-57.7%)
<i>OR_{max}</i>	.90	982.787	12.660	5.062
<i>OR_{max}</i>	.9935	7583.538 (+671.6%)	3.812 (-69.9%)	1.963 (-61.2%)

Table 2: Values for $AvgScore(\varphi_\rho^f)$, $AvgDF(\varphi_\rho^f)$ and $AvgCF(\varphi_\rho^f)$ computed for various FEFs f at different reduction factors ρ . Every entry lists the score for the unigrams case (upper row), for the uni+bigrams case (lower row), and the percentage increase between the former and the latter.

B_x are the numbers of unigrams and bigrams still available for selection after the x top-scoring features have been selected, and suppose features are selected at random from $U_x \cup B_x$. If a unigram is selected as the $(x + 1)$ -th feature, this causes a decrease in the odds for U_x (denoted as $\frac{P(U_x)}{P(B_x)}$) much larger in magnitude than the increase in the same odds caused by the selection of a bigram, since U_x is much smaller than B_x . This means that, on average, these odds tend to decrease with x , which means that the proportion of bigrams in the set of the top x features tends to grow with x .

Also, from Figure 1 it is evident that the six FEFs we have studied may be partitioned in two groups (Group 1, consisting of $\{DF, IG, \chi_{avg}^2\}$, and Group 2, consisting of $\{\chi_{max}^2, OR_{avg}, OR_{max}\}$) of FEFs that display a very similar behaviour, with the FEFs of Group 2 displaying much higher penetration levels than those of Group 1. Incidentally, this also confirms one results of Yang and Pedersen [32], who in an experiment involving unigrams only and two different collections had shown DF and IG to be highly correlated, and had conjectured that this pattern was general rather than corpus-dependent.

The third observation is that penetration levels are indeed high! If we define bigrams as in Definition 2, many of them have statistical characteristics that, according to the FEFs we have employed, make them preferable to many of the unigrams rated high by the same FEFs. This means that, if we trust that our FEFs give a faithful picture of the value of a feature, it looks that bigrams may substantially improve the overall value of the indexing language used.

The results listed in the first column of Table 2 clearly show that the FEFs that achieve high penetration levels (see Figure 1) also achieve a high increase in $AvgScore(\varphi_\rho^f)$. This would seem to confirm that penetration levels are indeed a reasonable way to compute the contribution of n -grams to the overall quality of a feature set.

6 “Direct” experiments

Following this learner-independent study, we have performed a number of “direct” experiments aimed at assessing whether an increase in the overall quality of the indexing language, as measured by the above-mentioned criteria, also results in an increase in classification effectiveness.

6.1 Evaluation methodology

In the experiments that follow, classification effectiveness has been measured in terms of the classic IR notions of precision (Pr) and recall (Re) adapted to the case of document categorization. *Precision wrt c_i* (Pr_i) is defined as the probability that if a random document d_x is categorized under c_i (i.e. it is deemed a *positive* example of c_i), this decision is correct (i.e. it is a *true* positive for c_i). In what follows, TP , TN , FP and FN will denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. *Recall wrt c_i* (Re_i) is instead defined as the probability that, if a random document d_x ought to be categorized under c_i , this decision is taken. Estimates of Pr_i and Re_i (indicated by \hat{Pr}_i and \hat{Re}_i) may be obtained in the obvious way by counting occurrences on the test set. These category-relative values may in turn be averaged to obtain \hat{Pr} and \hat{Re} , i.e. values global to the whole category set C , according to two alternative methods:

- *microaveraging* (indicated by the “ μ ” superscript): \hat{Pr} and \hat{Re} are obtained by globally summing over all individual decisions, i.e.:

$$\begin{aligned}\hat{Pr}^\mu &= \frac{TP}{TP + FP} = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m (TP_i + FP_i)} \\ \hat{Re}^\mu &= \frac{TP}{TP + FN} = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m (TP_i + FN_i)}\end{aligned}$$

- *macroaveraging* (indicated by the “M” superscript): precision and recall are first evaluated “locally” for each category, and then “globally” by averaging over the results of the different categories, i.e.:

$$\hat{Pr}^M = \frac{\sum_{i=1}^m Pr_i}{m} = \frac{\sum_{i=1}^m \frac{TP_i}{TP_i + FP_i}}{m}$$

$$\hat{Re}^M = \frac{\sum_{i=1}^m Re_i}{m} = \frac{\sum_{i=1}^m \frac{TP_i}{TP_i + FN_i}}{m}$$

Whether microaveraging or macroaveraging is the “right” measure to use depends on the application. In our experiments we have thus evaluated both.

As a measure of effectiveness that combines the contributions of both \hat{Pr} and \hat{Re} , we have used the well-known F_β function, defined as

$$F_\beta = \frac{(\beta^2 + 1) \cdot \hat{Pr} \cdot \hat{Re}}{\beta^2 \cdot \hat{Pr} + \hat{Re}}$$

with $0 \leq \beta \leq +\infty$. Similarly to most other researchers we have used the parameter value $\beta = 1$, which places equal emphasis on \hat{Pr} and \hat{Re} .

6.2 Experimental results

Table 3 compares the effectiveness of unigrams and uni+bigrams on a linear classifier induced according to the Rocchio method, for the four FEFs of Table 1 and for different reduction factors. The Rocchio parameters have been set to $\beta = 16$ and $\gamma = 4$ (see [28, Section 6.6] for a full discussion of the Rocchio method). Feature weighting has been done by means of the standard “ltc” variant of the *tfidf* function, i.e.

$$tfidf(t_k, d_j) = tf(t_k, d_j) \cdot \log \frac{|Tr|}{\#_{Tr}(t_k)}$$

where $\#_{Tr}(t_k)$ denotes the number of documents in Tr in which t_k occurs at least once and

$$tf(t_k, d_j) = \begin{cases} 1 + \log \#(t_k, d_j) & \text{if } \#(t_k, d_j) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

where $\#(t_k, d_j)$ denotes the number of times t_k occurs in d_j . Weights have been further normalized by cosine normalization, i.e.

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{r'} tfidf(t_s, d_j)^2}}$$

where r' is the set of features resulting from feature selection.

The conventions used in the formatting of Table 3 are similar to those discussed for Table 2. In particular, we recall that every entry describes the performance of the Rocchio classifier on a unigram representation (upper row) and on a uni+bigram representation (lower row), where these representations use the same number of features; this ultimately means that the value of bigrams for our TC purposes can be measured by how often the second sub-row reports a better result than the first, and by the magnitude of these improvements.

The results of Table 3 show that an increase in the value of $AvgScore(\varphi_\rho^f)$ as a result of the insertion of high-scoring bigrams in the feature set, does not always contribute to the categorization effectiveness of the Rocchio classifier: 20 out of 48 cases witness an improvement in effectiveness,

while in the other 28 cases we actually have a loss in performance. Moreover, when bigrams bring about a performance improvement, this is seldom significant (the best improvement is 2.8%, obtained for F_1^M with χ_{max}^2 and $\rho = .60$). Conversely, when bigrams cause a deterioration in performance, this is often very significant (the worst deterioration is 35.1%, obtained for F_1^M with OR_{max} and $\rho = .70$). All this is in some sense unexpected, as the results of Figure 1 and Table 2 would seem to indicate that, particularly when penetration levels and increases in $AvgScore(\varphi_\rho^f)$ are high, the overall “quality” of the feature set increases.

Improvements are evenly distributed in the microaveraged and macroaveraged cases. Rather, we may observe that:

1. improvements are achieved more often for low than for high reduction factors. For instance, a reduction factor of .60/.9740 often tends to be associated to performance gains, while a reduction factor of .90/.9870 almost invariably brings about effectiveness losses.
2. the loss in effectiveness introduced by bigrams is higher for those FEFs that have achieved high penetration levels. For instance, the cases in which bigrams improve performance are obtained for IG (7 out of 8 cases), DF (5 out of 8), and χ_{avg}^2 (5 out of 8); these are the FEFs of Group 1, i.e. the ones that had yielded the smallest penetration levels (Figure 1) and the smallest increases in $AvgScore(\varphi_\rho^f)$ (Table 2). Conversely, the FEFs that had produced high penetration levels and increases in $AvgScore(\varphi_\rho^f)$ (i.e. those of Group 2) perform badly, as in the case of χ_{max}^2 (3 out of 8), or even disastrously, as in the case of OR_{avg} and OR_{max} (both achieve 0 out of 8).
3. increases in the values of $AvgDF(\varphi_\rho^f)$ and $AvgCF(\varphi_\rho^f)$ (Table 2) seem to be associated with an increase in performance, although this is not a definitely clear pattern.

These observations (especially 1 and 2) seem to indicate that an excessive use of bigrams at the expense of unigrams may be detrimental to effectiveness, even if the value of $AvgScore(\varphi_\rho^f)$ is increased by letting bigrams in. This may indicate that important unigrams are pushed out of the top set by bigrams that somehow “duplicate” the information carried by existing unigrams. For instance, *inform retriev*, *inform* and *retriev* may all be selected for the top set, with *inform retriev* pushing out a unigram that is quite unrelated to all other remaining features. This is a consequence of the reductionistic nature of the “filtering” approach to feature selection, i.e. the fact that a feature is evaluated independently of all other features. In principle, a better approach would be the “wrapper” approach to feature selection [14], whereby feature subsets are evaluated globally. However, this approach is impractical in TC, since it is computationally infeasible when the set of features to choose from is large.

Besides eliminating potentially informative unigrams, the selection of too many bigrams has the further drawback that it increases the pairwise stochastic dependence between features, a situation which is at odds with the principles underlying most text classifiers currently used (including Rocchio). There are methods designed to handle such situations, e.g. maximum entropy [16]. Maximum entropy combines feature selection with a classifier, somewhat similarly to Bayesian methods. When confronted with words that co-occur frequently (a situation that can be the effect of the above-mentioned “duplication”), maximum entropy avoids the conclusion that this co-occurrence is a significant predictor of class membership. However, [16, 24] have reported mixed performance of maximum entropy when used in practical applications. While in some domains an improvement has been reported with respect to Bayesian classifiers, in some others a deterioration in classification accuracy has been noted [24]. Kantor and Lee [16] report similarly mixed results on an information retrieval task.

7 Related work

Phrase indexing is closely related to the problem of *automatic term recognition* (ATR) in *terminology*, a subfield of computational linguistics that investigates the identification and extraction from

FEF	ρ	Re^μ	Pr^μ	F_1^μ	Re^M	Pr^M	F_1^M
<i>DF</i>	.60	.674	.778	.723	.521	.678	.589
<i>DF</i>	.9740	.683	.788	.732	.530	.688	.599
<i>DF</i>	.70	.674	.778	.723	.522	.679	.590
<i>DF</i>	.9805	.683	.788	.732	.525	.679	.592
<i>DF</i>	.80	.680	.785	.728	.528	.683	.595
<i>DF</i>	.9870	.681	.785	.729	.512	.651	.573
<i>DF</i>	.90	.686	.791	.734	.524	.670	.588
<i>DF</i>	.9935	.669	.772	.717	.493	.616	.548
<i>IG</i>	.60	.674	.777	.722	.520	.679	.589
<i>IG</i>	.9740	.684	.789	.732	.532	.680	.597
<i>IG</i>	.70	.676	.780	.724	.526	.683	.594
<i>IG</i>	.9805	.684	.789	.733	.532	.682	.598
<i>IG</i>	.80	.680	.785	.728	.527	.684	.595
<i>IG</i>	.9870	.685	.790	.733	.536	.685	.601
<i>IG</i>	.90	.688	.793	.737	.531	.680	.597
<i>IG</i>	.9935	.682	.788	.731	.534	.697	.604
χ_{avg}^2	.60	.674	.778	.722	.520	.680	.590
χ_{avg}^2	.9740	.686	.791	.734	.538	.693	.606
χ_{avg}^2	.70	.676	.780	.724	.522	.680	.591
χ_{avg}^2	.9805	.686	.792	.735	.538	.695	.606
χ_{avg}^2	.80	.681	.786	.730	.534	.690	.602
χ_{avg}^2	.9870	.685	.790	.734	.520	.679	.589
χ_{avg}^2	.90	.688	.794	.737	.537	.700	.608
χ_{avg}^2	.9935	.674	.778	.722	.495	.622	.551
χ_{max}^2	.60	.676	.780	.725	.518	.676	.587
χ_{max}^2	.9740	.679	.788	.729	.537	.691	.604
χ_{max}^2	.70	.678	.783	.727	.520	.679	.589
χ_{max}^2	.9805	.658	.768	.708	.528	.688	.598
χ_{max}^2	.80	.683	.788	.732	.525	.686	.595
χ_{max}^2	.9870	.619	.748	.677	.513	.675	.583
χ_{max}^2	.90	.682	.787	.731	.530	.692	.600
χ_{max}^2	.9935	.507	.621	.558	.445	.653	.529
<i>OR_{avg}</i>	.60	.667	.770	.715	.518	.673	.585
<i>OR_{avg}</i>	.9740	.608	.711	.655	.486	.697	.573
<i>OR_{avg}</i>	.70	.652	.753	.699	.512	.675	.582
<i>OR_{avg}</i>	.9805	.583	.693	.633	.449	.661	.535
<i>OR_{avg}</i>	.80	.631	.731	.677	.483	.661	.558
<i>OR_{avg}</i>	.9870	.566	.692	.623	.437	.641	.520
<i>OR_{avg}</i>	.90	.607	.725	.661	.470	.650	.546
<i>OR_{avg}</i>	.9935	.549	.671	.604	.401	.654	.497
<i>OR_{max}</i>	.60	.627	.723	.671	.514	.663	.579
<i>OR_{max}</i>	.9740	.414	.483	.446	.422	.593	.493
<i>OR_{max}</i>	.70	.618	.713	.662	.524	.684	.594
<i>OR_{max}</i>	.9805	.387	.484	.430	.410	.597	.486
<i>OR_{max}</i>	.80	.565	.655	.607	.490	.665	.564
<i>OR_{max}</i>	.9870	.337	.470	.392	.365	.621	.460
<i>OR_{max}</i>	.90	.460	.538	.496	.449	.644	.529
<i>OR_{max}</i>	.9935	.261	.666	.375	.264	.733	.388

Table 3: Comparison between the unigram and the uni+bigram effectiveness of a Rocchio classifier for different FEFs and different reduction factors.

texts of linguistic units which characterise specialised domains. In their review of ATR research, Kageura and Umino [15] draw a distinction between research that emphasizes “unithood” (i.e. the fact that a given linguistic expression qualifies as a “term” from a *syntactic* point of view) and research that instead emphasizes “termhood” (i.e. the fact that a given linguistic expression qualifies as a “term” from a *semantic* point of view). The distinction we have drawn between syntactic and statistical phrases for use in IR and TC is very similar.

7.1 Related work in information retrieval

Work on the use of either syntactic or statistical phrases in IR dates back to the early ’70s (see [7] for a review of this early work). However, it was not until Fagan’s work [7, 8] that thorough experimental comparison between standard indexing, syntactic phrase indexing and statistical phrase indexing was performed. In his experiments Fagan found syntactic phrases to yield very small effectiveness improvements, notwithstanding the fact that a sophisticated linguistic technique had been employed for phrase extraction. More importantly, he also found that statistical phrases obtained by a simple method improved performance a lot more than the syntactic phrases.

Lewis and Croft [20] investigated the idea of extracting syntactic phrases and then clustering them in order to endow the resulting indexing language with better statistical properties, but this did not result in significant effectiveness improvements.

Mitra et al. [21] investigated the impact of both syntactic and statistical phrases in IR. Their research showed that the difference in effectiveness between the two is almost negligible, and that there is a significant overlap between the sets of phrases identified by the two methods (41% of the union of the two sets is in their intersection). They also showed that phrase indexing gives little benefits at low recall levels, but the benefits tend to increase at high recall levels. This is an important observation for TC applications, since in TC the recall level is usually a parameter learnt on a validation set; this means that if phrases are used, in TC the recall level that maximizes overall performance is automatically chosen by the system. The statistical phrases of [21] are exactly equivalent to our bigrams (they do not consider n -grams for $n \geq 3$), with the only difference that an empirical statistical filter is used in place of our FEFs (i.e. only bigrams occurring in more than 25 documents are considered).

The results of [21] concerning statistical phrases have essentially been confirmed by a later study by Turpin and Moffat [30], who have also tried to use non-alphabetically-ordered phrases without obtaining substantially different results.

7.2 Related work in text categorization

While quite a few researchers have investigated the usefulness of phrase indexing for IR purposes, relatively few have done the same in a TC context. A number of researchers, although using syntactic [10, 31] or statistical [1, 25, 26, 27] phrases for TC purposes, do not provide explicit comparisons between performance with and without phrases.

7.2.1 Syntactic phrases

Lewis [18, 19] has been the first to study the effects of syntactic phrase indexing in a TC context. He reported that, in the context of a Naïve Bayes classifier, this yields significantly lower effectiveness than standard “set-of-words” indexing, regardless of whether the syntactic phrases are successively clustered (similarly to [20]) or not. It has to be remarked, though, that Lewis’ phrase indexing language consisted of *phrases only*; this is different from most other works (including the present one), in which the indexing language includes both unigrams and phrases.

Dumais et al. [5] reported no benefit at all from the use of syntactic phrases with a variety of text classifiers in the context of Reuters-21578 experimentation.

Fürnkranz et al. [12] showed that syntactic phrases yield precision improvements at low recall levels, somehow confirming the results obtained by Mitra et al. [21] in an IR context.

7.2.2 Statistical phrases

Mladenić and Grobelnik [23] have extracted n -grams of length up to 5 by means of a fast (although incomplete) algorithm that relies on document frequency as a statistical filter. On a Naïve Bayes classifier applied to a corpus of Web pages they have found that n -grams of length up to 4 give significant benefits with respect to the single words case, while 5-grams do not provide additional benefit.

Fürnkranz [11] uses an algorithm similar to that of [23] to extract n -grams of length up to 5. On Reuters-21578 he has found that RIPPER [3] has a significant improvement in performance when n -grams of length up to 2 are used, but that longer n -grams reduce classification performance; on another dataset of Usenet newsgroup articles he instead found also 3-grams to have some utility, whereas the negative contribution of larger n -grams was confirmed.

8 Conclusion

We have investigated the usefulness of bigrams in text categorization by first performing a learner-independent study and then assessing whether the indications of this study were confirmed by real text categorization experiments. Although our experiments have been restricted to bigrams, the methodology we have employed can be used for general n -grams. The learner-independent study has shown that feature evaluation functions routinely used in text categorization experiments tend to score many bigrams higher than unigrams that they would themselves select in unigram-only feature selection tasks, sometimes giving rise to high bigram “penetration levels”. This would seem to indicate that there is value added in using an indexing language that also contains bigrams.

Our hypothesis that a high bigram penetration level were conducive to improving effectiveness was not confirmed. In particular, our experiments have shown that if the feature evaluation function being used gives rise to a too high bigram penetration level, effectiveness may decrease. It is easy to conjecture that this is due to the elimination of informative unigrams on the part of bigrams that partly duplicate the information carried by existing unigrams. We think that the issue of information duplication as a result of bigram insertion is central to understanding why significant bigram penetration levels do not result in classifier effectiveness improvements. The investigation of ways to avoid this duplication is the main direction along which we plan to continue our work.

We think that this approach sheds some light on the role of bigrams in TC, a role that in previously published experiments had been clouded by learner-dependent issues.

Further, we remark that this study uses a definition of n -grams that is standard in IR contexts and, nevertheless, has never been evaluated in TC experiments. In fact, one difference between the experiments in [11, 23] and our experiment, apart from the obvious issue of learner-independence, is that [11, 23] used no stemming and no alphabetical ordering. This is an important difference since, as discussed in Section 3, stemming and alphabetical ordering allow to factor out a significant number of morphological, syntactic and semantic differences between linguistic expressions. The comparison between their experiments, which have uniformly shown effectiveness improvements, and our own, which have produced more mixed results, might induce one to believe that stemming and alphabetical ordering should be avoided in bigram extraction. However, we remark that another difference between our research and all other works discussed in this section (including the one of [11, 23]) is that, in comparing the effectiveness deriving from standard indexing with that deriving from phrase indexing, we keep the number of features fixed (i.e. bigrams *substitute* some unigrams in the representations) while in all other works this is not the case (i.e. bigrams *are added* to the unigrams in the representation). It might be the case that if we too had worked by addition, rather than by substitution, effectiveness would have uniformly improved. We have chosen to work by substitution because, unlike in IR, in TC the dimensionality of the feature space is an important parameter (see Section 2.1), and because of this any comparison between different representation schemes is significant only if the numbers of features used are the same.

Acknowledgements

We thank Luigi Galavotti for making available his REALCAT text classification software environment [13], which greatly simplified our experimental work, William Cohen for making available his RIPPER system, and Gianni Ginesi and Emanuele Granucci for their collaboration in the implementation of the n -gram extraction software. Finally, thanks to Irene Loiseau for making the three of us meet.

References

- [1] C. Apté, F. J. Damerou, and S. M. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.
- [2] C. Buckley, A. Singhal, and M. Mitra. Using query zoning and correlation within SMART: TREC-5. In E. M. Voorhees and D. K. Harman, editors, *Proceedings of TREC-5, 5th Text Retrieval Conference*, Gaithersburg, US, 1996.
- [3] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999.
- [4] F. J. Damerou. Evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*, 29(4):433–447, 1993.
- [5] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In G. Gardarin, J. C. French, N. Pissinou, K. Makki, and L. Bouganim, editors, *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US, 1998. ACM Press, New York, US.
- [6] C. Enguehard and L. Pantera. Automatic natural acquisition of a terminology. *Journal of Quantitative Linguistics*, 2(1):27–32, 1995.
- [7] J. L. Fagan. *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods*. PhD thesis, Department of Computer Science, Cornell University, Ithaca, US, 1987.
- [8] J. L. Fagan. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, 40(2):115–132, 1989.
- [9] W. B. Frakes. Stemming algorithms. In W. B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms*, pages 131–160. Prentice Hall, Englewood Cliffs, US, 1992.
- [10] N. Fuhr, S. Hartmann, G. Knorz, G. Lustig, M. Schwantner, and K. Tzeras. AIR/X – a rule-based multistage indexing system for large subject fields. In A. Lichnerowicz, editor, *Proceedings of RIAO-91, 3rd International Conference “Recherche d’Information Assistée par Ordinateur”*, pages 606–623, Barcelona, ES, 1991. Elsevier Science Publishers, Amsterdam, NL.
- [11] J. Fürnkranz. A study using n -gram features for text categorization. Technical Report TR-98-30, Oesterreichisches Forschungsinstitut für Artificial Intelligence, Wien, AT, 1998. <http://www.ai.univie.ac.at/cgi-bin/tr-online?number+98-30>.
- [12] J. Fürnkranz, T. M. Mitchell, and E. Riloff. A case study in using linguistic phrases for text categorization on the WWW. In *Proceedings of the 1st AAAI Workshop on Learning for Text Categorization*, pages 5–12, Madison, US, 1998.

- [13] L. Galavotti, F. Sebastiani, and M. Simi. Experiments on the use of feature selection and negative evidence in automated text categorization. Technical Report IEI-B4-xx-2000, Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 2000. Submitted for publication to *ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries*, Lisbon, PT.
- [14] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of ICML-94, 11th International Conference on Machine Learning*, pages 121–129, New Brunswick, US, 1994.
- [15] K. Kageura and B. Umino. Methods of automatic term recognition: a review. *Terminology*, 3(2):259–289, 1996.
- [16] P. B. Kantor and J. J. Lee. Testing the maximum entropy principle for information retrieval. *Journal of the American Society for Information Science*, 49(6):557–566, 1998.
- [17] S. L. Lam and D. L. Lee. Feature reduction for neural network based text categorization. In A. L. Chen and F. H. Lochovsky, editors, *Proceedings of DASFAA-99, 6th IEEE International Conference on Database Advanced Systems for Advanced Application*, pages 195–202, Hsinchu, TW, 1999. IEEE Computer Society Press, Los Alamitos, US.
- [18] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In N. J. Belkin, P. Ingwersen, and A. M. Pejtersen, editors, *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK, 1992. ACM Press, New York, US.
- [19] D. D. Lewis. *Representation and learning in information retrieval*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US, 1992.
- [20] D. D. Lewis and W. B. Croft. Term clustering of syntactic phrases. In *Proceedings of SIGIR-90, 13th ACM International Conference on Research and Development in Information Retrieval*, pages 385–404, Bruxelles, BE, 1990.
- [21] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO-97, 5th International Conference "Recherche d'Information Assistee par Ordinateur"*, pages 200–214, Montreal, CA, 1997. An extended version is forthcoming on *Information Processing and Management*.
- [22] D. Mladenić. Feature subset selection in text learning. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398 in Lecture Notes in Computer Science, pages 95–100, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.
- [23] D. Mladenić and M. Grobelnik. Word sequences as features in text-learning. In *Proceedings of ERK-98, the Seventh Electrotechnical and Computer Science Conference*, pages 145–148, Ljubljana, SL, 1998.
- [24] K. Nigam, J. Lafferty, and A. K. McCallum. Using maximum entropy for text classification. In *Proceedings of the IJCAI-99 Workshop on Information Filtering*, Stockholm, SE, 1999.
- [25] R. E. Schapire and Y. Singer. BOOSTEXTER: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.
- [26] R. E. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio applied to text filtering. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 215–223, Melbourne, AU, 1998. ACM Press, New York, US.

- [27] H. Schütze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 229–237, Seattle, US, 1995. ACM Press, New York, US.
- [28] F. Sebastiani. Machine learning in automated text categorisation. Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell’Informazione, Consiglio Nazionale delle Ricerche, Pisa, IT, 1999. Submitted for publication to *ACM Computing Surveys*.
- [29] F. Smadja. Retrieving collocations from text: XTRACT. *Computational Linguistics*, 19(1):143–178, 1993.
- [30] A. Turpin and A. Moffat. Statistical phrases for vector-space information retrieval. In *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 309–310, Berkeley, US, 1999.
- [31] K. Tzeras and S. Hartmann. Automatic indexing based on Bayesian inference networks. In R. Korfhage, E. Rasmussen, and P. Willett, editors, *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 22–34, Pittsburgh, US, 1993. ACM Press, New York, US.
- [32] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.
- [33] J. Zhou. Phrasal terms in real-world IR applications. In T. Strzalkowski, editor, *Natural language information retrieval*, pages 215–259. Kluwer Academic Publishers, Dordrecht, NL, 1999.