An Analysis of the Relative Hardness of Reuters-21578 Subsets

Franca Debole & 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

Abstract

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, since 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 ten 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 paper 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.

1 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, since 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 ten years[1]. Reuters-21578 is a set of 21,578 news stories 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:

- similarly to many other applicative contexts, it is multi-label, 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

[1] While a new Reuters corpus has recently been made available for TC research [27, 20], its takeup has been somehow slow, and also hindered by terms of use that are not universally acceptable by interested parties. For instance, it has been reported that some universities in the US are not willing to sign the "licence 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.

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, since several categories have (under any possible split between training and test documents) very few 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 this collection, and tested their systems on one of these subcollections only. The most frequent direction for extracting a 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 paper we present a systematic, comparative experimental study of the above-mentioned three subsets of Reuters-21578. 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,

[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 Section 3 for more details.

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 paper is structured as follows. Section 2 briefly introduces the TC task and the related terminology, thus setting the stage for the description of our experimental work. In Section 3 we describe in some detail the Reuters-21578 test collection and the subsets of it that have been used most often in TC research. Section 4 presents a systematic experimental study in which we test the relative hardness of these subsets and give theoretical justifications for these results. Section 5 concludes.

2 Preliminaries: an introduction to text categorization

Text categorization (TC – aka *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 *multi-label* (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 $\overline{c}_i$. Multi-label TC under $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ is usually tackled as $|\mathcal{C}|$ independent binary classification problems under $\{c_i, \overline{c}_i\}$, for $i = 1, \ldots, |\mathcal{C}|$. A *classifier for* c_i is then a function $\hat{\Phi}_i : \mathcal{D} \rightarrow \{T, F\}$ that approximates the unknown target function $\Phi_i : \mathcal{D} \rightarrow \{T, F\}$. Multi-label (and, as a consequence, binary) TC, rather than single-label TC, will be the focus of

this paper.

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 Sections 5, 6 and 7, respectively, of [28].

2.1 Document indexing

Document indexing denotes the mapping of a document d_j into a compact representation of its content that can be directly interpreted (i) by a classifier-building algorithm and (ii) 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* (aka *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 (i) a definition of what a term is, and (ii) a method to compute term weights. Concerning (i), 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 pre-processing phase), or with their *stems* (i.e. their morphological roots, obtained by applying a stemming algorithm). Concerning (ii), 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 has both effects of reducing *overfitting* (i.e. the tendency of the classifier to better classify the data it has been trained on than new unseen data), and of making the problem more manageable for the learning method, since 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 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.

2.2 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 pre-classified under c_i or $\overline{c}_i$, gleans the characteristics that a new unseen document should have in order to belong to c_i. In order 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 observing which the learner builds the classifier, while 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 $Tr - Va$, and then 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.

2.3 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 behaviour) are measures of success for a TC system. However, effectiveness is considered the most important criterion, since 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, since 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. When effectiveness is computed for several categories, the results for individual categories must be averaged in some way; here, one may opt for

	Microaveraging	**Macroaveraging**
Precision (π)	$\pi = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i + FP_i}$	$\pi = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \pi_i}{\lvert\mathcal{C}\rvert} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \frac{TP_i}{TP_i + FP_i}}{\lvert\mathcal{C}\rvert}$
Recall (ρ)	$\rho = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i}{\sum_{i=1}^{\lvert\mathcal{C}\rvert} TP_i + FN_i}$	$\rho = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \rho_i}{\lvert\mathcal{C}\rvert} = \frac{\sum_{i=1}^{\lvert\mathcal{C}\rvert} \frac{TP_i}{TP_i + FN_i}}{\lvert\mathcal{C}\rvert}$

Table 1: Averaging precision and recall across different categories; TP_i, TN_i, FP_i and FN_i refer to the sets of *true positives wrt* c_i (documents correctly deemed to belong to c_i), *true negatives wrt* c_i (documents correctly deemed not 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.

microaveraging ("categories count proportionally to the number of their positive test examples") or for *macroaveraging* ("all categories count the same"), 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 and recall, in both their microaveraging and macroaveraging variants. Since a classifier can be tuned to emphasize precision at the expense of recall, or viceversa, only combinations of the two are significant, the most popular combination nowadays being $F_1 = \frac{2\pi\rho}{\pi+\rho}$ [19].

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

3 The Reuters-21578 collection and its subsets

The data contained in the "Reuters-21578, Distribution 1.0" corpus consist of news stories appeared on the Reuters newswire in 1987[3]. The data was originally labelled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system [12], 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 up until 1996, when a revision of the collection resulted in the correction of several other errors and 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.

The Reuters-21578 documents actually used in TC experiments are only 12,902, since the creators of the collection found ample evidence that the other 8,676 documents had not been considered for labelling by the people who manually assigned categories to documents ("indexers"). In order 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 paper we will always refer to the ModApté split.

There are 5 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, since 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 unlabelled because the indexers deemed

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

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, since 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.

Since 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, $\mathsf{R(10)}$). Among others, this has been used in [3, 4, 9, 23, 26, 30].

- The set of 90 categories with at least one positive training example and one test example (hereafter, $\mathsf{R(90)}$). This appears to be the most frequently chosen subset; among others, it has been used in [1, 6, 7, 11, 14, 16, 22, 24, 29, 31, 33].

- The set of 115 categories with at least one positive training example ($\mathsf{R(115)}$). Among others, this has been used in [2, 5, 9, 10, 25].

It follows from this discussion that $\mathsf{R(10)} \subset \mathsf{R(90)} \subset \mathsf{R(115)}$.

Reasons for using one or the other subset have been different. Several researchers claim that $\mathsf{R(10)}$ is more realistic since machine learning techniques cannot perform adequately when positive training examples are scarce, and/or since small numbers of positive test examples make the interpretation of effectiveness results problematic due to high variance. Other researchers claim instead that only by striving to work on infrequent categories too we can hope to push the limits of TC technology, and this consideration leads them to use

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

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 exactly *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.

4 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 the following choices:

- As for the *learning methods*, we have used a choice among (i) a standard Rocchio method [13] for learning linear classifiers, (ii) a standard k-NN algorithm [33], and (iii) the support vector machine (SVM) learner as implemented in the SVMLIGHT package (version 3.5) [15]. For reasons of brevity we do not discuss these methods in detail; the interested reader will find detailed presentations of them in [8].

- 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 [8]. Each of the three functions has been used according to the global policy described in Section 2.1, essentially for efficiency reasons[5]. Globalization has been achieved by means of the f_{max}

[5]For instance, recall that the k-NN learner computes, for each test document d_j, its similarity with each training document, and then ranks these training documents in terms of the computed similarity 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

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 [8], consisting in replacing the idf component of $tf * idf$ with the function that, in the same experiment, has been previously used for term selection (this yields e.g. cosine-normalized $tf * GR$ if GR has been previously used for feature selection). For reasons of brevity we do not discuss these policies in detail; the interested reader will find detailed presentations of them in [8].

- 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 $F_1 = 1$, since the classifier always returns the correct decision [21].

In all the experiments discussed in this paper, stop words have been removed using the stop list provided in [18, pages 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.

4.1 Experimental results

The results of these experiments are reported in Figures 1 to 6. Each table includes six plots: the leftmost plots concern microaveraging, while the rightmost

to be performed $|\mathcal{C}|$ times if the local policy is used, since in this case the same document has $|\mathcal{C}|$ different representations, and similarity scores (and rankings) thus vary across categories.

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

Table 2: Term selection functions used in this work.

concern macroaveraging; results obtained with the Rocchio, k-NN, and SVM learners are displayed in the top, mid, and bottom row, respectively. Each individual plot includes three curves, each corresponding to a feature selection function (chosen among IG, GR, and χ^2). The six figures report these results for each combination of a term weighting policy (chosen among $tf * idf$ and supervised term weighting) and a feature reduction factor (chosen among $\xi \in \{0.90, 0.50, 0.0\}$).

Figure 7 summarizes these results by averaging them for each studied technique; for instance, the curve marked "SVM" reports average results of all experiments run with the SVM learner, thus averaging across all term weighting policies, feature selection policies, feature selection functions, and reduction factors for feature selection. Separate plots for microaveraging and macroaveraging are given. Table 3 reports figures obtained by averaging across *all* the reported experiments; each numeric value is the result of averaging across 48 different experiments, and can thus be considered fairly representative. Finally, Table 4 reinterprets the results of Table 3 in terms of the relative hardness of the three Reuters-21578 subsets studied.

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, etc. This was largely to be expected, given that

	Microaveraged F_1	**Microaveraged F_1**
R(10)	0.85223540	0.72393364
R(90)	0.78707075	0.52659655
R(115)	0.78421640	0.57822156

Table 3: Effectiveness averaged across all the text classifiers tested in our experiments on the three Reuters-21578 subsets.

	Microaveraging			Macroaveraging		
	R(10)	R(90)	R(115)	R(10)	R(90)	R(115)
R(10)	–	+8.2%	+8.6%	–	+37.4%	+25.2%
R(90)	-7.6%	–	+0.3%	-27.2%	–	-8.9%
R(115)	-7.9%	-0.3%	–	-20.1%	+9.8%	–

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.

its categories are the ones with the highest number of positive examples, and as such allow taming the "curse of dimensionality" more effectively.

On average, the decrease in performance in going from R(10) to R(90) is much sharper for macroaveraging (-27.2%) 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. In fact (see Table 1):

- 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, which in Reuters-21578 are (unsurprisingly, given that the train/test partition was obtained by a random split) the same categories that have many positive *training* examples. 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[6].

As a result, the microaveraged performance obtained on R(90) is heavily influenced by the performance obtained on the 10 most frequent categories, and

[6]Note also that in optimizing the thresholds for our learners (see Section 2.2) we have used the well-known *proportional thresholding method* [17, 32], according to which for the threshold τ_i we choose 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. This means that 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 our thresholding policy explicitly seeks to bring about.

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. Since 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 than the latter[7] 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 (+9.8%) if macroaveraging is used. Both facts seem, on the surface, surprising, since 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 expoused above show that there is indeed a rationale for this. Microaveraged effectiveness is marginally hurt by the performance obtained on the 25 additional categories, since 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 below re: macroaveraged precisions) hurt only scarcely.

The fact that macroaveraged effectiveness even *benefits* from the added 25 categories is less obvious, but can be explained by two facts:

- Macroaveraged recall is trivially equal to 1 on all of these categories.
- Macroaveraged precision is 1 for each category c_i on which no negative test examples are incorrectly classified in 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, since the validation set on which τ_i is tuned (see Section 2.2) also contains very few positive examples (if any – these 25 categories have, on average, 2.52 training *or* validation examples); this means that, in order to correctly

[7]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.

classify the validation examples, high values for τ_i tend to be chosen.

A third fact that also emerges clearly (see Figure 7) is that these conclusions are largely independent of the techniques employed, regardless of whether they are concerned with learning, or feature selection, or weighting, etc. While for macroaveraging some exceptions to the general trend do exist (e.g. the Rocchio learner performs worse on R(115) than on R(90)), microaveraging displays little or no variance among different techniques.

5 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 (i) 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 (ii) 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.

References

[1] L. Douglas Baker and Andrew K. McCallum. Distributional clustering of words for text classification. In W. Bruce Croft, Alistair Moffat, Cornelis J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US.

[2] Mohammed Benkhalifa, Abdelhak Mouradi, and Houssaine Bouyakhf. Integrating external knowledge to supplement training data in semi-supervised learning for text categorization. *Information Retrieval*, 4(2):91–113, 2001.

[3] Paul N. Bennett. Using asymmetric distributions to improve text classifier probability estimates. In Jamie Callan, Gordon Cormack, Charles Clarke, David Hawking, and Alan Smeaton, editors, *Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval*, pages 111–118, Toronto, CA, 2003. ACM Press, New York, US.

[4] Paul N. Bennett, Susan T. Dumais, and Eric Horvitz. Probabilistic combination of text classifiers using reliability indicators: models and results. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalervo Järvelin, editors, *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pages 207–214, Tampere, FI, 2002. ACM Press, New York, US.

[5] Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In Amita G. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, US, 2001.

[6] Kian M. Chai, Hwee T. Ng, and Hai L. Chieu. Bayesian online classifiers for text classification and filtering. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalervo Järvelin, editors, *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pages 97–104, Tampere, FI, 2002. ACM Press, New York, US.

[7] Koby Crammer and Yoram Singer. A new family of online algorithms for category ranking. In Micheline Beaulieu, Ricardo Baeza-Yates, Sung Hyon Myaeng, and Kalervo Järvelin, editors, *Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval*, pages 151–158, Tampere, FI, 2002. ACM Press, New York, US.

[8] Franca Debole and Fabrizio Sebastiani. Supervised term weighting for automated text categorization. In *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing*, pages 784–788, Melbourne, US, 2003. ACM Press, New York, US.

[9] Susan T. Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In Georges Gardarin, James C. French, Niki Pissinou, Kia Makki, and Luc 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.

[10] Luigi Galavotti, Fabrizio Sebastiani, and Maria Simi. Experiments on the use of feature selection and negative evidence in automated text categorization. In José L. Borbinha and Thomas Baker, editors, *Proceedings of ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 59–68, Lisbon, PT, 2000. Springer Verlag, Heidelberg, DE. Published in the "Lecture Notes in Computer Science" series, number 1923.

[11] Sheng Gao, Wen Wu, Chin-Hui Lee, and Tat-Seng Chua. A maximal figure-of-merit learning approach to text categorization. In Jamie Callan, Gordon Cormack, Charles Clarke, David Hawking, and Alan Smeaton, editors, *Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval*, pages 174–181, Toronto, CA, 2003. ACM Press, New York, US.

[12] Philip J. Hayes and Steven P. Weinstein. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In Alain Rappaport and Reid Smith, editors, *Proceedings of IAAI-90, 2nd Conference on Innovative Applications of Artificial Intelligence*, pages 49–66. AAAI Press, Menlo Park, US, 1990.

[13] David A. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In W. Bruce Croft and Cornelis J. van Rijsbergen, editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 282–289, Dublin, IE, 1994. Springer Verlag, Heidelberg, DE.

[14] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE. Published in the "Lecture Notes in Computer Science" series, number 1398.

[15] Thorsten Joachims. Making large-scale SVM learning practical. In Bernhard Schölkopf, Christopher J. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 11, pages 169–184. The MIT Press, Cambridge, US, 1999.

[16] Wai Lam and Kwok-Yin Lai. A meta-learning approach for text categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International*

Conference on Research and Development in Information Retrieval, pages 303–309, New Orleans, US, 2001. ACM Press, New York, US.

[17] David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In Nicholas J. Belkin, Peter Ingwersen, and Annelise Mark 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.

[18] David D. Lewis. *Representation and learning in information retrieval*. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US, 1992.

[19] David D. Lewis. Evaluating and optmizing autonomous text classification systems. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, US, 1995. ACM Press, New York, US.

[20] David D. Lewis, Fan Li, Tony Rose, and Yiming Yang. Reuters Corpus Volume I as a text categorization test collection. *Journal of Machine Learning Research*, 2003. Forthcoming.

[21] David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In Hans-Peter Frei, Donna Harman, Peter Schäuble, and Ross Wilkinson, editors, *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 298–306, Zürich, CH, 1996. ACM Press, New York, US.

[22] Hang Li and Kenji Yamanishi. Text classification using ESC-based stochastic decision lists. In *Proceedings of CIKM-99, 8th ACM International Conference on Information and Knowledge Management*, pages 122–130, Kansas City, US, 1999. ACM Press, New York, US.

[23] Andrew K. McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *Proceedings of the 1st AAAI Workshop on Learning for Text Categorization*, pages 41–48, Madison, US, 1998.

[24] Alessandro Moschitti. A study on optimal parameter tuning for Rocchio text classifier. In Fabrizio Sebastiani, editor, *Proceedings of ECIR-03, 25th European Conference on Information Retrieval*, pages 420–435, Pisa, IT, 2003. Springer Verlag.

[25] Pio Nardiello, Fabrizio Sebastiani, and Alessandro Sperduti. Discretizing continuous attributes in AdaBoost for text categorization. In Fabrizio Sebastiani, editor, *Proceedings of ECIR-03, 25th European Conference on Information Retrieval*, pages 320–334, Pisa, IT, 2003. Springer Verlag.

[26] Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[27] Tony Rose, Mark Stevenson, and Miles Whitehead. The Reuters Corpus Volume 1 – from yesterday's news to tomorrow's language resources. In *Proceedings of LREC-02, 3rd International Conference on Language Resources and Evaluation*, pages 827–832, Las Palmas, ES, 2002.

[28] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[29] Fabrizio Sebastiani, Alessandro Sperduti, and Nicola Valdambrini. An improved boosting algorithm and its application to automated text categorization. In Arvin Agah, Jamie Callan, and Elke Rundensteiner, editors, *Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 78–85, McLean, US, 2000. ACM Press, New York, US.

[30] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, November 2001.

[31] Kristina Toutanova, Francine Chen, Kris Popat, and Thomas Hofmann. Text classification in a hierarchical mixture model for small training sets. In Henrique Paques, Ling Liu, and David Grossman, editors, *Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management*, pages 105–113, Atlanta, US, 2001. ACM Press, New York, US.

[32] Yiming Yang. A study on thresholding strategies for text categorization. In W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel, editors, *Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval*, pages 137–145, New Orleans, US, 2001. ACM Press, New York, US.

[33] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In Marti A. Hearst, Fredric Gey, and Richard Tong, editors, *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, US, 1999. ACM Press, New York, US.

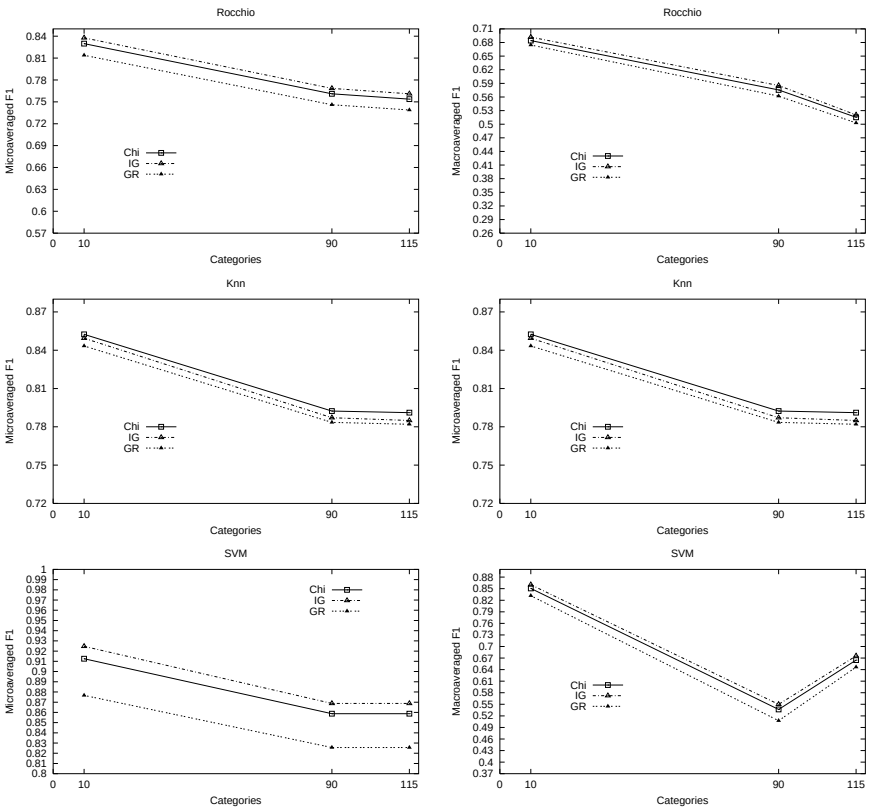

Figure 1: Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with** $tf * idf$ **weighting and a** $\xi = 0.90$ **reduction factor**. Plots indicate results obtained with Rocchio (top), k-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578 described in Section 3.

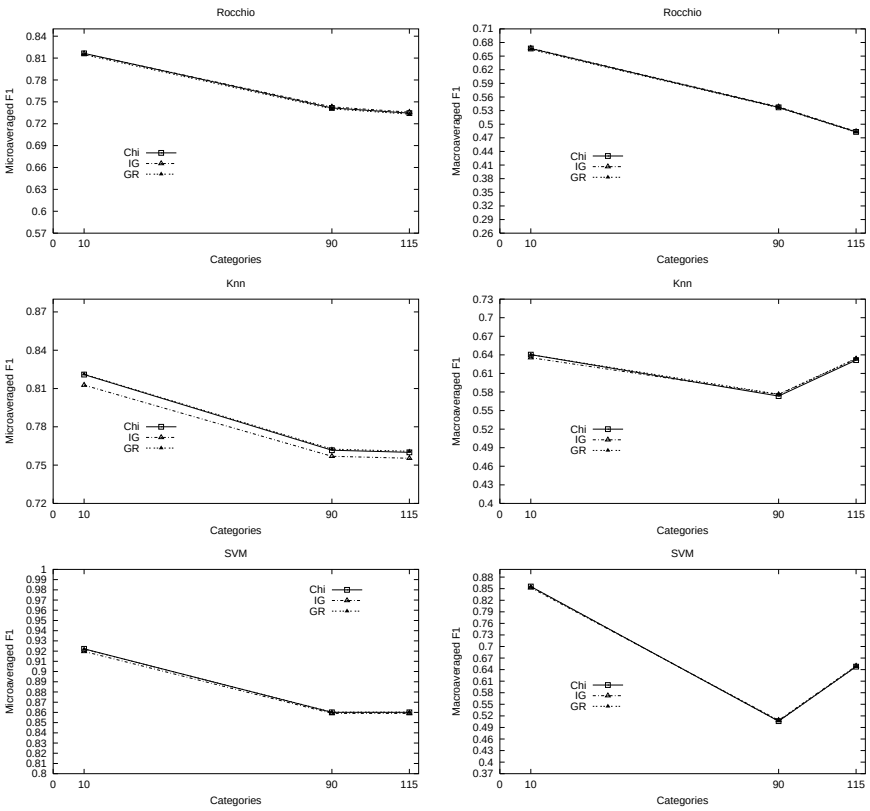

Figure 2: Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with** $tf * idf$ **weighting and a** $\xi = 0.50$ **reduction factor**. Plots indicate results obtained with Rocchio (top), k-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578 described in Section 3.

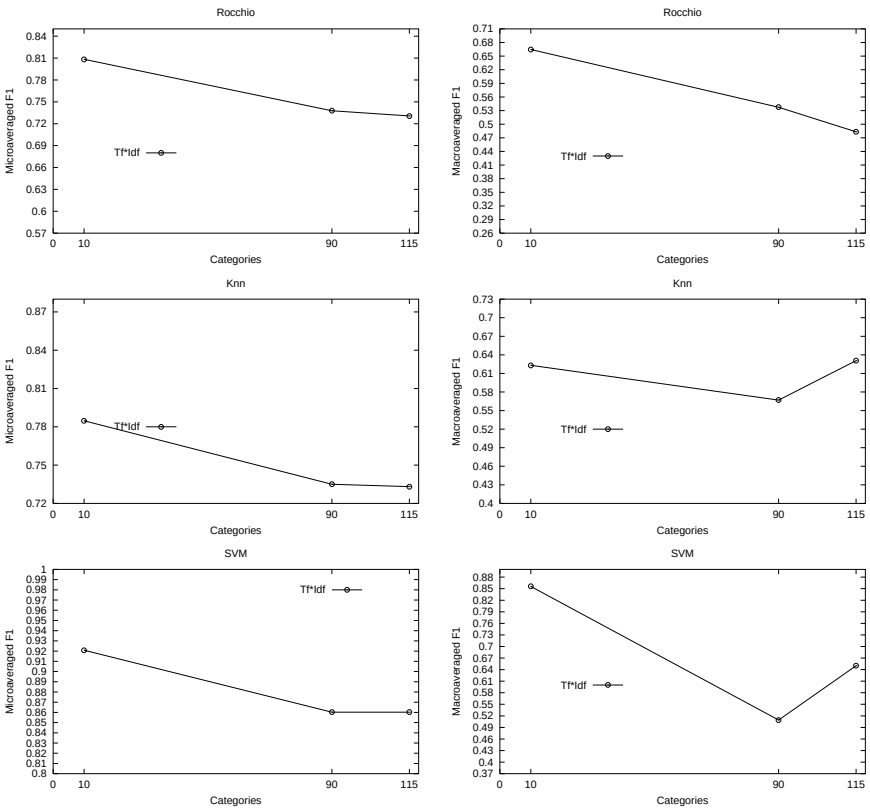

Figure 3: Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with** $tf * idf$ **weighting and a** $\xi = 0.0$ **reduction factor**. Plots indicate results obtained with Rocchio (top), k-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578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

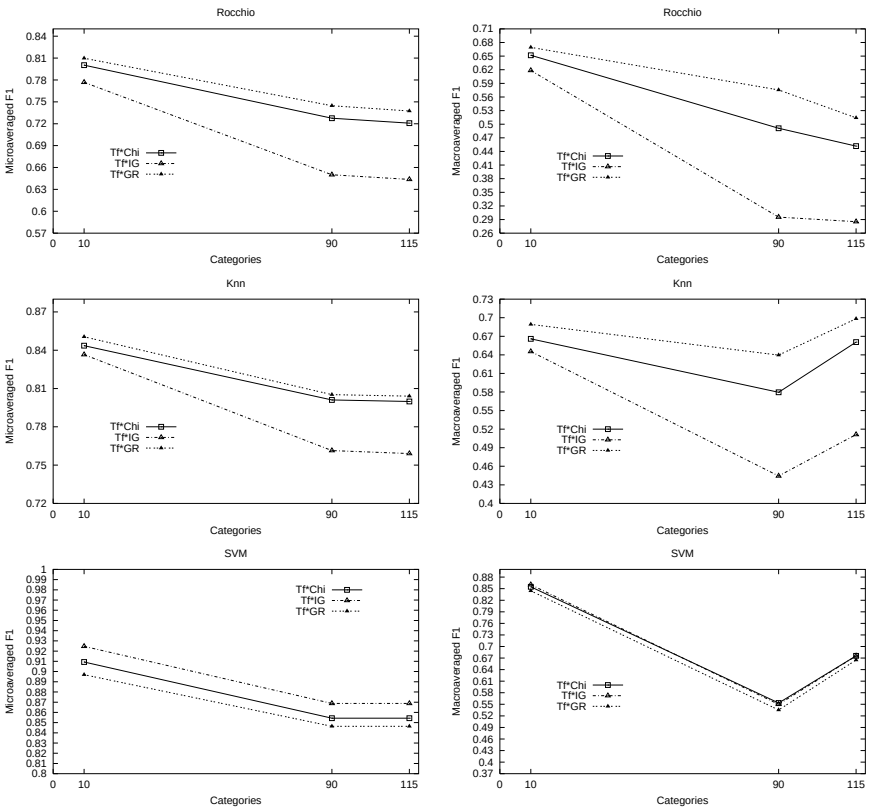

Figure 4: Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with supervised weighting and a $\xi = 0.90$ reduction factor**. Plots indicate results obtained with Rocchio (top), k-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578 described in Section 3.

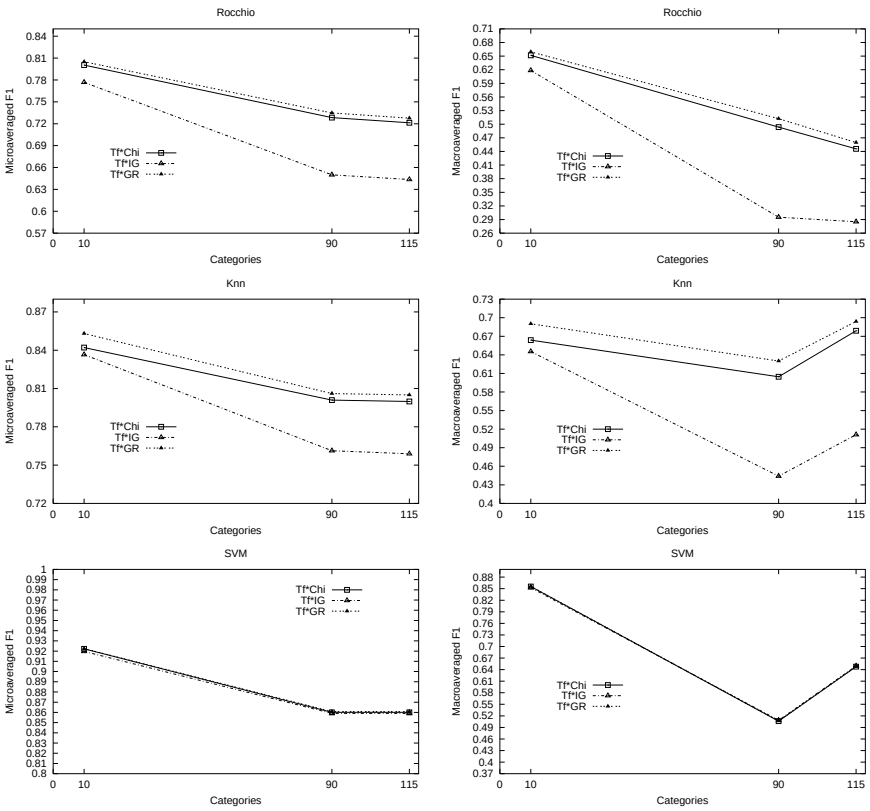

Figure 5: Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with supervised weighting and a $\xi = 0.50$ reduction factor**. Plots indicate results obtained with Rocchio (top), k-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578 described in Section 3.

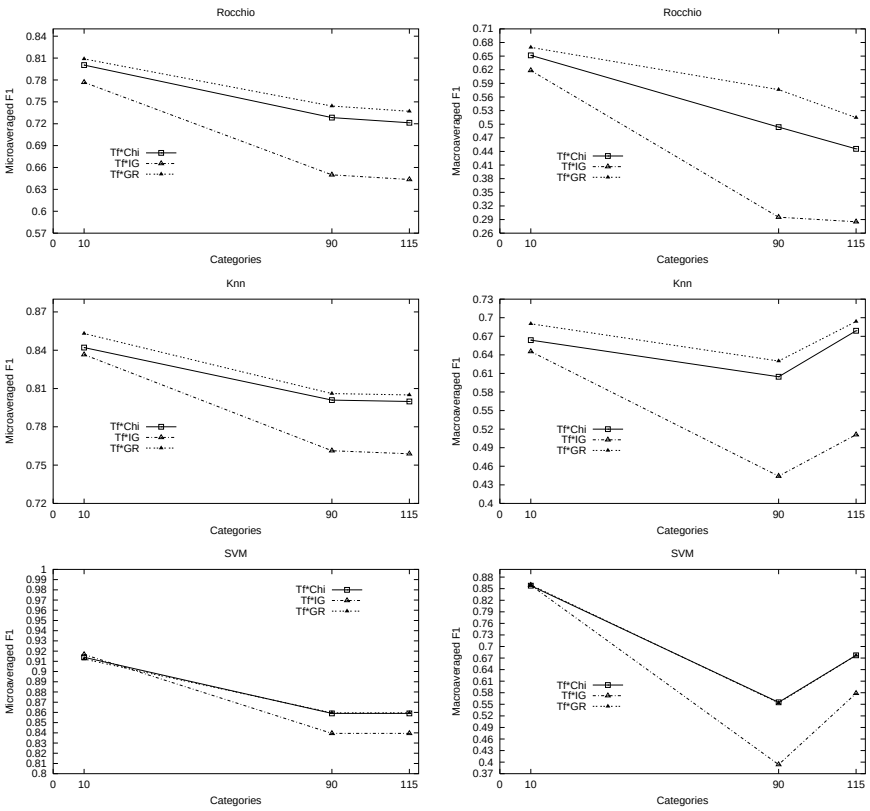

Figure 6: Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained **with supervised weighting and a $\xi = 0.0$ reduction factor**. Plots indicate results obtained with Rocchio (top), k-NN (middle) and SVMs (bottom). The X axis indicates the three subsets of Reuters-21578 described in Section 3.

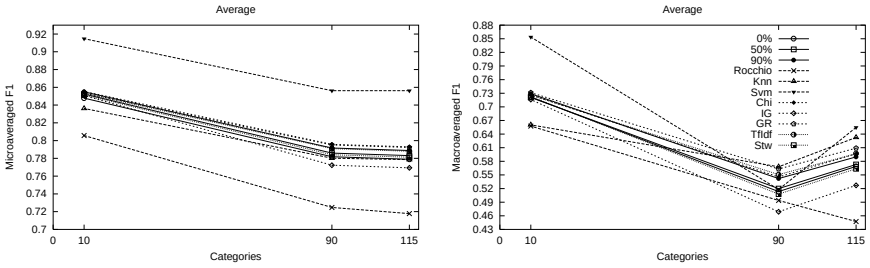

Figure 7: Plots of micro-averaged F_1 (leftmost) and macro-averaged F_1 (rightmost) obtained by averaging across term weighting policies, feature selection policies, feature selection functions, reduction factors for feature selection, and learning methods. The X axis indicates the three subsets of Reuters-21578 described in Section 3.