An Analysis of the Relative Hardness of Reuters-21578 Subsets

Franca Debole and Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via Giuseppe Moruzzi, 1, 56124 Pisa, Italy. E-mail: {franca.debole, fabrizio.sebastiani}@isti.cnr.it

The existence, public availability, and widespread acceptance of a standard benchmark for a given information retrieval (IR) task are beneficial to research on this task, because they allow different researchers to experimentally compare their own systems by comparing the results they have obtained on this benchmark. The Reuters-21578 test collection, together with its earlier variants, has been such a standard benchmark for the text categorization (TC) task throughout the last 10 years. However, the benefits that this has brought about have somehow been limited by the fact that different researchers have "carved" different subsets out of this collection and tested their systems on one of these subsets only; systems that have been tested on different Reuters-21578 subsets are thus not readily comparable. In this article, we present a systematic, comparative experimental study of the three subsets of Reuters-21578 that have been most popular among TC researchers. The results we obtain allow us to determine the relative hardness of these subsets, thus establishing an indirect means for comparing TC systems that have, or will be, tested on these different subsets.

Introduction

The existence, public availability, and widespread acceptance of a standard benchmark for a given information retrieval (IR) task are beneficial to research on this task, because they allow different researchers to experimentally compare their own systems by comparing the results they have obtained on this benchmark.

The Reuters-21578 test collection, together with its earlier variants, has been such a standard benchmark for the text categorization (TC) task throughout the last 10 years.[1] Reuters-21578 is a set of 21,578 news stories that appeared in the Reuters newswire in 1987, which are classified according to 135 thematic categories mostly concerning business and economy. This collection has several characteristics that make it interesting for TC experimentation:

- Similar to many other applicative contexts, it is multilabel, i.e., each document d_i may belong to more than one category.
- The set of categories is not exhaustive, i.e., some documents belong to no category at all.
- The distribution of the documents across the categories is highly skewed, in the sense that some categories have very few documents classified under them ("positive examples") while others have thousands.
- There are several semantic relations among the categories (e.g., there is a category WHEAT and a category GRAIN, which are obviously related), but these relations are "hidden" (i.e., there is no explicit hierarchy defined on the categories).

This collection is also fairly challenging for TC systems based on machine learning (ML) techniques, because several categories have (under any possible split between training and test documents) very few positive training examples, making the inductive construction of a classifier a hard task. All of these properties have made Reuters-21578 the benchmark of choice for TC research in the past years.

Unfortunately, the benefits to TC research that Reuters-21578 has brought about have been somehow limited by the fact that different researchers have "carved" different subcollections out of it, and tested their systems on one of these subcollections only. The most frequent direction for extracting a

Received October 31, 2003; revised January 20, 2004; accepted March 11, 2004

© 2005 Wiley Periodicals, Inc. • Published online 4 February 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20147

[1] While a new Reuters corpus has recently been made available for TC research (Lewis, Li, Rose, & Yang, 2004; Rose, Stevenson, & Whitehead, 2002), its take-up has been somehow slow, and also hindered by terms of use that are not universally acceptable by interested parties. For example, it has been reported that some universities in the United States are not willing to sign the license of use agreement with Reuters on the ground that the agreement requires that all legal disputes be settled in England. This *de facto* prevents researchers from these universities to experiment on this corpus.

subcollection out of Reuters-21578 has been that of restricting the attention to a subset of categories only. The subsets that have been most frequently used in TC experimentation are:[2]

- The set of the 10 categories with the highest number of positive training examples [hereafter, R(10)]
- The set of the 90 categories with at least one positive training example and one positive test example [hereafter, R(90)]
- The set of the 115 categories with at least one training example [hereafter, R(115)]

Systems that have been tested on these different Reuters-21578 subsets are thus not readily comparable. In this article, we present a systematic, comparative experimental study of the three subsets of Reuters-21578 just listed. We test the relative hardness of these subsets in a variety of experimental TC contexts, generated by two different term weighting policies, three different feature selection functions, three different "reduction factors" for feature selection, three different learning methods, and two different experimental measures, in all possible combinations. Our results allow us to obtain a reliable estimation of the relative difficulty of these subsets, thus establishing an indirect means for comparing TC systems that have, or will be, tested on these different subsets.

This article is structured as follows. The next section briefly introduces the TC task and the related terminology, thus setting the stage for the description of our experimental work. In the section that follows, we describe in some detail the Reuters-21578 test collection and the subsets of it that have been used most often in TC research. The section after that presents a systematic experimental study in which we test the relative hardness of these subsets and give theoretical justifications for these results. The conclusion is in the final section.

Preliminaries: An Introduction to Text Categorization

Text categorization (TC, also known as *text classification*) is the task of approximating the unknown *target function* $\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$ (that describes how documents ought to be classified) by means of a function $\hat{\Phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$ called the *classifier*, where $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ is a predefined set of categories and $\mathcal{D}$ is a domain of documents. If $\Phi(d_j, c_i) = T$, then d_j is called a *positive example* (or a *member*) of c_i, while if $\Phi(d_j, c_i) = F$ it is called a *negative example* of c_i.

Depending on the application, TC may be either *single-label* (i.e., exactly one $c_i \in \mathcal{C}$ must be assigned to each $d_j \in \mathcal{D}$), or *multilabel* (i.e., any number $0 \leq n_j \leq |\mathcal{C}|$ of categories may be assigned to each $d_j \in \mathcal{D}$). A special case of single-label TC is *binary* TC, in which, given a category c_i, each $d_j \in \mathcal{D}$ must be assigned either to c_i or to its complement $\bar{c}_i$; a *classifier for* c_i is then a function $\hat{\Phi}_i : \mathcal{D} \rightarrow \{T, F\}$ that approximates the unknown target function $\Phi_i : \mathcal{D} \rightarrow \{T, F\}$. Multilabel TC under $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ is usually tackled as $|\mathcal{C}|$ independent binary classification problems under $\{c_i, \bar{c}_i\}$, for $i = 1, \ldots, |\mathcal{C}|$. Multilabel (and, as a consequence, binary) TC, rather than single-label TC, will be the focus of this article.

We can roughly distinguish three different phases in the life cycle of a TC system: document indexing, classifier learning, and classifier evaluation. The three following paragraphs are devoted to these three phases, respectively; for a more detailed treatment see Sebastiani (2002, Sections 5, 6, and 7).

Document Indexing

Document indexing denotes the mapping of a document d_j into a compact representation of its content that can be directly interpreted (1) by a classifier-building algorithm and (2) by a classifier, once it has been built. The indexing methods usually employed in TC are borrowed from IR, where a text d_j is usually represented as a vector $\vec{d_j} = \langle w_{1j}, \ldots, w_{|\mathcal{T}|j} \rangle$ of term *weights*. Here, $\mathcal{T}$ is the *dictionary*, i.e., the set of *terms* (also know as *features*) that occur at least once in at least one document, and $0 \leq w_{kj} \leq 1$ quantifies the importance of t_k in characterizing the semantics of d_j.

An indexing method is characterized by (1) a definition of what a term is, and (2) a method to compute term weights. Concerning (1), the most frequent choice is to identify terms either with the *words* occurring in the document (with the exception of *stop words*, which are eliminated in a preprocessing phase), or with their *stems* (i.e., their morphological roots, obtained by applying a stemming algorithm). Concerning (2), either statistical or probabilistic techniques are used to compute terms weights, the former being the most common option. One popular class of statistical term weighting functions is $tf * idf$, where two intuitions are at play: (a) the more frequently t_k occurs in d_j, the more important for d_j it is; (b) the more documents t_k occurs in, the less discriminating it is (i.e., the smaller its contribution is in characterizing the semantics of a document in which it occurs). Weights computed by $tf * idf$ techniques are often normalized so as to contrast the tendency of $tf * idf$ to emphasize long documents.

In TC, unlike in IR, a *dimensionality reduction* phase is often applied so as to reduce the size of the document representations from $|\mathcal{T}|$ to a much smaller, predefined number $|\mathcal{T}'| \ll |\mathcal{T}|$; the value $\xi = \frac{|\mathcal{T}| - |\mathcal{T}'|}{|\mathcal{T}|}$ is called the *reduction factor*. Dimensionality reduction reduces *overfitting* (i.e., the tendency of the classifier to better classify the data it has been trained on than new unseen data), and makes the problem more manageable for the learning method, because many such methods are known not to scale well to high problem sizes. Dimensionality reduction often takes the form of *term selection*: each term t_k is scored by means of a

[2] As for which Reuters-21578 documents are used as training examples, we here refer to the ModApté split, a partition of the collection into a training set and a test set that has almost universally been adopted by TC experimenters. See the section on the Reuters-21578 collection and its subsets for more details.

scoring function $f(t_k, c_i)$ that captures its degree of (positive or negative) correlation with c_i, and only the highest scoring terms (i.e., the most highly correlated with c_i) are used for document representation. The TC literature discusses two main policies to perform term selection: (a) a *local* policy, according to which different sets of terms $\mathcal{T}_i' \subset \mathcal{T}$ are selected for different categories c_i, and (b) a *global* policy, according to which a single set of terms $\mathcal{T}' \subset \mathcal{T}$, to be used for all categories, is selected by extracting a single score $f_{glob}(t_k)$ from the individual scores $f(t_k, c_i)$ by means of some *globalization* policy.

Classifier Learning

A text classifier for c_i is automatically generated by a general inductive process (the *learner*) which, by observing the characteristics of a set of documents preclassified under c_i or $\bar{c}_i$, gleans the characteristics that a new unseen document should have to belong to c_i. To build classifiers for $\mathcal{C}$ one thus needs a corpus Ω of documents such that the value of $\Phi(d_j, c_i)$ is known for every $\langle d_j, c_i\rangle \in \Omega \times \mathcal{C}$. In experimental TC it is customary to partition Ω into two disjoint sets *Tr* (the *training set*) and *Te* (the *test set*). The training set is the set of documents that the learner observes in order to build the classifier, whereas the test set is the set on which the effectiveness of the classifier is finally evaluated. Sometimes the engineer extracts a *validation set Va* from *Tr* before training, for fine-tuning purposes. The learner builds the classifier by observing only the documents in $Tr - Va$. Subsequently, the engineer may fine-tune the classifier by choosing for a parameter p on which the classifier depends (e.g., a threshold), the value that has yielded the best effectiveness when evaluated on *Va*. In both the validation and test phase, evaluating the effectiveness means running the classifier on a set of preclassified documents (*Va* or *Te*) and checking the degree of correspondence between the output of the classifier and the preassigned labels.

Different learners have been applied in the TC literature, including probabilistic methods, regression methods, decision tree and decision rule learners, neural networks, batch and incremental learners of linear classifiers, example-based methods, support vector machines, genetic algorithms, hidden Markov models, and classifier committees. Some of these methods generate binary-valued classifiers of the required form $\hat{\Phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$, but some others generate real-valued functions of the form $CSV : \mathcal{D} \times \mathcal{C} \rightarrow [0, 1]$ (CSV standing for *categorization status value*). For these latter, a set of thresholds τ_i needs to be determined (typically, by experimentation on a validation set) allowing to turn real-valued CSVs into the final binary decisions.

Classifier Evaluation

Both *training efficiency* (i.e., average time required to build a classifier $\hat{\Phi}_i$ from a corpus Ω), *classification efficiency* (i.e., average time required to classify a document by means of $\hat{\Phi}_i$), and *effectiveness* (i.e., average correctness of $\hat{\Phi}_i$'s classification behavior) are measures of success for a TC system. However, effectiveness is considered the most important criterion, because in most applications one is willing to trade training time and classification time for correct decisions. Also, it is the most reliable one when it comes to comparing different learners, because efficiency depends on too volatile parameters.

In binary TC, effectiveness is always measured by a combination of *precision* (π_i), the percentage of documents classified into c_i that indeed belong to c_i, and *recall* (ρ_i), the percentage of documents belonging to c_i that are indeed classified into c_i. Because a classifier can be tuned to emphasize precision at the expense of recall, or vice versa, only combinations of the two are significant, the most popular combination nowadays being (Lewis, 1995)

$$F_{1_i} = \frac{2\pi_i \rho_i}{\pi_i + \rho_i} = \frac{2 \cdot TP_i}{2 \cdot TP_i + FP_i + FN_i} \qquad (1)$$

where TP_i, FP_i and FN_i refer to the sets of *true positives wrt* c_i (documents correctly deemed to belong to c_i), *false positives wrt* c_i (documents incorrectly deemed to belong to c_i), and *false negatives wrt* c_i (documents incorrectly deemed not to belong to c_i), respectively.

When effectiveness is computed for several categories, the results for individual categories must be averaged in some way; here, one may opt for *microaveraging* (categories count proportionally to the number of their positive test examples—indicated by the μ superscript) or for *macroaveraging* (all categories count the same—indicated by the M superscript), depending on the application. The former rewards classifiers that behave well on *frequent categories* (i.e., categories with many positive examples), while classifiers that perform well also on infrequent categories are emphasized by the latter. Table 1 displays the mathematical definitions of precision, recall, and F_1, in both their microaveraged and macroaveraged variants.

Measuring effectiveness requires a *test collection*; in multilabel TC, this consists of a set of documents each of which is labeled with zero, one, or several categories from a prespecified set. The following section will discuss in detail the test collection that is the object of study of this article.

The Reuters-21578 Collection and Its Subsets

The data contained in the "Reuters-21578, Distribution 1.0" corpus consist of news stories that appeared on the Reuters newswire in 1987.[3] The data was originally labeled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system (Hayes & Weinstein, 1990), and was subsequently collected and formatted by David Lewis with the help of several other people. A previous version of the collection, known as Reuters-22173, was used in a number of published studies

[3] The Reuters-21578 corpus is freely available for experimentation purposes from http://www.daviddlewis.com/resources/testcollections/~reuters21578/.

TABLE 1. Averaging precision, recall, and F_1, across different categories.

	Microaveraging (μ)	Macroaveraging (M)
Precision (π)	$\pi^{\mu} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert} (TP_i + FP_i)}$	$\pi^{M} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \pi_i}{\lvert\mathcal{C}\rvert} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \dfrac{TP_i}{TP_i + FP_i}}{\lvert\mathcal{C}\rvert}$
Recall (ρ)	$\rho^{\mu} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert} (TP_i + FN_i)}$	$\rho^{M} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \rho_i}{\lvert\mathcal{C}\rvert} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \dfrac{TP_i}{TP_i + FN_i}}{\lvert\mathcal{C}\rvert}$
F_1	$F_1^{\mu} = \dfrac{2 \cdot \sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{2 \cdot \sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i + \sum_{i=1}^{\lvert\mathcal{C}\rvert} FP_i + \sum_{i=1}^{\lvert\mathcal{C}\rvert} FN_i}$	$F_1^{M} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} F_{1_i}}{\lvert\mathcal{C}\rvert} = \dfrac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \dfrac{2 \cdot TP_i}{2 \cdot TP_i + FP_i + FN_i}}{\lvert\mathcal{C}\rvert}$

up until 1996, when a revision of the collection resulted in the removal of 595 duplicates from the original set of 22,173 documents, thus leaving the 21,578 documents that now make Reuters-21578, and in the correction of several other errors.

The Reuters-21578 documents actually used in TC experiments are only 12,902, because the creators of the collection found ample evidence that the other 8,676 documents had not been considered for labeling by the people who manually assigned categories to documents (indexers). To make different experimental results comparable, standard *splits* (i.e., partitions into a training and a test set) have been defined by the creators of the collection on the 12,902 documents. Apart from very few exceptions, TC researchers have used the ModApté split, in which 9,603 documents are selected for training and the other 3,299 form the test set. In this article we will always refer to the ModApté split.

There are five groups of categories that label Reuters-21578 documents: EXCHANGES, ORGS, PEOPLE, PLACES, and TOPICS. Only the TOPICS group has actually been used in TC experimental research, because the other four groups do not constitute a very challenging benchmark for TC.

The TOPICS group contains 135 categories. Some of the 12,902 legitimate documents have no categories attached to them, but unlike the 8,676 documents removed from consideration they are unlabeled because the indexers deemed that none of the TOPICS categories applied to them. Among the 135 categories, 20 have (in the ModApté split) no positive training documents; as a consequence, these categories have never been considered in any TC experiment, because the TC methodology requires deriving a classifier either by automatically training an inductive method on the training set only, and/or by human knowledge engineering based on the analysis of the training set only.

Because the 115 remaining categories have at least one positive training example each, in principle they can all be used in experiments. However, several researchers have preferred to carry out their experiments on different subsets of categories. Globally, the three subsets that have been most popular are[4]

- The set of the 10 categories with the highest number of positive training examples, hereafter, R(10). Among others, this has been used in Bennett, 2003; Bennett, Dumais, and Horvitz, 2002; Dumais, Platt, Heckerman, and Sahami, 1998; McCallum and Nigam, 1998; Nigam, McCallum, Thrun, and Mitchell, 2000; Tong and Koller, 2001.
- The set of 90 categories with at least one positive training example and one test example, hereafter, R(90). This appears to be the most frequently chosen subset; among others, it has been used in Baker and McCallum, 1998; Chai, Ng, and Chieu, 2002; Crammer and Singer, 2002; Gao, Wu, Lee, and Chua, 2003; Joachims, 1998; Lam and Lai, 2001; Li and Yamanishi, 1999; Nigam, McCallum, Thrun, and Mitchell, 2000; Sebastiani, Sperduti, and Valdambrini, 2000; Toutanova, Chen, Popat, and Hofmann, 2001; Yang and Liu, 1999.
- The set of 115 categories with at least one positive training example, hereafter, R(115). Among others, this has been used in Benkhalifa, Mouradi, and Bouyakhf, 2001; Caropreso, Matwin, and Sebastiani, 2001; Dumais et al., 1998; Galavotti, Sebastiani, and Simi, 2000; Nardiello, Sebastiani, and Sperduti, 2003.

It follows from this discussion that $R(10) \subset R(90) \subset R(115)$. Reasons for using one or the other subset have been different. Several researchers claim that R(10) is more realistic because machine learning techniques cannot perform adequately when positive training examples are scarce, and/or because small numbers of positive test examples make the interpretation of effectiveness results problematic because of high variance. Other researchers claim instead that only by striving to work on infrequent categories too can we hope to push the limits of TC technology, and this consideration leads them to use R(90) or R(115). The only clear fact is that the 10 most frequent categories provide an easier testbed than the other two sets, although it is not clear ex-

[4] Note that the three subsets, although differing in the number of categories considered, contain the same 12,902 documents.

actly *how easier*. Furthermore, it is not clear at all whether R(90) is any easier than R(115). The experiments that we describe in this section are exactly aimed at answering these two questions, and in general at establishing the relative difficulty of the three relevant Reuters-21578 subsets.

Experiments

The experiments we have conducted test the relative hardness of the three *above-mentioned* Reuters-21578 subsets in *all* experimental TC contexts corresponding to any combination of a learning method, a term selection function, a reduction factor, a term weighting policy, and an effectiveness function, chosen from the following.

- As for the learning methods, we have used three different methods that allow weighted (nonbinary) input. The first is a standard Rocchio method (Hull, 1994) for learning linear classifiers. A classifier for category c_i consists of a vector of weights

  $$w_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{w_{kj}}{|NEG_i|} \tag{2}$$

  where w_{kj} is the weight of term t_k in document d_j, $POS_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = T\}$ and $NEG_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = F\}$. Conforming to common practice we have set the β and γ control parameters to 16 and 4, respectively. Classification is achieved by performing a dot product between the document vector and the classifier and then thresholding on the result. We have individually optimized each threshold τ_i on a validation set by the *proportional thresholding* method (Lewis, 1992a; Yang, 2001), according to which the threshold τ_i is set to the value such that the proportion of *validation* examples that are classified into c_i is as close as possible to the proportion of *training* examples that are classified into c_i.

  The second learning method is a standard k-NN algorithm, computing the formula

  $$score(d_j, c_i) = \sum_{d_z \in Tr_k(d_j)} (\vec{d_j} \cdot \vec{d_z}) \Phi(d_z, c_i) \tag{3}$$

  where $Tr_k(d_j)$ is the set of the k documents d_z that maximize the dot product $\vec{d_j} \cdot \vec{d_z}$. Classification is performed by thresholding on the scores resulting from Equation 3; here too we have individually optimized each threshold τ_i on a validation set by proportional thresholding. The k parameter has been set to 30, following the results in Galavotti, Sebastiani, and Simi, 2000.

  The third learning method is a support vector machine (SVM) learner as implemented in the SVMlight package (version 3.5; Joachims, 1999). SVMs attempt to learn a surface in $|\mathcal{T}|$-dimensional space that separates the positive training examples from the negative ones with the maximum possible margin, such that the minimal distance between the surface and a training example is maximum. Results in computational learning theory indicate that this tends to minimize the generalization error, i.e., the error of the resulting classifier on yet unseen examples. We have simply opted for the default parameter setting of SVMlight; in particular, this means that a linear kernel has been used.
- As for the term selection functions, we have used a choice among the three functions $\{\chi^2, IG, GR\}$, whose mathematical forms are detailed in Table 2. The first two (chi-square and information gain) are standard tools of the trade in the term selection literature, while the third is an entropy-normalized version of information gain whose use as a term selection function was first proposed in Debole and Sebastiani, 2003. Each of the three functions has been used according to the global policy described in the section on document indexing, essentially for efficiency reasons.[5] Globalization has been achieved by means of the f_{max} function, the globalization function of choice in the TC literature, defined as $f_{max}(t_k) = \max_{i=1}^{|\mathcal{C}|} f(t_k, c_i)$.
- As for the reduction factors for feature selection, we have used a choice among the three values $\xi \in \{0.90, 0.50, 0.0\}$, where a 0.0 reduction factor means no reduction at all.
- As for the term weighting policies, we have used a choice between a standard, cosine-normalized form of $tf * idf$, or a supervised term weighting policy (Debole & Sebastiani, 2003), consisting in replacing the idf component of $tf * idf$ with the function that, in the same experiment, has been

[5] For example, recall that the k-NN learner computes, for each test document d_j, its dot product with each training document, and then ranks these training documents in terms of the computed dot product score. This process is extremely costly from a computational point of view. While this process needs to be performed only once if the global policy is used, it needs to be performed $|\mathcal{C}|$ times if the local policy is used, because in this case the same document has $|\mathcal{C}|$ different representations, and similarity scores (and rankings) thus vary across categories.

TABLE 2. Term selection functions used in this work.

Function	Denoted by	Mathematical form
Chi-square	$\chi^2(t_k, c_i)$	$\frac{[P(t_k, c_i)P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i)P(\bar{t}_k, c_i)]^2}{P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}$
Information gain	$IG(t_k, c_i)$	$\sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \log_2 \frac{P(t, c)}{P(t)P(c)}$
Gain ratio	$GR(t_k, c_i)$	$\frac{\sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \log_2 \frac{P(t, c)}{P(t)P(c)}}{- \sum_{c \in \{c_i, \bar{c}_i\}} P(c) \log_2 P(c)}$

previously used for term selection (this yields, e.g., cosine-normalized *tf* $*$ *GR* if *GR* has been previously used for feature selection). The version of *tf* $*$ *idf* we have used is

$$tfidf(t_k, d_j) = tf(t_k, d_j) \cdot \log\frac{|Tr|}{\#_{Tr}(t_k)} \qquad (4)$$

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 obtained by Equation 4 are then normalized by cosine normalization, finally yielding

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|\mathcal{T}|} tfidf(t_s, d_j)^2}} \qquad (5)$$

- As for the effectiveness functions, we have considered both the microaveraged and macroaveraged version of the F_1 function. Note that when all documents are "true negatives" of the category c_i (i.e., when, for each document d_j, it is the case that $\Phi(d_j, c_i) = \hat{\Phi}(d_j, c_i) = F$, in which case F_1 is technically undefined), we have opted for a value of $F_1 = 1$, because in this case the classifier always returns the correct decision (Lewis, Schapire, Callan, & Papka, 1996).

In all the experiments discussed in this article, stop words have been removed using the stop list provided in Lewis (1992b, pp. 117–118), punctuation has been removed, all letters have been converted to lowercase, numbers have been removed, and stemming has been performed by means of Porter's stemmer.

Experimental Results

The results of our experiments are reported in Figures 1 through 6; the six figures report results for each combination of a term weighting policy (chosen among *tf* $*$ *idf* and

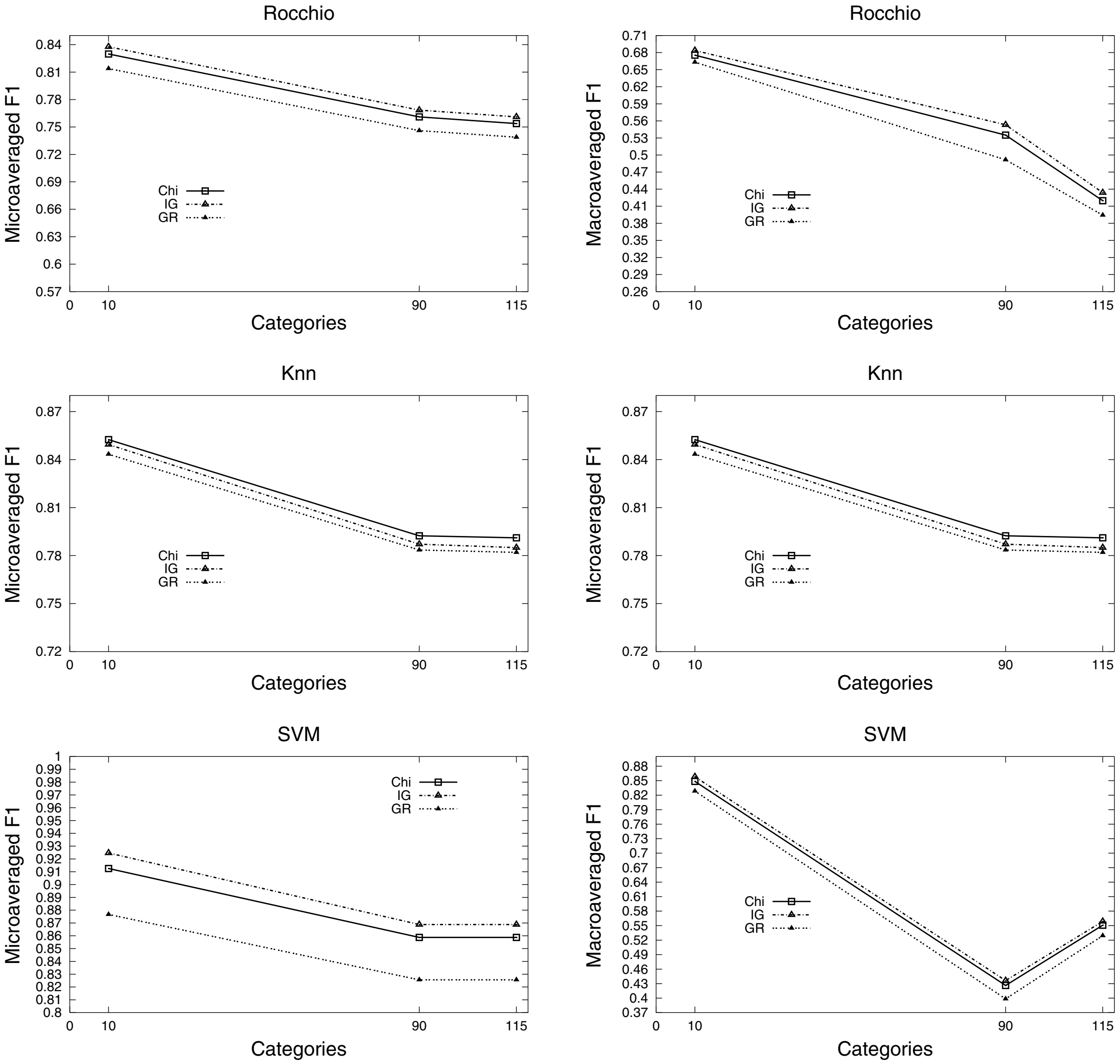

FIG. 1. Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with *tf* $*$ *idf* weighting and a $\xi = 0.90$ reduction factor**. Plots indicate results obtained with Rocchio (top), *k*-NN (middle), and SVMs (bottom). The *X* axis indicates the three subsets of Reuters-21578.

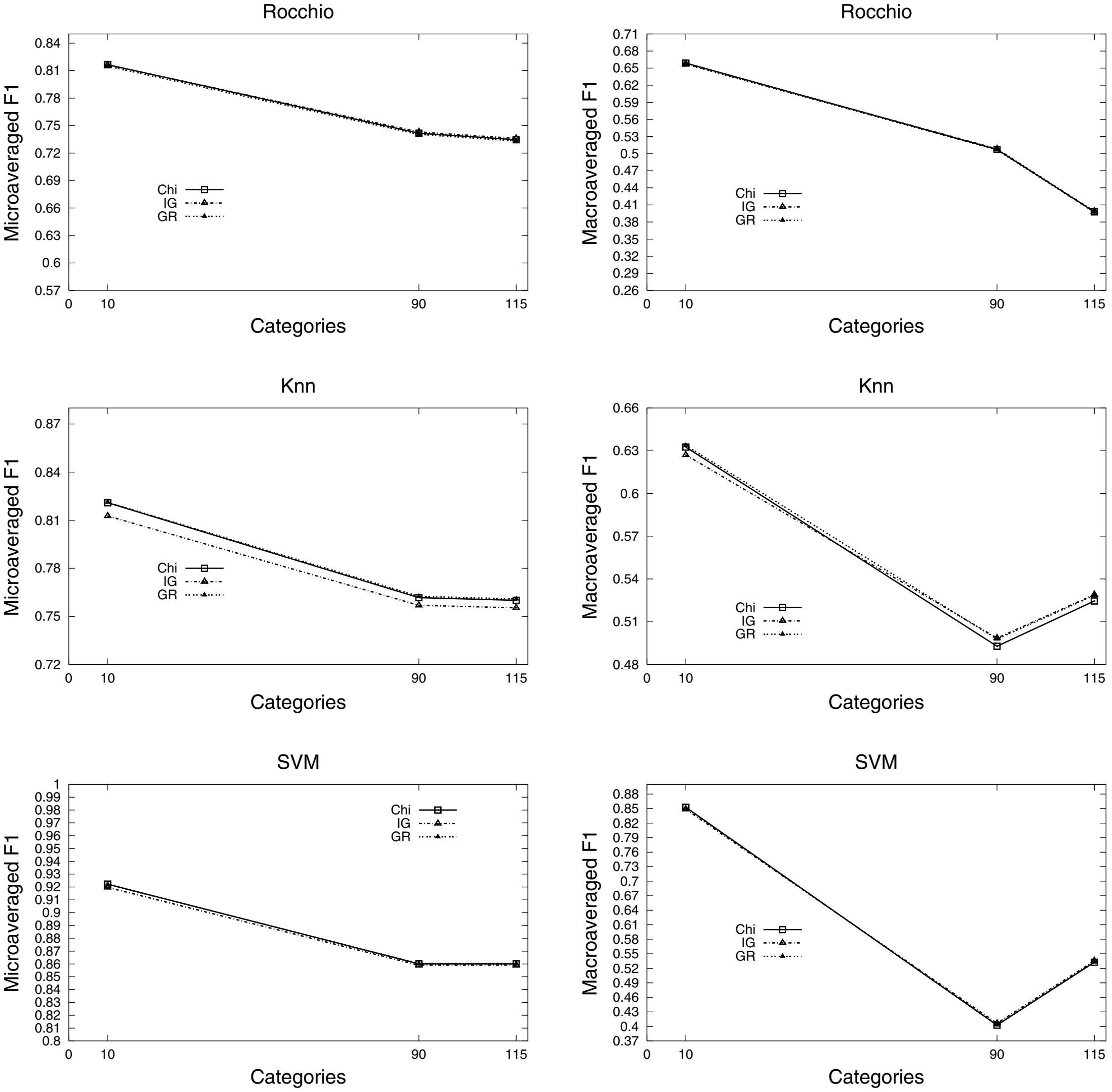

FIG. 2. Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with** ***tf*** **$*$** ***idf*** **weighting and a $\xi = 0.50$ reduction factor**. Plots indicate results obtained with Rocchio (top), *k*-NN (middle), and SVMs (bottom). The *X* axis indicates the three subsets of Reuters-21578.

supervised term weighting) and a feature reduction factor (chosen among $\xi \in \{0.90, 0.50, 0.0\}$). Each figure in turn includes six plots: the leftmost plots report microaveraged F_1 scores while the rightmost report macroaveraged F_1 scores; results obtained with the Rocchio, *k*-NN, and SVM learners are displayed in the top, mid, and bottom row, respectively. Each individual plot (with the obvious exception of Figure 3, which corresponds to *tf* $*$ *idf* weighting and no feature selection) includes three curves, each corresponding to a feature selection function (chosen among *IG*, *GR*, and χ^2).[6]

Figure 7 summarizes these results by averaging them for each studied technique. For example, the curve marked "SVM" reports the average results of all the experiments run with the SVM learner. This means that the average is computed across all possible combinations of term weighting policies, feature selection policies, feature selection functions, and reduction factors for feature selection; separate plots for microaveraged F_1 and macroaveraged F_1 are given. Table 3 reports mean and standard deviation scores obtained across all 48 different experiments, and can thus be considered fairly representative. Finally, Table 4 reinterprets the results of Table 3 in terms of relative hardness of the three Reuters-21578 subsets studied; the values contained in the table can be used for computing the likely performance that a given method tested on Reuters-21578 subset *x* could approximately have obtained if tested on subset *y*.

[6] Note that representing these results as curves is not meant to suggest that the number of categories is a meaningful ordered variable. Rather, the three different points of the *X* axis at which performance values are computed are best viewed as three isolated cases. The curve representation was only chosen for convenience; a histogram representation would have been equally suitable.

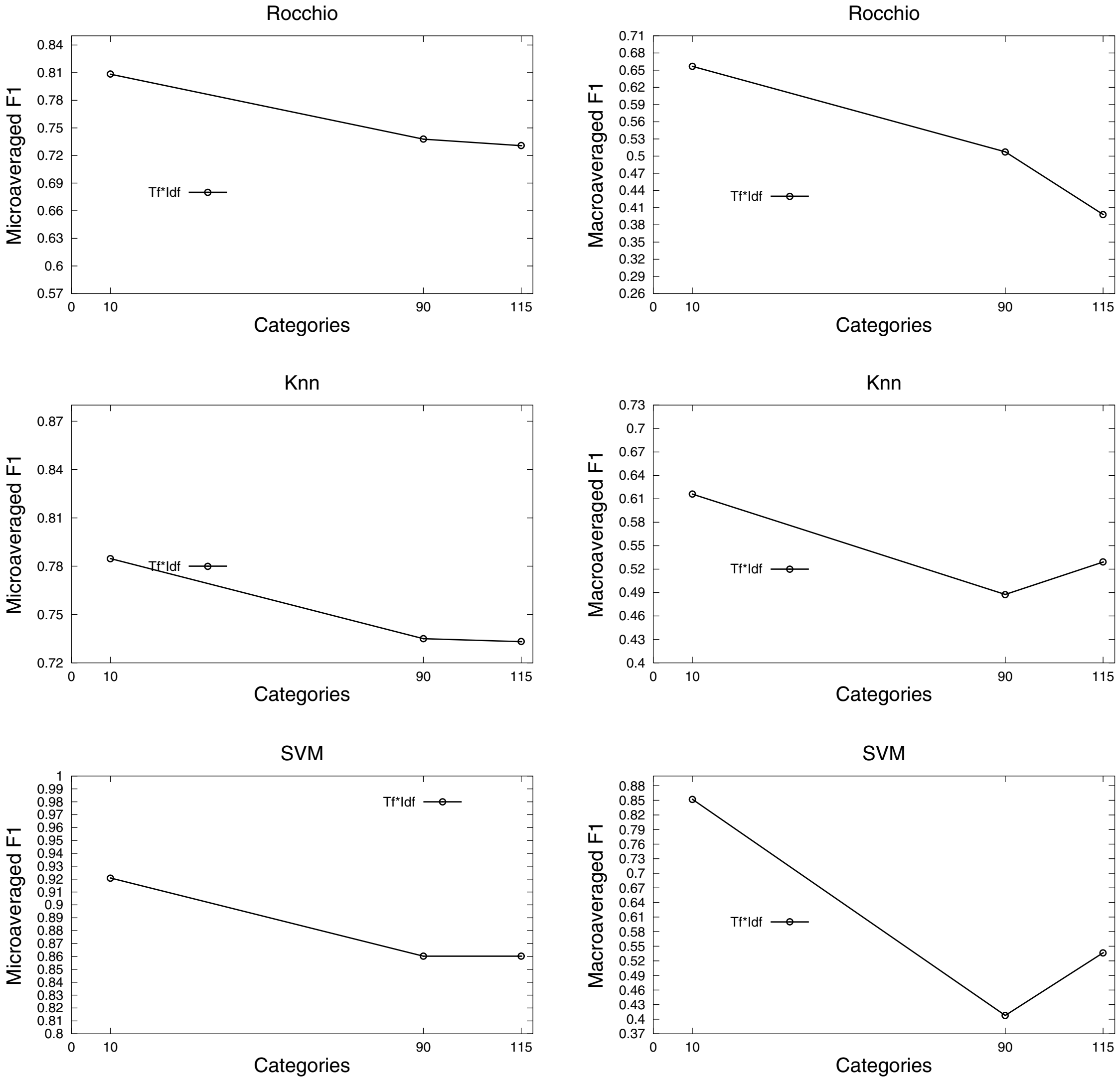

FIG. 3. Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with** ***tf* $*$ *idf* weighting and a $\xi = 0.0$ reduction factor**. Plots indicate results obtained with Rocchio (top), *k*-NN (middle), and SVMs (bottom). The *X* axis indicates the three subsets of Reuters-21578.

The fact that emerges most clearly from these experiments is that R(10) is the easiest subset, regardless of the choice of learning method, feature selection function, effectiveness function, and so on. This was largely to be expected, given that its categories are the ones with the highest number of positive examples, and as such allow taming the "curse of dimensionality" more effectively.

TABLE 3. Average effectiveness and standard deviation scores averaged across all the text classifiers tested in our experiments on the three Reuters-21578 subsets.

	Microaveraged F_1		Macroaveraged F_1	
	Avg	StDev	Avg	StDev
R(10)	0.852	0.048	0.715	0.097
R(90)	0.787	0.059	0.468	0.068
R(115)	0.784	0.062	0.494	0.118

On average, the decrease in performance in going from R(10) to R(90) is much sharper for macroaveraging (−53.1%) than for microaveraging (−7.6%). This can be explained by the fact that microaveraged effectiveness is dominated by the performance of the classifiers on the most frequent categories. To see this, note that microaveraged F_1

TABLE 4. Values of relative hardness of Reuters-21578 subsets as derived from the average effectiveness values of Table 3. The value in a given entry measures how easier the subset in the row proved with respect to the subset in the column.

	Microaveraging			Macroaveraging		
	R(10)	R(90)	R(115)	R(10)	R(90)	R(115)
R(10)	—	+8.2%	+8.6%	—	+46.8%	+44.6%
R(90)	−7.6%	—	+0.3%	−53.1%	—	−5.2%
R(115)	−7.9%	−0.3%	—	−50.5%	+5.5%	—

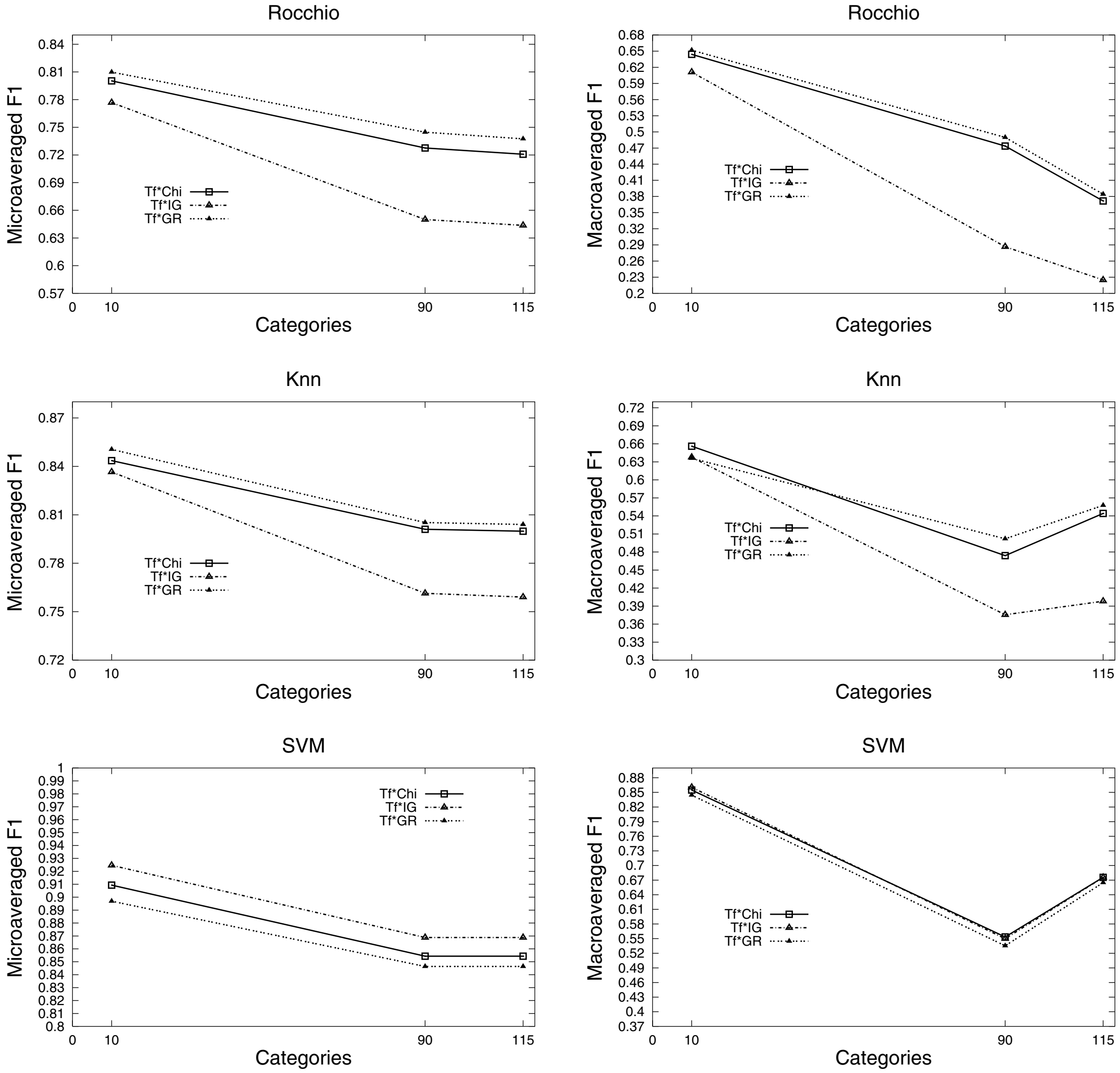

FIG. 4. Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with supervised weighting and a $\xi = 0.90$ reduction factor**. Plots indicate results obtained with Rocchio (top), *k*-NN (middle), and SVMs (bottom). The *X* axis indicates the three subsets of Reuters-21578.

is an increasing function of microaveraged precision and microaveraged recall, and that

- Microaveraged recall is the proportion of correct positive classification decisions that are indeed taken, and most correct positive classification decisions by definition concern categories that have many positive test examples. In Reuters-21578 the 10 categories that have the highest number of positive *test* examples are (unsurprisingly, given that the train/test partition was obtained by a random split) the same categories that have the highest number of positive *training* examples, i.e., are the categories in R(10). Note that the 10 categories in R(10) have altogether 2787 test examples, while the other 80 categories in R(90) have altogether just 957 of them; this shows that the former set of categories contributes three times as much as the latter in determining microaveraged recall on R(90).
- Microaveraged precision is the proportion of the positive classification decisions taken that are indeed correct, and it can be expected that most positive classification decisions taken concern categories that have many positive test examples, which are, as noted above, the same categories that have many positive training examples.[7]

As a result, the microaveraged performance obtained on R(90) is heavily influenced by the performance obtained on the 10 most frequent categories, and much less heavily by the performance obtained on the remaining 80 categories. This explains why the above-mentioned decrease in microaveraged effectiveness is not very sharp. Instead, macroaveraged effectiveness is, by definition, not dominated by any category in particular. Because each of the 80 least frequent categories counts the same as any of the 10 most frequent ones, the fact that the former categories are more difficult

[7] The fact that most positive classification decisions taken concern categories that have many positive test examples, rather than being just an intuitively likely fact, is a fact that the proportional thresholding policy we have adopted explicitly seeks to bring about.

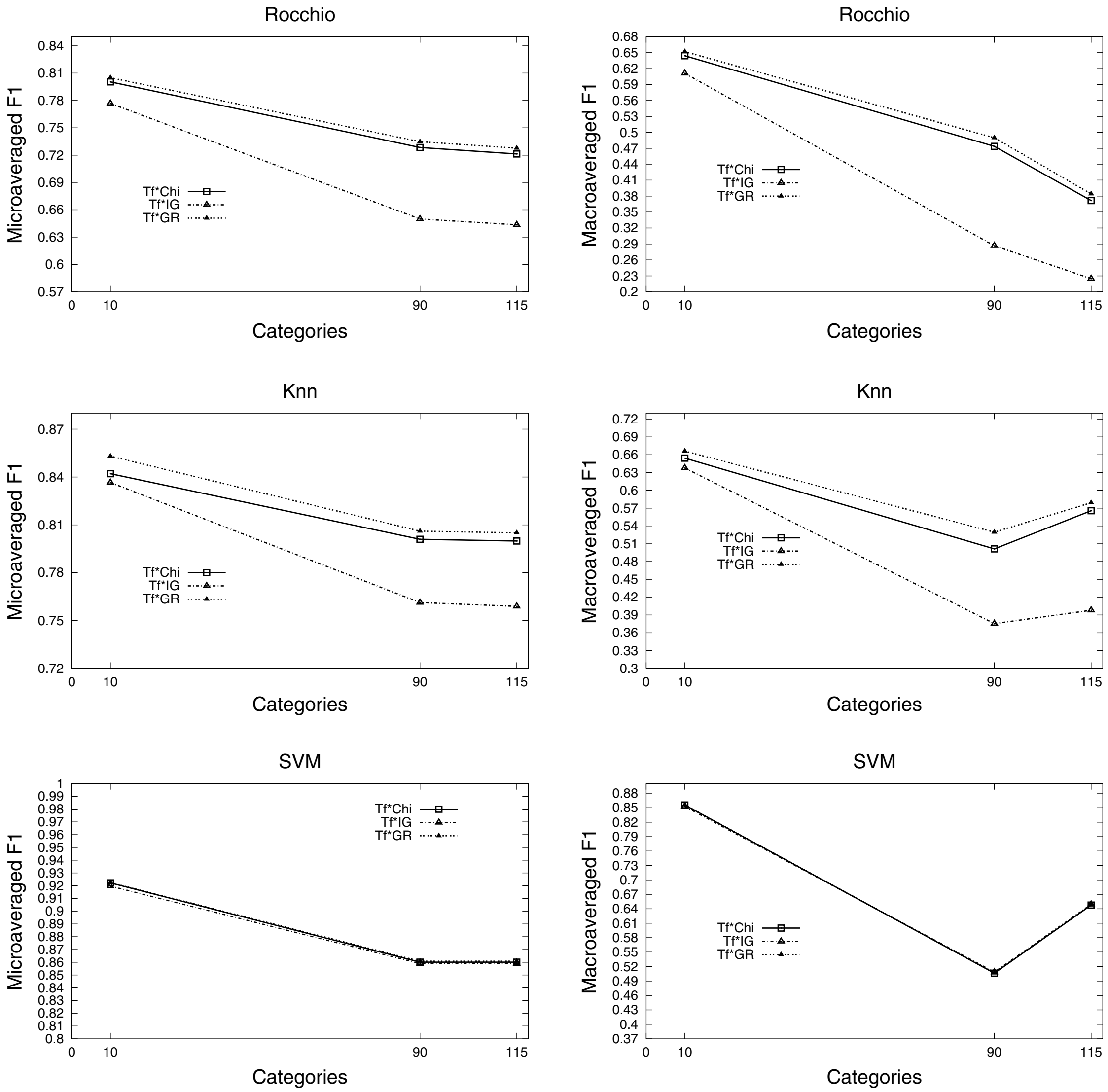

FIG. 5. Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) **obtained with supervised weighting and a $\xi = 0.50$ reduction factor**. Plots indicate results obtained with Rocchio (top), *k*-NN (middle), and SVMs (bottom). The *X* axis indicates the three subsets of Reuters-21578.

than the latter[8] weighs heavily on macroaveraged effectiveness, and the decrease in performance is more marked.

A second fact that also emerges clearly from the experiments is that R(115) is not significantly harder than R(90) when effectiveness is computed through microaveraging (−0.3%), while it is even easier (+5.5%) if macroaveraging is used. Both facts seem, on the surface, surprising, because the 25 additional categories have on average much fewer training examples (2.52 each) than the other 90 (107 each). However, arguments similar to the ones espoused above show that there is indeed a rationale for this. Microaveraged effectiveness is marginally hurt by the performance obtained on the 25 additional categories, because these categories contain no positive test examples: this means that microaveraged recall is by definition unaffected, while microaveraged precision is (for the same reasons discussed regarding macroaveraged precision) hurt only scarcely.

The fact that macroaveraged effectiveness even *benefits* from the added 25 categories is less obvious, but can be explained by the following fact. The value of F_{1_i} is equal to 1 for each category c_i on which no negative test examples are incorrectly classified under c_i (it is 0 otherwise). In order for this to happen, the threshold τ_i needs to be set high enough that for no test document d_j the CSV will exceed it. This indeed happens frequently, because the validation set on which τ_i is tuned (see the earlier section on classifier learning) also contains very few positive examples (if any—these 25 categories have, on average, 2.52 training *or* validation examples); this means that, to correctly classify the validation examples, high values for τ_i tend to be chosen.

[8] The 10 most frequent categories have, on average, 719.3 training examples each, while the 80 least frequent ones have, on average, 29.9 training examples each.

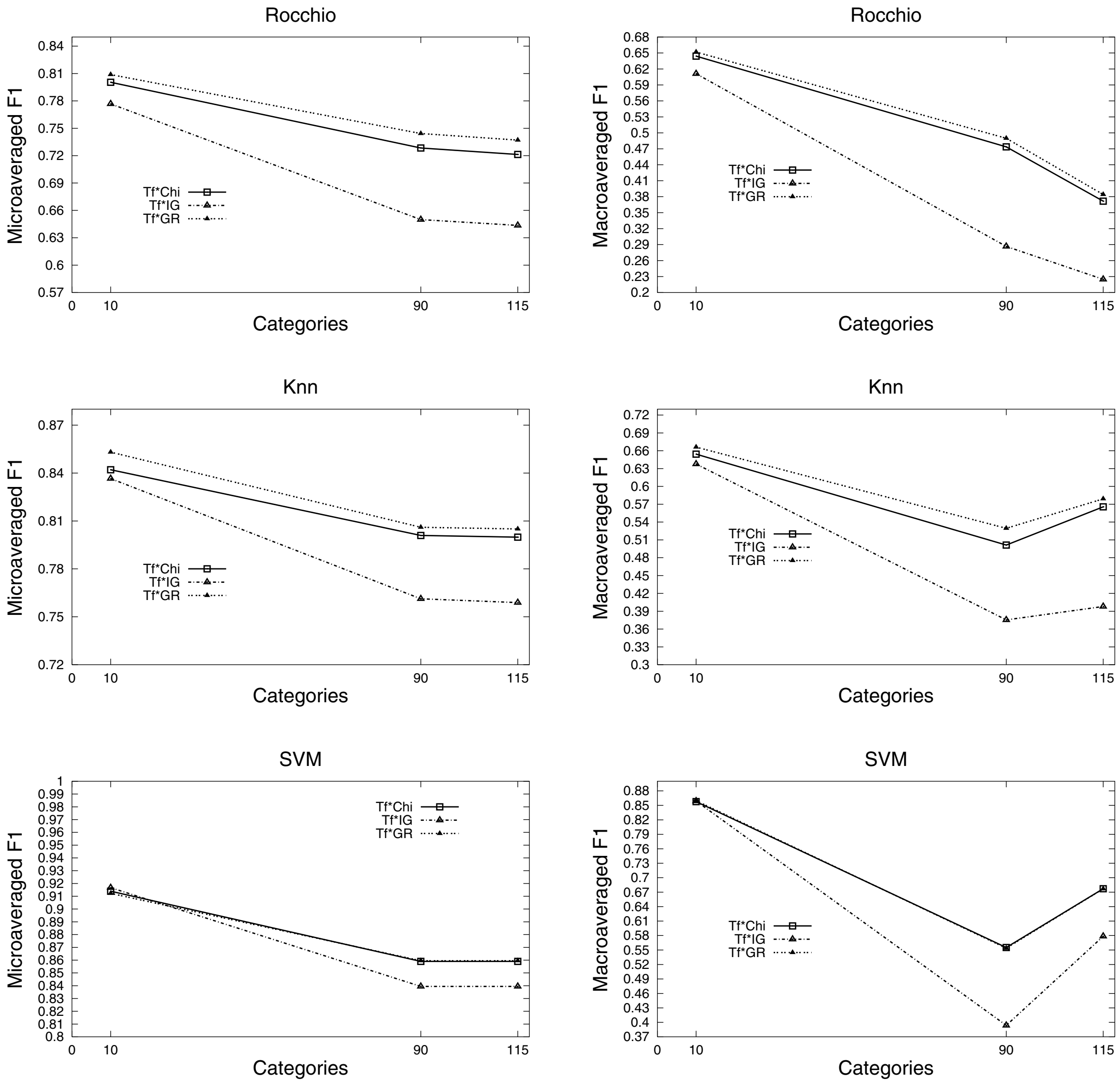

FIG. 6. Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with supervised weighting and a $\xi = 0.0$ reduction factor**. Plots indicate results obtained with Rocchio (top), *k*-NN (middle), and SVMs (bottom). The *X* axis indicates the three subsets of Reuters-21578.

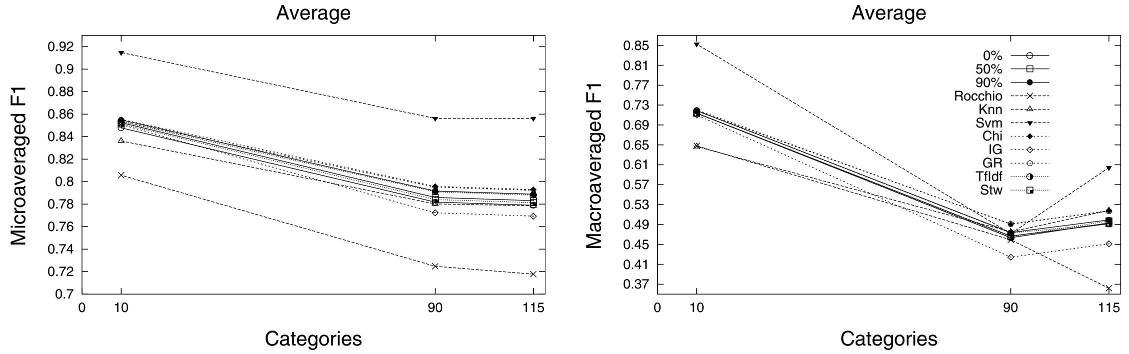

FIG. 7. Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained by averaging across term weighting policies, feature selection policies, feature selection functions, reduction factors for feature selection, and learning methods. The *X* axis indicates the three subsets of Reuters-21578.

As can be seen from Figure 7, the R(115) macroaveraged F_1 value for Rocchio represents the only exception to the general trend; this is obviously responsible for the fact that the R(115) macroaveraged F_1 entry in Table 3 is the one with the highest value in standard deviation. The likely explanation of the fact that Rocchio, unique among the studied learners, performs better on R(90) than on R(115) can probably be found in Rocchio's notoriously "crude" learning method (i.e., plain, unsophisticated generation of a centroid vector, with no attempt at margin maximization), which makes it particularly unsuitable to deal with "hard" categories comprising very few positive training examples (2.52 on average, in our case).

A fact that emerges clearly from the low values of standard deviation reported in Table 3 is that these conclusions are largely independent of the techniques employed, regardless of whether they are concerned with learning, or feature selection, or weighting, and so on. Figure 7 tells us that, while for macroaveraging some exceptions to the general trend do exist (e.g., the above-mentioned macroaveraged performance of Rocchio on R(115)), microaveraging displays little or no variance across different techniques. This suggests that our conclusions are fairly reliable, even if this degree of reliability cannot formally be measured.[9]

Conclusion

We have presented a systematic, comparative experimental study of the three most popular subsets of Reuters-21578, itself the most popular test collection of text categorization research. We have carried out experiments on a variety of experimental contexts, including all possible combinations of three learning methods, three term selection functions, three term selection reduction factors, two term weighting policies, and two effectiveness functions. The results we have obtained are thus fairly representative of the relative hardness of the three Reuters-21578 subsets, also as a result of the fact that the design choices that we have tested are widely different among each other and, at the same time, widely used in the text categorization literature. We have also presented theoretical, *a posteriori* justifications for these results, in particular explaining (1) why the decrease in performance that can be expected in going from R(10) to R(90) is sharper for macroaveraging than for microaveraging, and (2) why in going from R(90) to R(115) we may expect almost no decrease in microaveraged performance, and even an increase in macroaveraged performance.

The cumulative results we have obtained, which are conveniently summarized in Table 4, finally allow the comparison, albeit indirect, of different text classifiers, which in individual experiments, had been or will be tested by their proponents on different Reuters-21578 subsets.

Acknowledgments

We thank the anonymous referees for valuable comments. The work of the second author has been partially supported by the MIUR grant no. 20033091149 005 and by the Onto Text project funded by the Provincia Autonoma di Trento.

References

Baker, L.D., & McCallum, A.K. (1998). Distributional clustering of words for text classification. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, & J. Zobel (Eds.), Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval (pp. 96–103). New York: ACM Press.

Benkhalifa, M., Mouradi, A., & Bouyakhf, H. (2001). Integrating external knowledge to supplement training data in semi-supervised learning for text categorization. Information Retrieval, 4(2), 91–113.

Bennett, P.N. (2003). Using asymmetric distributions to improve text classifier probability estimates. In J. Callan, G. Cormack, C. Clarke, D. Hawking, & A. Smeaton (Eds.), Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval (pp. 111–118). New York: ACM Press.

Bennett, P.N., Dumais, S.T., & Horvitz, E. (2002). Probabilistic combination of text classifiers using reliability indicators: Models and results. In M. Beaulieu, R. Baeza-Yates, S.H. Myaeng, & K. Järvelin (Eds.), Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval (pp. 207–214). New York: ACM Press.

Caropreso, M.F., Matwin, S., & Sebastiani, F. (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 (pp. 78–102). Hershey, PA: Idea Group Publishing.

Chai, K.M., Ng, H.T., & Chieu, H.L. (2002). Bayesian online classifiers for text classification and filtering. In M. Beaulieu, R. Baeza-Yates, S.H. Myaeng, & K. Järvelin (Eds.), Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval (pp. 97–104). New York: ACM Press.

Crammer, K., & Singer, Y. (2002). A new family of online algorithms for category ranking. In M. Beaulieu, R. Baeza-Yates, S.H. Myaeng, & K. Järvelin (Eds.), Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval (pp. 151–158). New York: ACM Press.

Debole, F., & Sebastiani, F. (2003). Supervised term weighting for automated text categorization. In Proceedings of SAC-03, 18th ACM Symposium on Applied Computing (pp. 784–788). New York: ACM Press.

Dumais, S.T., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In G. Gardarin, J.C. French, N. Pissinou, K. Makki, & L. Bouganim (Eds.), Proceedings of CIKM-98, Seventh ACM International Conference on Information and Knowledge Management (pp. 148–155). New York: ACM Press.

Galavotti, L., Sebastiani, F., & Simi, M. (2000). Experiments on the use of feature selection and negative evidence in automated text categorization. In J.L. Borbinha & T. Baker (Eds.), Proceedings of ECDL-00, Fourth European Conference on Research and Advanced Technology for Digital Libraries (Lecture Notes in Computer Science series, No. 1923, pp. 59–68). Heidelberg, Germany: Springer Verlag.

Gao, S., Wu, W., Lee, C-H., & Chua, T-S. (2003). A maximal figure of-merit learning approach to text categorization. In J. Callan, G. Cormack, C. Clarke, D. Hawking, & A. Smeaton (Eds.), Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval (pp. 174–181). New York: ACM Press.

Hayes, P.J., & Weinstein, S.P. (1990). Construe/Tis: a system for content-based indexing of a database of news stories. In A. Rappaport & R. Smith (Eds.), Proceedings of IAAI-90, Second Conference on Innovative Applications of Artificial Intelligence (pp. 49–66). Menlo Park, CA: AAAI Press.

[9] It cannot formally be measured because it is not formally defined how representative the chosen learning methods are of the population of possible learning methods that could be used in a text categorization experiment; obviously, the same goes for term weighting policies, feature selection policies, feature selection functions, and reduction factors for feature selection.

Hull, D.A. (1994). Improving text retrieval for the routing problem using latent semantic indexing. In W.B. Croft & C.J. van Rijsbergen (Eds.), Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval (pp. 282–289). Heidelberg, Germany: Springer Verlag.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In C. N'edellec & C. Rouveirol (Eds.), Proceedings of ECML-98, 10th European Conference on Machine Learning (Lecture Notes in Computer Science series, No. 1398, pp. 137–142). Heidelberg, Germany: Springer Verlag.

Joachims, T. (1999). Making large-scale SVM learning practical. In B. Schölkopf, C.J. Burges, & A.J. Smola (Eds.), Advances in kernel methods–support vector learning (Chap. 11, pp. 169–184). Cambridge, MA: The MIT Press.

Lam, W., & Lai, K-Y. (2001). A meta-learning approach for text categorization. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval (pp. 303–309). New York: ACM Press.

Lewis, D.D. (1992a). An evaluation of phrasal and clustered representations on a text categorization task. In N.J. Belkin, P. Ingwersen, & A.M. Pejtersen (Eds.), Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval (pp. 37–50). New York: ACM Press.

Lewis, D.D. (1992b). Representation and learning in information retrieval. Unpublished doctoral dissertation, University of Massachusetts, Amherst.

Lewis, D.D. (1995). Evaluating and optimizing autonomous text classification systems. In E.A. Fox, P. Ingwersen, & R. Fidel (Eds.), Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval (pp. 246–254). New York: ACM Press

Lewis, D.D., Li, F., Rose, T., & Yang, Y. (2004, April). Reuters Corpus Volume 1 as a text categorization test collection. Journal of Machine Learning Research, 5, 361–397.

Lewis, D.D., Schapire, R.E., Callan, J.P., & Papka, R. (1996). Training algorithms for linear text classifiers. In H-P. Frei, D. Harman, P. Schäuble, & R. Wilkinson (Eds.), Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval (pp. 298–306). New York: ACM Press.

Li, H., & Yamanishi, K. (1999). Text classification using ESC-based stochastic decision lists. In Proceedings of CIKM-99, Eighth ACM International Conference on Information and Knowledge Management (pp. 122–130). New York: ACM Press.

McCallum, A.K., & Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In Proceedings of the First AAAI Workshop on Learning for Text Categorization (pp. 41–48).

Nardiello, P., Sebastiani, F., & Sperduti, A. (2003). Discretizing continuous attributes in AdaBoost for text categorization. In F. Sebastiani, (Ed.), Proceedings of ECIR-03, 25th European Conference on Information Retrieval (pp. 320–334). Heidelberg, Germany: Springer Verlag.

Nigam, K., McCallum, A.K., Thrun, S., & Mitchell, T.M. (2000). Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2/3), 103–134.

Rose, T., Stevenson, M., & Whitehead, M. (2002). The Reuters Corpus Volume 1–from yesterday's news to tomorrow's language resources. In Proceedings of LREC-02, Third International Conference on Language Resources and Evaluation (pp. 827–832).

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 1–47.

Sebastiani, F., Sperduti, A., & Valdambrini, N. (2000). An improved boosting algorithm and its application to automated text categorization. In A. Agah, J. Callan, & E. Rundensteiner (Eds.), Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management (pp. 78–85). New York: ACM Press.

Tong, S., & Koller, D. (2001, November). Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, 2, 45–66.

Toutanova, K., Chen, F., Popat, K., & Hofmann, T. (2001). Text classification in a hierarchical mixture model for small training sets. In H. Paques, L. Liu, & D. Grossman (Eds.), Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management (pp. 105–113). New York: ACM Press.

Yang, Y. (2001). A study on thresholding strategies for text categorization. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval (pp. 137–145). New York: ACM Press.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. In M.A. Hearst, F. Gey, & R. Tong, (Eds.), Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval (pp. 42–49). New York: ACM Press.