Supervised Term Weighting for Automated Text Categorization[⋆]

Franca Debole and Fabrizio Sebastiani

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
Via Giuseppe Moruzzi, 1
56124 Pisa, Italy
{debole,fabrizio}@iei.pi.cnr.it

Abstract. The construction of a text classifier usually involves (i) a phase of *term selection*, in which the most relevant terms for the classification task are identified, (ii) a phase of *term weighting*, in which document weights for the selected terms are computed, and (iii) a phase of *classifier learning*, in which a classifier is generated from the weighted representations of the training documents. This process involves an activity of *supervised learning*, in which information on the membership of training documents in categories is used.

Traditionally, supervised learning enters only phases (i) and (iii). In this paper we propose instead that learning from the training data should also affect phase (ii), i.e. that information on the membership of training documents to categories be used to determine term weights. We call this idea *supervised term weighting* (STW). As an example of STW, we propose a number of "supervised variants" of $tfidf$ weighting, obtained by replacing the *idf* function with the function that has been used in phase (i) for term selection. The use of STW allows the terms that are distributed most differently in the positive and negative examples of the categories of interest to be weighted highest.

We present experimental results obtained on the standard Reuters-21578 benchmark with three classifier learning methods (Rocchio, k-NN, and support vector machines), three term selection functions (information gain, chi-square, and gain ratio), and both local and global term selection and weighting.

1 Introduction

Text categorization (TC) is the activity of automatically building, by means of machine learning (ML) techniques, *automatic text classifiers*, i.e. programs capable of labelling natural language texts from a domain $\mathcal{D}$ with thematic categories from a predefined set $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ [14]. More formally, the task

[⋆] This paper is a substantially extended version of a paper by the same authors and the same title which appears in the Proceedings of SAC-03, 18th ACM Symposium on Applied Computing, Melbourne, US, 2003, pp. 784–788.

is to approximate the unknown *target function* $\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{0, 1\}$ (that describes how documents ought to be classified) by means of a function $\hat{\Phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{0, 1\}$ called the *classifier*, such that Φ and $\hat{\Phi}$ "coincide as much as possible".

The construction of an automatic text classifier relies on the existence of an *initial corpus* $\Omega = \{d_1, \ldots, d_{|\Omega|}\}$ of documents preclassified under $\mathcal{C}$. A general inductive process (called the *learner*) automatically builds a classifier for $\mathcal{C}$ by learning the characteristics of $\mathcal{C}$ from a *training set* $Tr = \{d_1, \ldots, d_{|Tr|}\}$ of documents. Once a classifier has been built, its effectiveness (i.e. its capability to take the right categorization decisions) may be tested by applying it to the *test set* $Te = \Omega - Tr$ and checking the degree of correspondence between the decisions of the classifier and those encoded in the corpus. This is called a *supervised* learning activity, since learning is "supervised" by the information on the membership of training documents in categories.

The construction of a text classifier may be seen as consisting of essentially two phases:

1. a phase of *document indexing*, i.e. the creation of internal representations for documents. This typically consists in
   (a) a phase of *term selection*, i.e. a form of *dimensionality reduction* consisting in the selection, from the set $\mathcal{T}$ (that contains of all the terms that occur in the documents of Tr), of the subset $\mathcal{T}' \subset \mathcal{T}$ of terms that, when used as dimensions for document representation, are expected to yield the best effectiveness, or the best compromise between effectiveness and efficiency; and
   (b) a phase of *term weighting*, in which, for every term t_k selected in phase (1a) and for every document d_j, a weight $0 \leq w_{kj} \leq 1$ is computed which represents, loosely speaking, how much term t_k contributes to the discriminative semantics of document d_j;
2. a phase of *classifier induction*, i.e. the creation of a classifier by learning from the internal representations of the training documents.

Traditionally, supervised learning affects only phases (1a) and (2), and does not affect phase (1b). In this paper we propose instead that information on the membership of training documents in categories is used also in phase (1b), so as to make the weight w_{kj} reflect the importance that term t_k has in deciding whether document d_j belongs or not to the categories of interest. We call this idea *supervised term weighting* (STW).

Concerning the actual computation of term weights, we propose that phase (1b) capitalizes on the results of phase (1a), since the selection of the best terms is usually accomplished by scoring each term t_k by means of a term selection function $f(t_k, c_i)$ that measures its capability to discriminate category c_i, and then selecting the terms that maximize $f(t_k, c_i)$. In our proposal the $f(t_k, c_i)$ scores are not discarded after term selection, but become an active ingredient of the term weight.

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 $f(t_k, c_i)$ through some "globalization" policy. In this paper we experiment with both policies, but always using the same policy for both term selection and term weighting. Note that a consequence of adopting the local policy and reusing the scores for term weighting is that weights, traditionally a function of a term t_k and a document d_j, now also depend on a category c_i; this means that, in principle, the representation of a document is no more a vector of $|\mathcal{T}'|$ terms, but a set of vectors of $\mathcal{T}'_i$ terms, with $i = 1, \ldots, |\mathcal{C}|$.

The paper is organized as follows. Section 2 sets the stage, by discussing the roles that term selection and term weighting play in current approaches to TC. In Section 3 we describe in detail the idea behind STW, and introduce some example weighting functions based on this idea. In Section 4 we describe the results of experimenting these functions on Reuters-21578, the standard benchmark of text categorization research. These results have been obtained with three classifier learning methods (Rocchio, k-NN, and support vector machines), three term selection functions (information gain, chi-square, and gain ratio), and both local and global term selection and weighting. Section 5 concludes.

2 Document indexing in TC: the received wisdom

2.1 Term weighting

In text categorization, text filtering, text routing, and other applications at the crossroads of IR and ML, term weighting is usually tackled by means of methods borrowed from IR, i.e. methods that are unaffected by the presence of a learning phase. Many weighting methods have been developed within IR, and their variety is astounding. However, as noted by Zobel and Moffat [19] (from which the passages below are quoted), there are three *monotonicity assumptions* that, in one form or another, appear in practically all weighting methods:

1. "rare terms are no less important than frequent terms". We call this the *IDF assumption*;
2. "multiple appearances of a term in a document are no less important than single appearances". We call this the *TF assumption*;
3. "for the same quantity of term matching, long documents are no more important than short documents". We call this the *normalization assumption*.

These assumptions are well exemplified by the $tfidf$ function (here presented in its standard "ltc" variant [13]), i.e.

$$tfidf(t_k, d_j) = tf(t_k, d_j) \cdot \log \frac{|Tr|}{\#_{Tr}(t_k)} \tag{1}$$

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. The $tf(t_k, d_j)$ component of Equation 1 enforces the TF assumption, while the $\log \frac{|Tr|}{\#_{Tr}(t_k)}$ component of the same equation enforces the IDF assumption. Weights obtained by Equation 1 are usually normalized by cosine normalization, i.e.

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

which enforces the normalization assumption.

2.2 Dimensionality reduction by term selection

Many classifier induction methods are computationally hard, and their computational cost is a function of the length of the vectors that represent the documents. It is thus of key importance to be able to work with vectors shorter than $|\mathcal{T}|$, which is usually a number in the tens of thousands or more. For this, *term selection* techniques are used to select from $\mathcal{T}$ a subset $\mathcal{T}'$ (with $|\mathcal{T}'| \ll |\mathcal{T}|$) of terms that are deemed most useful for compactly representing the meaning of the documents. The value

$$\xi = \frac{|\mathcal{T}| - |\mathcal{T}'|}{|\mathcal{T}|} \tag{3}$$

is called the *reduction factor*. Usually, these techniques consist in scoring each term in $\mathcal{T}$ by means of a *term evaluation function* f (TEF) and then selecting a set $\mathcal{T}'$ of terms that maximize f. Often, term selection is also beneficial in that it tends to reduce *overfitting*, i.e. the phenomenon by which a classifier tends to be better at classifying the data it has been trained on than at classifying other data.

Many functions, mostly from the tradition of information theory and statistics, have been used as TEFs in TC [4, 11, 16, 18]; those of interest to the present work are illustrated in Table 1. In the third column of this table, probabilities are interpreted on an event space of documents (e.g. $P(\bar{t}_k, c_i)$ indicates the probability that, for a random document x, term t_k does not occur in x and x belongs to category c_i), and are estimated by maximum likelihood.

Most of these functions try to capture the intuition according to which the most valuable terms for categorization under c_i are those that are distributed most differently in the sets of positive and negative examples of c_i. However, interpretations of this basic principle may vary subtly across different functions; see Section 4.1 for a discussion relative to the functions of Table 1.

In Table 1 every function $f(t_k, c_i)$ refers to a specific category c_i; in order to assess the value of a term t_k in a "global", category-independent sense, a

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 1. Term evaluation functions used in this work.

"globalization" technique is applied so as to extract a global score $f_{glob}(t_k)$ from the $f(t_k, c_i)$ scores relative to the individual categories. The most common globalization techniques are the sum $f_{sum}(t_k) = \sum_{i=1}^{|\mathcal{C}|} f(t_k, c_i)$, the weighted sum $f_{wsum}(t_k) = \sum_{i=1}^{|\mathcal{C}|} P(c_i) f(t_k, c_i)$, and the maximum $f_{max}(t_k) = \max_{i=1}^{|\mathcal{C}|} f(t_k, c_i)$ of their category-specific values $f(t_k, c_i)$.

3 Supervised term weighting

While the normalized *tfidf* function of Equation 2, or other similar term weighting functions from the IR literature, are routinely used in IR applications involving supervised learning such as text categorization or filtering, we think that their use in these contexts is far from being the optimal choice. In particular, the present paper challenges the IDF assumption. In standard IR contexts this assumption is reasonable, since it encodes the quite plausible intuition that a term t_k that occurs in too many documents is not a good discriminator, i.e. when it occurs in a query q it is not sufficiently helpful in discriminating the documents relevant to q from the irrelevant. However, if training data for the query were available (i.e. documents whose relevance or irrelevance to q is known), an even stronger intuition should be brought to bear, i.e. the one according to which the best discriminators are the terms that are distributed most differently in the sets of positive and negative training examples.

Training data is not available for queries in standard IR contexts, but is usually available for categories in TC contexts, where the notion of "relevance to a query" is replaced by the notion of "membership in a category". In these contexts, *category-based* functions (such as those listed in Table 1) that score terms according to how differently they are distributed in the sets of positive and negative training examples, are thus better substitutes of *idf*-like functions. The following example will help to clarify the point.

Example 1. Suppose $|\mathcal{C}| = 1$, i.e. we are just interested in deciding whether documents fall into category c or into its complement $\overline{c}$. Suppose there are 1000 training documents, 100 of which are positive examples of c and 900 of which are negative examples. Suppose term t_1 occurs in 90 out of the 100 positive examples and in none of the negative examples, term t_2 occurs in none of the positive examples and in 800 out of the 900 positive examples, and term t_3 occurs in 1 out of the 100 positive examples and 9 out of the 900 negative examples. An *idf*-like measure will weigh t_3 higher than both t_1 and t_2, since it occurs in less documents. A category-based function will instead weigh t_1 and t_2 higher than t_3, since they are distributed more differently across c and $\overline{c}$ than t_3, which is evenly distributed across them. The actual scores for this example are reported in Table 2.

	$\#(t,c)$	$\#(t,\overline{c})$	$idf(t)$	$IG(t,c)$	$\chi^2(t,c)$	$GR(t,c)$
t_1	090	000	3.474	0.267	890.110	0.822
t_2	000	800	0.322	0.232	444.444	0.714
t_3	001	009	6.644	0.000	000.000	0.000

Table 2. Values for *idf*, *IG*, χ^2 and *GR* for the terms of Example 1.

It might be argued that this idea is not novel, since this is what several probabilistic models do. For instance, the naive Bayesian classifier (see e.g. [9]) has the form

$$P(c_i|d_j) \propto \sum_{k=1}^{|\mathcal{T}|} w_{kj} \log \frac{P(t_k|c_i)(1 - P(t_k|\overline{c}_i))}{P(t_k|\overline{c}_i)(1 - P(t_k|c_i))} \tag{4}$$

The log factor in Equation 4 is computed from the training data, exactly as in our approach[1], and may be seen as a weight to be attributed to the terms which are present in the test document d_j (i.e. the terms for which $w_{kj} = 1$). However, the notion of STW we are proposing *does not coincide with the learning model* (as in the naive Bayesian model above), and may instead be used *together with any learning model* that admits non-binary representations as input, such as e.g. Rocchio, k-NN, SVMs, or neural networks.

One attractive aspect of using STW measures in TC is that, when such functions have been used for term selection, the scores they attribute to terms are already available. The approach we propose here puts thus the scores computed in the phase of term selection to maximum use: instead of discarding these scores after selecting the terms that will take part in the representations, these scores are used also in the term weighting phase.

[1] The log factor is itself a well-known TEF, known as *Odds Ratio* (see e.g. [11]).

4 Experiments

We have conducted a number of experiments to test the validity of the STW idea. The experiments have been run on a standard benchmark using three different TEFs, employed both according to the local and global policies, and always using the same TEF both as the term selection function *and* as a component of the term weighting function. Therefore, when we speak e.g. of using $IG(g)$ as a STW technique, we mean using IG (according to the global policy, denoted by "(g)" – local is denoted by "(l)") both as a term selection function *and* as a substitute of $\log \frac{|Tr|}{\#_{Tr}(t_k)}$ in Equation 1.

4.1 Term evaluation functions

In our experiments we have used the three TEFs illustrated in Table 1. The first two have been chosen since they are the two most frequently used category-based TEFs in the TC literature (document frequency is also often used as a TEF [18], but it is not category-based), while the third has been chosen since, as we discuss below, we consider it a theoretically better motivated variant of the second.

The first TEF we discuss is the chi-square (χ^2) statistics, which is frequently used in the experimental sciences in order to measure how the results of an observation differ (i.e. are independent) from the results expected according to an initial hypothesis (lower values indicate lower dependence)[2]. In term selection we measure how independent t_k and c_i are. The terms t_k with the lowest value for $\chi^2(t_k, c_i)$ are thus the most independent from c_i; since we are interested in the terms which are not, we select the terms t_k for which $\chi^2(t_k, c_i)$ is highest.

The second TEF we employ is *information gain* (IG), an information-theoretic function which measures the amount of information one random variable contains about another (or, in other words, the reduction in the uncertainty of a random variable that knowledge of the other brings about)[3]; it is 0 for two independent variables, and grows monotonically with their dependence [2]. In term selection we measure how much information term t_k contains about category c_i, and we are interested in selecting the terms that are more informative about (i.e. more indicative of the presence or of the absence of) the category, so we select the terms for which $IG(t_k, c_i)$ is highest.

The third TEF we discuss is *gain ratio* (GR), defined as the ratio between the information gain $IG(X, Y)$ of the two variables X and Y and the entropy of one of them ($H(X)$ or $H(Y)$) [12]. Although, to our knowledge, GR has never been used for feature selection purposes, we claim that for term selection it is a better alternative than IG since, as Manning and Schütze [10, p. 67] note,

[2] Since χ^2 is a statistics, it is usually best viewed in terms of actual counts from a contingency table, and not in terms of probabilities. In Table 3 we have formulated χ^2 in probabilistic terms for better comparability with the other functions listed.

[3] Information gain is also known as *mutual information* [10, pp. 66 and 583]. Although many TC researchers have used this function under one name or the other, the fact that the two names refer to the same object seems to have gone undetected.

IG grows not only with the degree of dependence of the two variables, but also with their entropy. Dividing $IG(t_k, c_i)$ by $H(c_i) = -\sum_{c \in \{c_i, \overline{c}_i\}} P(c) \log_2 P(c)$ allows us to compare the different values of term t_k for different categories on an equal basis. Note in fact that while $0 \leq IG(t_k, c_i) \leq min\{H(t_k), H(c_i)\}$, we have instead that $0 \leq GR(t_k, c_i) \leq 1$. Comparing the different scores that t_k has obtained on the different categories is especially important when applying the globalization techniques described in Section 2.2. For instance, it is clear that if we choose IG as our TEF and $f_{max}(t_k) = \max_{i=1}^{|\mathcal{C}|} f(t_k, c_i)$ as our globalization function, the score $IG(t_k, c_1)$ for a category c_1 with high entropy has a higher probability of being selected that the score $IG(t_k, c_2)$ for a category c_2 with low entropy. Instead, with GR these categories do not enjoy this "unfair advantage".

4.2 Learning methods

Since a document d_j can belong to zero, one or many of the categories in $\mathcal{C}$, we tackle the classification problem as $|\mathcal{C}|$ independent problems of deciding whether d_j belongs or not to c_i, for $i = 1, \ldots, |\mathcal{C}|$.

In our experiments we have used three different learning methods, which we have chosen with the aim of assembling a fairly representative sample of methods that allow weighted (non-binary) input. The first is a standard Rocchio method [5] for learning linear classifiers. A classifier for category c_i consists of a vector of weights

$$w_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{w_{kj}}{|NEG_i|} \tag{5}$$

where w_{kj} is the weight of t_k in document d_j, $POS_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = 1\}$ and $NEG_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = 0\}$. 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 on a validation set by the proportional thresholding method [7].

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

$$score(d_j, c_i) = \sum_{d_z \in \ Tr_k(d_j)} (\boldsymbol{d}_j \cdot \boldsymbol{d}_z) \Phi(d_z, c_i) \tag{6}$$

where $Tr_k(d_j)$ is the set of the k documents d_z which maximize the dot product $\boldsymbol{d}_j \cdot \boldsymbol{d}_z$. Classification is performed by thresholding on the scores resulting from Equation 6; here too we have individually optimized each threshold on a validation set by proportional thresholding. The k parameter has been set to 30, following the results in [4].

The third learning method is a support vector machine (SVM) learner as implemented in the SVMLIGHT package (version 3.5) [6]. SVMs attempt to learn a hyperplane in $|\mathcal{T}|$-dimensional space that separates the positive training examples from the negative ones with the maximum possible margin, i.e. such

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

Table 3. Effectiveness measures used in all the experiments reported in this paper; TP, TN, FP and FN refer to the sets of true positives, true negatives, false positives, and false negatives, respectively.

that the minimal distance between the hyperplane 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.

4.3 Experimental setting

In our experiments we have used the "Reuters-21578, Distribution 1.0" corpus, currently the most widely used benchmark in text categorization research[4]. Reuters-21578 consists of a set of 12,902 news stories, partitioned (according to the "ModApté" split we have adopted) into a training set of 9,603 documents and a test set of 3,299 documents. The documents are labelled by 118 categories; the average number of categories per document is 1.08, ranging from a minimum of 0 to a maximum of 16. The number of positive examples per category ranges from a minimum of 1 to a maximum of 3964.

All our results are reported (a) for the set of 115 categories with at least one training example (hereafter, Reuters-21578(115)), (b) for the set of 90 categories with at least one training example and one test example (Reuters-21578(90)), and (c) for the set of the 10 categories with the highest number of training examples (Reuters-21578(10)). Sets (a) and (b) are obviously the hardest, since they include categories with very few positive instances for which inducing reliable classifiers is obviously a haphazard task. Reporting the results for the three different sets has the double aim of

- allowing a finer-grained analysis of the performance of our techniques;
- assessing the relative "hardness" of the three subsets of Reuters-21578 which have been most frequently used in the TC literature, thus allowing an "indirect" comparison among previously published techniques that have been tested on different subsets.

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

In all the experiments discussed in this section, stop words have been removed using the stop list provided in [7, 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. We have measured effectiveness in terms of precision wrt c_i (π_i) and recall wrt c_i (ρ_i), defined in the usual way. Values relative to individual categories are averaged to obtain values of precision (π) and recall (ρ) global to the entire category set according to the two alternative methods of microaveraging and macroaveraging, defined in Table 3. Neither microaveraging nor macroaveraging is the "absolute" evaluation measure, and which one should be adopted obviously depends on the application requirements. In general, the ability of a classifier to behave well also on categories with few positive training instances is emphasized by macroaveraging and much less so by microaveraging.

As a measure of effectiveness that combines the contributions of π and ρ we have used the well-known F_β function [8], defined as

$$F_\beta = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho}$$

with $0 \leq \beta \leq +\infty$. Similarly to most other researchers we have set $\beta = 1$, which places equal emphasis on π and ρ.

The results of our experiments are reported in Figures 1 and 2.

In all the experiments reported, term selection was performed with a reduction factor $\xi = .90$. Although we have also thoroughly tested values of $\xi = .50$ and $\xi = 0$ (i.e. no term selection), we omit to include them (i) for reasons of space, and (ii) because the $\xi = .90$ experiments are the ones that have yielded the best effectiveness for most STW functions and for $tfidf$ too, and are then the most significant. This is in accordance with the findings of Yang and Pedersen [18], who found that the effectiveness of most term selection functions peaks close to the $\xi = .90$ value. The SVMs experiments we include are an exception, since they are the ones with $\xi = 0$. The reason for reporting them instead of the $\xi = .90$ experiments is that they were generally the best performing ones; again, this is in accordance with results of Brank et al. [1], who found that for each of the numerous term selection techniques they tested SVMs perform best with $\xi = 0$ ([16, 17] also reached similar conclusions).

Whenever term selection has been performed according to the global policy, the $f_{max}(t_k) = \max_{i=1}^{|\mathcal{C}|} f(t_k, c_i)$ has been used as the globalization technique, since in preliminary experiments it consistently outperformed the other globalization techniques described in Section 2.2. The reason why $f_{max}(t_k)$ performs well is that it prefers terms that are very good separators even on a single category, rather than terms that are only "fair" separators on many categories. In fact, if t_k is a very good separator for c_i, then $f(t_k, c_i)$ is going to be very high, so that there are good chances that $f_{max}(t_k) = f(t_k, c_i)$, which means that there are good chances that t_k is selected, which means in turn that there is a good separator for c_i in the selected term set.

In all experiments, STW techniques have been compared with a baseline formed by cosine-normalized $tfidf$ weighting (in the "ltc" variant of Equations 1 and 2) preceded by term selection performed with the TEF that, in combination with *tfidf* weighting, has yielded the best performance (namely, IG for Rocchio, χ^2 for k-NN, and no term selection for SVMs). Note that although stronger weighting functions than "ltc" *tfidf* have been reported in the literature [19], all of them are based on the three monotonicity assumptions mentioned in Section 2.1; this means that our STW techniques could be applied to them too, probably yielding similar performance differentials.

4.4 Analysis of the results: STW functions

The thorough experiments we have performed have not shown a uniform superiority of STW with respect to standard term weighting: in some cases *tfidf* has outperformed all STW techniques, while in other cases some of the STW techniques have improved on *tfidf*. Let us try to analyze the results more in detail; for ease of discussion we will refer to the results obtained on Reuters-21578(90).

Rocchio, as a learning method, and macroaveraging, as an evaluation measure, are the contexts in which the different techniques exhibit the biggest difference in performance wrt each other; for the other two learning methods, and for microaveraging in general, differences are less remarkable, although statistically significant. Different weighting techniques are the best performers for different learning methods: *tfidf* for Rocchio (with $GR(g)$ almost as good), $GR(g)$ for k-NN, and both $GR(g)$ and $\chi^2(g)$ on SVMs (although on SVMs *tfidf* is just as good on microaveraging). The fact that both $\chi^2(g)$ and $GR(g)$ have achieved an 11% improvement (.582 vs. .524) on macroaveraged effectiveness over the best *tfidf* for SVMs, while basically maintaining the same microaveraged effectiveness, is of particular relevance, since SVMs are currently the best performing TC method in the literature. Analogously, the 9% improvement obtained on k-NN by $GR(g)$ and $\chi^2(g)$ wrt *tfidf* is also noteworthy, since k-NN is also known as a very good performer [17].

Among the various STW techniques, $GR(g)$ is a uniformly high scoring one, and often the best of the lot. From Table 4, in which we report the average results of our 6 STW functions across the 3 different learning methods we have used, we may see that $GR(g)$ is the best performer for both micro- and macroaveraging and for all three Reuters-21578 subsets examined. Chi-square is also a good performer across the board. $IG(g)$ is instead a disappointing performer, sometimes disastrously so (namely, in all macroaveraged experiments). Among the local policies, $GR(l)$ is again generally the best, with $IG(l)$ usually faring better than $\chi^2(l)$.

We are not surprised by the good performance of $GR(g)$ since, as we have remarked in Section 4.1, we consider $GR(g)$ a theoretically superior alternative to $IG(g)$. The disappointing performance that this latter has produced is a striking contrast with the well-known good performance of IG as a term selection function [18]. Note that $IG(l)$ and $GR(l)$ perform identically. This is due to the fact that the two differ only by the entropy of c_i being used as a normalization factor in $GR(l)$. Therefore, it is quite obvious that, locally to category c_i, $IG(l)$

		$\chi^2(g)$	$IG(g)$	$GR(g)$	$\chi^2(l)$	$IG(l)$	$GR(l)$
Micro F_1	Reuters-21578(10)	0.852	0.843	**0.857**	0.810	0.816	0.816
	Reuters-21578(90)	0.795	0.750	**0.803**	0.758	0.767	0.767
	Reuters-21578(115)	0.793	0.747	**0.800**	0.756	0.765	0.765
Macro F_1	Reuters-21578(10)	0.725	0.707	**0.739**	0.674	0.684	0.684
	Reuters-21578(90)	0.542	0.377	**0.589**	0.527	0.559	0.559
	Reuters-21578(115)	0.596	0.458	**0.629**	0.581	0.608	0.608

Table 4. Average micro- and macro-averaged F_1 on the three major subsets of Reuters-21578 described in Section 4.3 for the six STW functions discussed in this paper.

and $GR(l)$ select the same terms and give them weights that differ only by a constant multiplicative factor.

A surprising result is that global STW techniques are almost everywhere superior to the corresponding local technique. We say this is surprising because the global policy openly contradicts the decision to view the classification problem as $|\mathcal{C}|$ independent binary classification problems. That is, if these $|\mathcal{C}|$ problems are really to be seen as independent, then the problem of building representations for them should also be viewed on a category-by-category basis, which is what the local policy does. We conjecture that this surprising behaviour is due to the fact that the statistics that can be collected from scarcely populated categories are not robust enough for the local policy to be effective, and that for these categories the global policy makes up for their unreliable statistics by providing more robust statistics collected over the entire category set.

4.5 Analysis of the results: different Reuters-21578 subsets

As a by-product of this investigation, in Table 5 we list the average micro- and macro-averaged effectiveness resulting from *all* our experiments on the three subsets of Reuters-21578 mentioned in Section 4.3. Each average has been computed across the three STW functions, each one in its local and global version, and the three learning methods; each value is thus the average of 18 different values. Although the absolute performance levels are not necessarily significant, their difference is, since this is somehow indicative of the relative "hardness" of these subsets, and allows us to compare previously published techniques that have been tested on different subsets. Note that there is no published result, to our knowledge, that compares these three subsets experimentally in a systematic way[5]. The comparison we carry out is of some significance since, among other things, it is performed across widely different learning methods and widely used term selection functions.

[5] An experimental comparison of subsets (a) and (c) is reported in [3]. However, note that subset (b) is by far the most frequently used in the TC literature.

	Micro π	**Micro ρ**	**Micro F_1**	**Macro π**	**Macro ρ**	**Macro F_1**
Reuters-21578(10)	0.808	0.863	0.832	0.685	0.726	0.704
Reuters-21578(90)	0.754	0.805	0.773	0.669	0.481	0.522
Reuters-21578(115)	0.749	0.805	0.770	0.654	0.593	0.576

Table 5. Average micro- and macro-averaged F_1 on the three major subsets of Reuters-21578 described in Section 4.3.

The fact that Reuters-21578(10) turns out to be the easiest subset is quite obvious, given that its categories are the ones with the highest number of positive examples. The average decrease in performance in going from Reuters-21578(10) to Reuters-21578(90) is much higher on macroaveraging than on microaveraging; this is no surprise, since adding scarcely populated categories does not penalize microaveraging much (since for microaveraging categories count proportionally to the number of their test examples), while it does for macroaveraging (since for microaveraging categories count all the same).

It is instead somehow surprising that Reuters-21578(90) is no easier than Reuters-21578(115), since the 25 additional categories have on average much fewer training examples than the other 90. A possible explanation is that many of the classifiers learnt for these categories are frequent rejectors (namely, classifiers with very high thresholds), and that, since these categories have no positive test examples, this often results in both $\pi_i = 1$ and $\rho_i = 1$. Of course, this boosts macroaveraging, so this might also explain the apparently surprising increase of macroaveraged performance in going from Reuters-21578(90) to Reuters-21578(115).

5 Conclusion

We have proposed *supervised term weighting* (STW), a term weighting methodology specifically designed for IR applications involving supervised learning, such as text categorization and text filtering. Supervised term indexing leverages on the training data by weighting a term according to how different its distribution is in the positive and negative training examples. We have also proposed that this should take the form of replacing *idf* by the category-based term evaluation function that has previously been used in the term selection phase; as such, STW is also efficient, since it reuses for weighting purposes the scores already computed for term selection purposes.

We have tested STW in all the combinations involving three different learning methods and three different term weighting functions, each tested in its local and global version. One of these functions (gain ratio) was not known from the TC term selection literature, and was proposed here since we think it is a theoretically superior alternative to the widely used information gain (aka mutual information) function. The results have confirmed the overall superiority of gain ratio over information gain and chi-square when used as a STW function.

Although not proving consistently superior to $tfidf$, STW has given several interesting results. In particular, a STW technique based on gain ratio has given very good results across the board, showing an improvement of 11% over $tfidf$ in macroaveraging for SVMs, currently the best performing TC method in the literature, and an improvement of 9% over $tfidf$ in macroaveraging for k-NN, another very good performer.

As a by-product of this investigation, we have reported a study on the relative "hardness" of the three major subsets of Reuters-21578, which will allow researchers to compare previously published techniques that have been tested on different subsets.

6 Acknowledgements

We thank Luigi Galavotti for making available his RealCat software [4], with which most of these experiments were performed. A similar thank goes to Thorsten Joachims for making available the SVMlight package [6]. Thanks also to Henri Avancini, Pio Nardiello, Guido Ricci, and Alessandro Sperduti for many fruitful discussions.

References

1. J. Brank, M. Grobelnik, N. Milić-Frayling, and D. Mladenić. Interaction of feature selection methods and linear classification models. In *Proceedings of the ICML-02 Workshop on Text Learning*, Sydney, AU, 2002.
2. T. M. Cover and J. A. Thomas. *Elements of information theory.* John Wiley & Sons, New York, US, 1991.
3. S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In G. Gardarin, J. C. French, N. Pissinou, K. Makki, and L. Bouganim, editors, *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US, 1998. ACM Press, New York, US.
4. L. Galavotti, F. Sebastiani, and M. Simi. Experiments on the use of feature selection and negative evidence in automated text categorization. In J. L. Borbinha and T. 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.
5. D. A. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In W. B. Croft and C. 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.
6. T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. J. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 11, pages 169–184. The MIT Press, Cambridge, US, 1999.
7. D. D. Lewis. *Representation and learning in information retrieval.* PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, US, 1992.

8. D. D. Lewis. Evaluating and optmizing autonomous text classification systems. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, US, 1995. ACM Press, New York, US.
9. D. D. Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 4–15, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE. Published in the "Lecture Notes in Computer Science" series, number 1398.
10. C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, US, 1999.
11. D. Mladenić and M. Grobelnik. Feature selection for unbalanced class distribution and naive Bayes. In I. Bratko and S. Dzeroski, editors, *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 258–267, Bled, SL, 1999. Morgan Kaufmann Publishers, San Francisco, US.
12. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
13. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988. Also reprinted in [15], pp. 323–328.
14. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
15. K. Sparck Jones and P. Willett, editors. *Readings in information retrieval*. Morgan Kaufmann, San Mateo, US, 1997.
16. H. Taira and M. Haruno. Feature selection in SVM text categorization. In *Proceedings of AAAI-99, 16th Conference of the American Association for Artificial Intelligence*, pages 480–486, Orlando, US, 1999. AAAI Press, Menlo Park, US.
17. Y. Yang and X. Liu. A re-examination of text categorization methods. In M. A. Hearst, F. Gey, and R. 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.
18. Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.
19. J. Zobel and A. Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998.

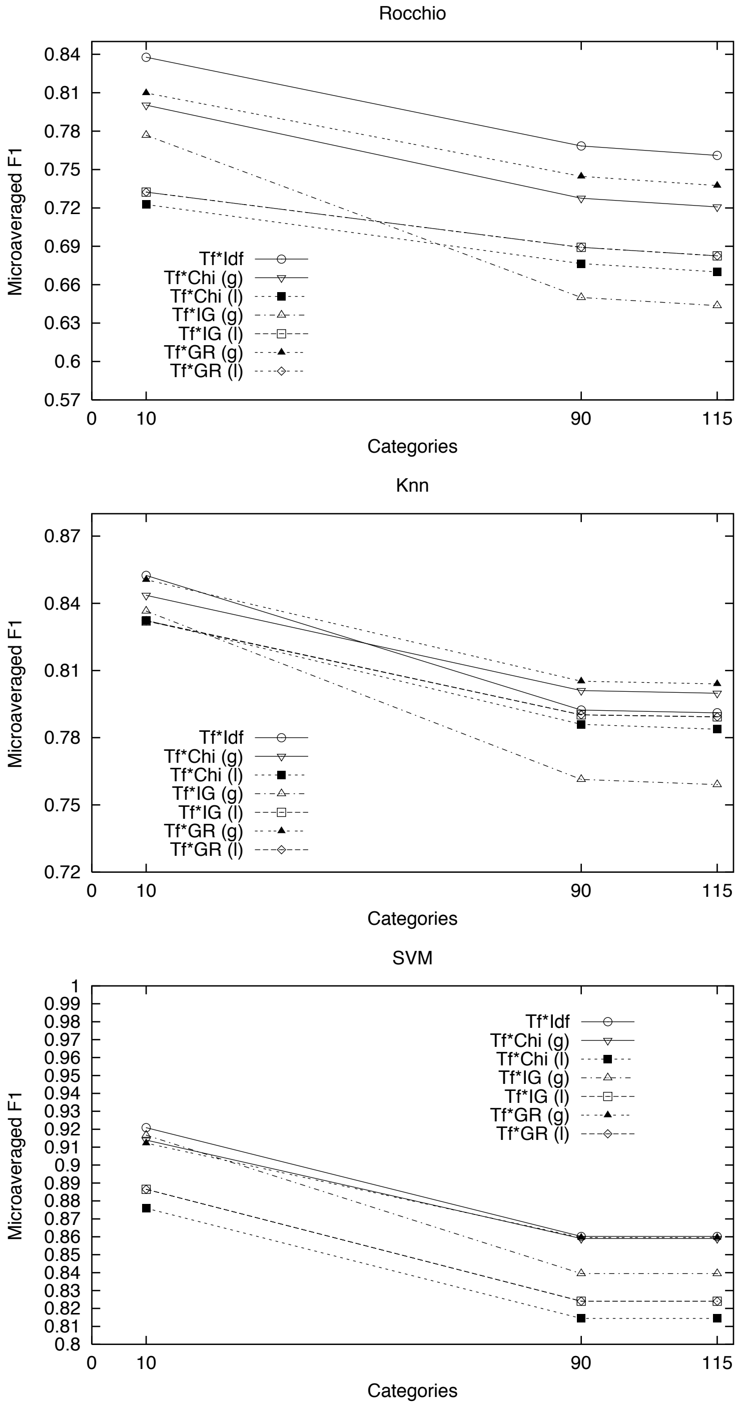

Fig. 1. Plots of micro-averaged F_1 for Rocchio (top), k-NN (middle) and SVMs (bottom). The X axis indicates the three major subsets of Reuters-21578 described in Section 4.3, while each curve represents a different term weighting function. For instance, the notation Tf*Chi(g) indicates the use of χ^2 (with the global policy, indicated by the notation "(g)") both for term selection *and* as a substitute of *idf* in $tfidf$. The notation Tf*Idf always refers to $tfidf$ weighting and term selection obtained with the (global) method that, in connection with $tfidf$ weighting, has performed best (IG for Rocchio, χ^2 for k-NN, no selection for SVMs). Note the different scales used for the Y axis.

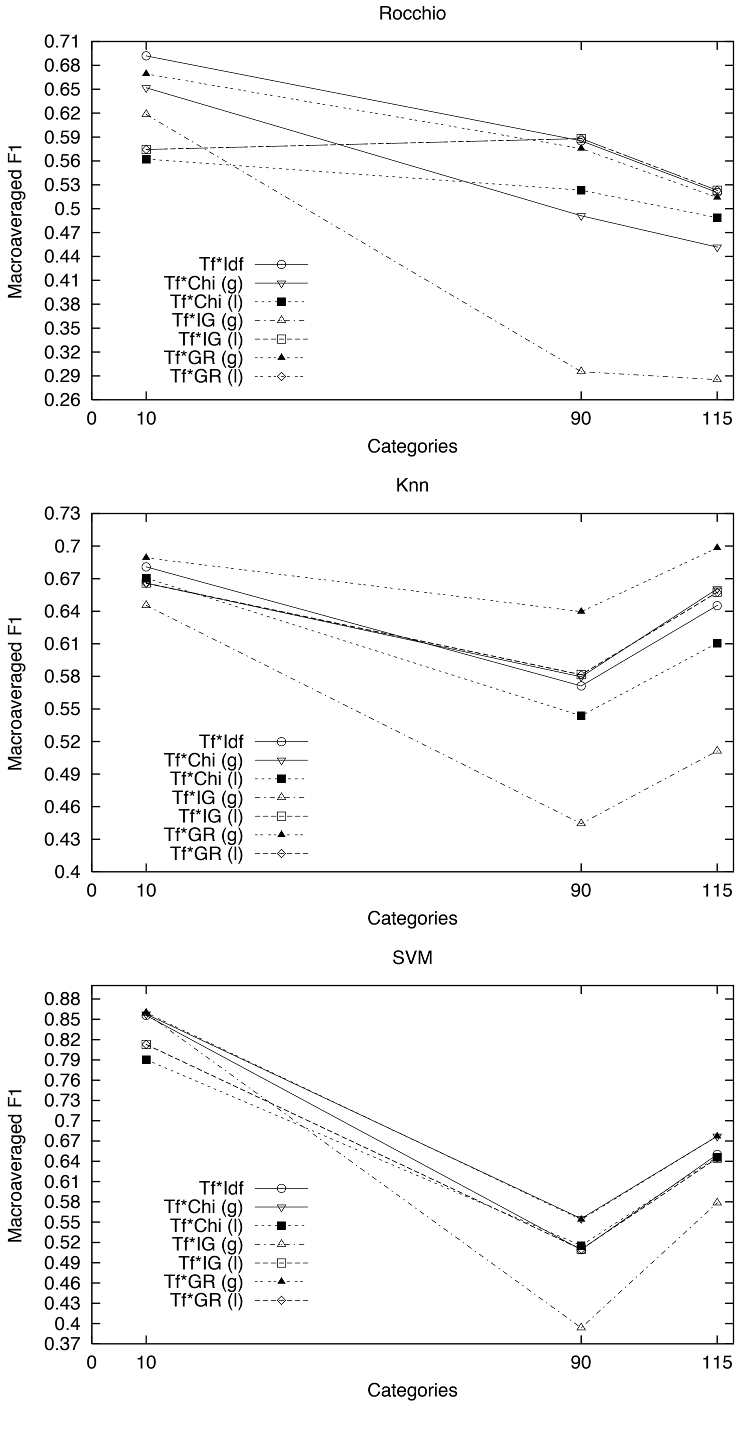

Fig. 2. Plots of macro-averaged F_1 for Rocchio (top), k-NN (middle) and SVMs (bottom). Notational conventions are as for Figure 1.