Knowl Inf Syst (2016) 49:909–931
DOI 10.1007/s10115-016-0924-1

REGULAR PAPER

Combining supervised term-weighting metrics for SVM text classification with extended term representation

Mounia Haddoud[1,2] · **Aïcha Mokhtari**[1] · **Thierry Lecroq**[2] · **Saïd Abdeddaïm**[2]

Received: 18 March 2015 / Revised: 25 November 2015 / Accepted: 3 February 2016 /
Published online: 19 February 2016

Abstract The accuracy of a text classification method based on a SVM learner depends on the weighting metric used in order to assign a weight to a term. Weighting metrics can be classified as supervised or unsupervised according to whether they use prior information on the number of documents belonging to each category. A supervised metric should be highly informative about the relation of a document term to a category, and discriminative in separating the positive documents from the negative documents for this category. In this paper, we propose 80 metrics never used for the term-weighting problem and compare them to 16 functions of the literature. A large number of these metrics were initially proposed for other data mining problems: feature selection, classification rules and term collocations. While many previous works have shown the merits of using a particular metric, our experience suggests that the results obtained by such metrics can be highly dependent on the label distribution on the corpus and on the performance measures used (microaveraged or macroaveraged F_1-Score). The solution that we propose consists in combining the metrics in order to improve the classification. More precisely, we show that using a SVM classifier which combines the outputs of SVM classifiers that utilize different metrics performs well in all situations. The second main contribution of this paper is an extended term representation for the vector space model that improves significantly the prediction of the text classifier.

Keywords Text classification · Term weighting · Text representation · Support vector machines · Classifier combination

✉ Mounia Haddoud
mounia.haddoud@etu.univ-rouen.fr

Saïd Abdeddaïm
said.abdeddaim@univ-rouen.fr

1 RIIMA, USTHB, BP 32, El-Alia, Bab-Ezzouar, 16111 Algiers, Algeria

2 LITIS, Université de Rouen, 76821 Mont-Saint-Aignan Cedex, France

1 Introduction

Text classification is the problem of automatically labeling natural language texts with predefined thematic categories. In the last two decades, a huge number of machine learning techniques were proposed to automatically classify and organize text documents [1,28]. These studies were motivated by the exponential growing of the number of texts available online. Applications includes classification of news articles, web pages and scientific publications into controlled vocabulary, sentiment analysis and opinion mining among social networks, spam filtering, protein classification.

In text classifier systems, documents are preprocessed in order to be suitable as training data for a learning algorithm. Traditionally, each text document is converted into a vector where each dimension represents a term which value is the weight that will be used in the learning process. As the weight reflects the importance of the term in the document, an appropriate choice of the metric function used for weighting terms is crucial for correct classification.

Traditional unsupervised term weights metrics, as the popular TFIDF, depend only on term frequency in the document and the (inverse) number of training documents containing this term. For the purpose of text classification, supervised alternatives have been developed to take into account the categories of the corpus documents [3,5,7–10,14,20,21,23,24,27]. The key idea is to build a metric function that discriminates the terms according to the category for which we are testing the document membership. Such a metric should be highly informative about the relation of a document term t to a category c and discriminative in separating the positive documents from the negative documents of category c. While unsupervised term weights depend only on term frequency in the document and the number of training corpus documents containing this term, supervised term weights are computed for each category and use the number of documents belonging to the category containing this term. Intuitively, supervised term weights measure the degree of correlation between term's presence in a document and membership of this document in the category.

In this work, we propose to experiment for term weighting the large number of metric functions which were proposed for other data mining problems in order to measure the correlation between two events. We use metrics collected from papers dealing with feature selection [13,26,31], supervised term weighting [3,8,10,14,21,23,27], classification rules [15] and term collocations [25]. We compare experimentally 96 metrics for term weighting. Only 16 of them have been used for term-weighting problem in the literature, and 9 are new metrics designed by the authors of this paper. It appears that using these metrics instead of already used weights can improve the performances of SVM classifiers. Moreover, we show that combining metrics improves the quality of the classification. While many previous works have shown the merits of using a particular metric [3,5,7–10,14,20,21,23,24,27], our experience suggests that the results obtained by such metrics can be highly dependent on the label distribution on the corpus and also on the performance measures used (microaveraged or macroaveraged F_1-Score). However, we show that using a SVM classifier which combines the outputs of SVM classifiers that use different metrics improves significantly text classification performances in all situations.

The second main contribution of this paper is an extended term representation for the vector space model. Following the scheme used by TFIDF metric, which is the product of the term frequency (TF) and inverse document frequency (IDF), alternative supervised forms have been traditionally formulated by replacing the IDF term with a supervised metric function [3]. However, merging by product TF with a factor such as IDF, Chi square or odds ratio is

problematic. It is not clear why a two times larger TF is equivalent to a two times larger IDF (Chi square or odds ratio). From our point of view, TFIDF is the product of two quantities that are not of the same statistical nature. TF counts the number of occurrences of a term t in a document, while IDF counts the (inverse) number of documents containing the term t. We propose as an alternative to count the (inverse) number of documents containing the term with frequency at least n. By doing this, we integrate the term frequency in a document in the number of documents. The IDF quantity becomes the number of documents containing the term t at least n times. Precisely, in our model, we propose to convert a text document into a vector where each dimension represents a feature of the form (t, n), meaning that the term t appears in the document at least n times. If the term t appears 10 times in the document, we generate all the term frequency features (t, n) with $n = 1, 2, 4$ and 8 (powers of 2 in order to limit the number of features). Hence, rather than associating the weight TFIDF to a term t, we affect in our model the weight IDF to features (t, n) which depends on the inverse number of documents containing a term t at least n times. Our intuition is that this new definition of IDF keeps more information for learning. This assumption is confirmed experimentally as it improves the quality of the text classification. As all term weighting metrics (such as Chi square or odds ratio) depend on the document frequency, our extended term representation, explained here for IDF, is applicable to any metric. We also propose another type of extended term feature based on the idea that the terms which are more correlated with the subject of the document tend to appear at the beginning. We generate term position features (t, p), meaning that the first position of t in the document is lower or equal to p. Experimental results show that using these two extended term representations improves significantly the prediction of the text classifier.

A brief review about supervised term-weighting metrics for text classification is presented in Sect. 2. Our extended term representation is presented in Sect. 3. The metrics that we proposed to compare are described in Sect. 4. The experimental comparison on Reuters-21578, Ohsumed and 20 Newsgroups datasets are presented in Sect. 5.

2 Related works

First term-weighting metrics for text classification were unsupervised and generally borrowed from information retrieval (IR) field. The simplest IR metric is the binary representation BIN which assigns a weight of 1 if the term appears in the document and 0 otherwise. The term can be assigned a weight TF that depends on its frequency tf in the document. Different variants of term frequency have been presented, for example, the raw term frequency tf or its logarithm $\log(1+tf)$. TFIDF is the most commonly used weighting metric in text classification. TFIDF is the product of TF and IDF, the inverse document frequency which favors rare terms in the corpus over frequent ones. However, there are some drawbacks on using unsupervised weighting functions, as the category information is omitted.

Previous studies proposed different supervised weighting metrics where the document frequency factor IDF of TFIDF is replaced by a factor that use prior information on the number of documents belonging to each category. Several classical metrics were tested in the literature, for instance, Chi square (χ^2), information gain (IG), gain ratio (GR) and odds ratio (OR) [5,7,8,10]. These early studies get an improvement with TF.χ^2, TF.IG, TF.GR and TF.OR term weights trained with SVM.

Accurate SVM text classification was obtained using Bi-Normal Separation (BNS) metric [13] for supervised term weighting [14]. In the later study, Forman tested two variants TF.BNS

and BIN.BNS (considering the term frequency or not) and noticed that "best F-measure was obtained by using binary features with BNS scaling" (BIN.BNS) but "recall was slightly better with TF.BNS features". This observation shows that merging TF with a factor such as BNS is problematic, i.e., not using TF yields to a better F_1-Score but decreases the fraction of the predicted categories that are relevant for a document.

More recently, other specific metrics were proposed for the supervised term-weighting problem. Liu et al. [21] use a probability-based (PB) term weight in order to tackle the problem of imbalanced distribution of documents among categories. Lan et al. [20] utilize a term weight TF.RF based of the relevance frequency (RF) metric. The relationship and differences between these term-weighting metrics are studied in [2]. Martineau et al. [23] propose a metric TF.δIDF where IDF is replaced by the class inverse document frequency difference (δIDF). Altinçay and Erenel [3] combine RF metric with mutual information and the difference of term occurrence probabilities in the collection of the documents belonging to the category and in its complementary set. Nguyen et al. [24] propose a weighting scheme based on the Kullback–Leibler (KL) and Jensen–Shannon (JS) divergence measures for centroid-based classifiers. Ren and Sohrab [27] test two metrics TF.IDF.ICF and TF.IDF.ICS$_\delta$F that incorporate the inverse class frequency (ICF) and inverse class space density frequency (ICS$_\delta$F) to TF.IDF. Bouillot et al. [6] propose alternative metrics for centroid term weighting and investigated the influence of numbers of categories, documents and terms in the classification of small datasets. Deng et al. [9] and Fattah [12] adapt and compare various text classification weighting metrics for sentiment analysis. This application is also considered in the two pre-cited papers [23] and [24]. Badawi and Altinçay [4] propose a framework based on employing the co-occurrence statistics of pairs of terms for term selection and weighting in binary text classification. Escalante et al. [11] use genetic programming for weighting terms. Ko [19] use a weighting scheme based on the term relevance ratio (TRR).

From this state-of-the-art, we notice that each paper in the literature gives a new metric and demonstrates its classification improvement on some corpora considering a certain number of categories (typically 10 categories for Reuters corpus). However, as we will show in the following, there is no metric among the literature and also among the 80 metrics we propose that yields the best results in all situations (corpus and number of categories). In order to overcome this problem, we propose to combine the metrics.

3 Extended term representation

Text classification is traditionally achieved by applying a learning method to a representation of the text document. In the vector space model, the document is represented as a vector in the term space. Each dimension of the vector space represents a term which value is the weight that will be used in the learning process. In this section, we propose to represent each dimension by a term together with its minimal frequency or its minimal first position in the document. We call these alternatives extended term representations.

3.1 Term features

In this classical representation, terms are viewed as the dimensions of the learning space. A term may be a single word or a phrase (n-gram).

3.2 Term frequency features

The number of occurrences of a term t in a document d is by itself a property that we propose to use as a feature. Let us consider, for example, a particular term t such that 25 % of the documents where t appears are in category c. If 45 % of the documents where t appears at least 3 times are in category c, then the term t is probably more correlated with the category c when its frequency exceeds 2. Hence, we propose features of the form (t, n) in documents containing t with a term frequency at least n. If a document d contains ten times a term t, we must generate ten features (t, i) $(i = 1, 2, \ldots, 10)$, meaning that t occurs at least once, twice, …, ten times. This could unnecessarily grow the number of features so we consider only powers of 2 less or equal to n. Then, if t occurs ten times, we will generate the features $(t, 1)$, $(t, 2)$, $(t, 4)$ and $(t, 8)$. The number of frequency features associated with a term t which appears n times in a document d will only be $\log_2 n$ in the worst case. In practice, however, most terms have a low frequency and the number of features grows moderately as we will show in the experiments (see Sect. 5.4).

3.3 Term position features

Most of the terms that are related to the main topics of a document occur at its beginning. In order to validate this assumption, we propose features of the form (t, p), meaning that the first position of t in the document is lower or equal to p. The position being defined as the number of words preceding the term occurrence. As for term frequency features, we generate only features (t, p) when p is a power of 2. For example, if a term t first appears at position 5 in a document of size 100 words, we generate the features $(t, 8)$, $(t, 16)$, $(t, 32)$ et $(t, 64)$, meaning that the first position of t is lower or equal than 8, 16, 32 and 64. The number of position features associated with a term t which appears in a document d at first position p will be $\log_2 |d|$ in the worst case, where $|d|$ is the size of d in number of words. However, using term frequency features augments moderately the number of features (see Sect. 5.4).

4 Weighting metrics

Supervised term metrics try to give a high weight to a feature that is particularly present in documents that belong to a category. Hence, a *good* term-weighting metric must be a measure of an observed correlation between two events in the set of training documents: containing a term and belonging to a category. In this section, we propose to use the large number of metric functions proposed for other data mining problems, but not yet used for term weighting, in order to measure the correlation between the two events.

4.1 Notations

We consider a corpus D of N documents and d a particular document of D.

Let x denotes a nominal feature of d representing either:

- t a term that occurs in d,
- (t, n) a term that occurs at least n times in d,
- or (t, p) a term which first position is lower or equal to p in the document d.

Each document can belong to one or many categories (labels or classes) $c_1, c_2, \ldots, c_M$. We denote by y a particular category c_i.

Table 1 Two-way contingency table for nominal feature x and category y

	y	$\bar{y}$	$*$
x	$f(xy) = f_{11} = a$	$f(x\bar{y}) = f_{12} = b$	$f(x*) = f_1$
$\bar{x}$	$f(\bar{x}y) = f_{21} = c$	$f(\bar{x}\bar{y}) = f_{22} = d$	$f(\bar{x}*)$
$*$	$f(*y) = f_2$	$f(*\bar{y})$	$f(**) = N$

Table 2 Expected contingency table for nominal feature x and category y

	y	$\bar{y}$	$*$
x	$\hat{f}(xy) = \frac{f(x*)f(*y)}{N}$	$\hat{f}(x\bar{y}) = \frac{f(x*)(N-f(*y))}{N}$	$\hat{f}(x*)$
$\bar{x}$	$\hat{f}(\bar{x}y) = \frac{(N-f(x*))f(*y)}{N}$	$\hat{f}(\bar{x}\bar{y}) = \frac{(N-f(x*))(N-f(*y))}{N}$	$\hat{f}(\bar{x}*)$
$*$	$\hat{f}(*y)$	$\hat{f}(*\bar{y})$	N

We denote by $\bar{x}$ the fact that the feature x is not present in d and by $\bar{y}$ the fact that d does not belong to the category y.

The number of documents containing the feature x and belonging to the category y is denoted by $f(xy)$ and represents the document frequency. In general, $f(uv)$ denotes the number of documents containing u and belonging to v, u being x, $\bar{x}$ or $*$ (documents containing any term) and v being y, $\bar{y}$ or $*$ (documents belonging to any category). These frequencies are represented in the contingency table (Table 1) in which the number of documents is denoted by N, $f(xy)$ by a and f_{11}, $f(x\bar{y})$ by b and f_{12}, and so on.

Many metrics are based on the estimation of the probability $P(uv)$, the probability that a document containing u belongs to the category v, u being x, $\bar{x}$ or $*$ and v being y, $\bar{y}$ or $*$. Under the maximum-likelihood hypothesis, this probability is estimated by:

$$p(uv) = \frac{f(uv)}{N}$$

Some metrics are based on the difference between the observed and the expected frequencies. The expected contingency frequencies under the null hypothesis of independence H_0 are given in Table 2.

Few metrics use the number of categories containing a document that contains a feature x. This quantity is denoted by $f_c(x)$ and corresponds to:

$$f_c(x) = |\{y | f(xy) > 0\}|$$

4.2 Metrics

Giving a weight to a feature x associated with a term in a document labeled with y depends on the correlation between x and y in the training corpus. This correlation can be estimated by different metrics, and all the metrics used in this paper depend only on five values:

- N the number of training documents.
- $f(xy)$ the joint frequency.
- $f(x*)$ *and* $f(*y)$ the marginal frequencies.
- $f_c(x)$ the number of categories containing (a document that contains) feature x.
- M the number of categories.

Given these values, one can compute the contingency table and then compute any of the 96 metrics described in Table 3. Most of these metrics are collected from papers dealing with feature selection [13,26,31], supervised term weighting [3,8,10,14,21,23,27], classification rules [15] and term collocations [25]. The first 16 metrics of Table 3 are those already used for the term- weighting problem in the literature [3,8,10,14,20,21,23,27]. The last 9 metrics are proposed by the authors of this paper.

5 Experiments

5.1 Benchmark

In order to compare experimentally the metrics, we use Reuters-21578, Ohsumed and 20 Newsgroups corpora. These datasets are the most widely used benchmarks for text classification.

The distribution of the categories in Reuters-21578 corpus is highly unbalanced. In order to study the performances obtained by each weighting metric in more or less unbalanced situations, our results on Reuters-21578 are reported:

- for the 115 categories with at least one training example,
- for the 52 categories with at least 16 training examples,
- and for the set of the 10 categories with the highest number of training examples.

Ohsumed is a medical abstract corpus with 23 cardiovascular diseases categories. Twenty Newsgroups corpus contains articles taken from 20 Usenet newsgroups (categories).

Term variation can affect its frequency which is an important parameter in the term weight, and the solution consists in replacing each word by its stem. For all the corpora, we used Porter stemmer which gives the best performances in our experiments. After stemming, we have tokenized the text documents. For each sentence in a document, we generate all possible n-grams (terms). We choose the size of n-grams according to the performances obtained in each corpus. For Reuters-21578 corpus, the size of n was fixed to 1; for Ohsumed and 20 Newsgroups corpora, we fixed $n \leq 2$.

We used the training/testing split proposed in Reuters-21578 (Mobapte split) and Ohsumed corpora. There is no fixed literature split for 20 Newsgroups. It is usually used for cross-validation. We have adopted a fivefold cross-validation on 20 Newsgroups corpus in order to evaluate the statistical significance of the achieved performance improvements.

Traditionally, the performance of a classifier on a corpus is estimated by learning the classification on the training data and evaluating the accuracy of the prediction obtained on the evaluation data. The evaluation metrics used are the *precision* which is the proportion of documents placed in the category that are really in the category, the *recall* which is the proportion of documents in the category that are actually placed in the category, and the F_1-Score defined as:

$$F_1\text{-Score} = \frac{2.\text{precision}.\text{recall}}{\text{precision} + \text{recall}}$$

The microaveraged F_1-Score is computed globally for all the categories, while the macroaveraged F_1-Score is the average of the F_1-Scores computed for each category. The latter measures the ability of a classifier to perform well when the distribution of the categories is unbalanced, while the microaveraged F_1-Score gives a global view of the document classification performance.

Table 3 Metrics used for supervised feature weighting

Metric	Mathematical form
IDF	$\log(\frac{N}{f(x*)})$
Pearson's χ^2 test[a]	$\sum_{i,j} \frac{(f_{ij}-\hat{f_{ij}})}{\hat{f_{ij}}}$
Information gain[a]	$\sum_{u\in\{x,\bar{x}\}} \sum_{v\in\{y,\bar{y}\}} p(uv) \log \frac{p(uv)}{p(u*)p(*v)}$
Gain ratio[a]	$\frac{\sum_{u\in\{x,\bar{x}\}} \sum_{v\in\{y,\bar{y}\}} p(uv) \log \frac{p(uv)}{p(u*)p(*v)}}{\sum_{v\in\{y,\bar{y}\}} p(v) \log p(v)}$
Odds ratio[b]	$\frac{ad}{bc}$
Log odds ratio[c]	$\log \frac{ad}{bc}$
Forman log odds ratio[c]	$\lvert \log \frac{ad}{bc} \rvert$
Bi-normal separation (BNS)[c]	$\lvert F^{-1}(p(x\lvert y)) - F^{-1}(p(x\lvert \bar{y}))\rvert$[d]
Probability-based term weight[e]	$\log(1+\frac{a^2}{bc})$
Pointwise mutual information[e]	$\log \frac{p(xy)}{p(x*)p(*y)}$
Relevance frequency[f]	$\log_2(2+\frac{a}{\max(b,1)})$
Relevance frequency$_{OR}$[g]	$\log_2(2+\frac{a}{\max(b,1)})(1+p(x\lvert y)-p(x\lvert \bar{y}))$
Relevance frequency$_{\chi^2}$[g]	$\log_2(2+\frac{a}{\max(b,1)})\lvert p(x\lvert \bar{y})-p(x\lvert y)\rvert$
δIDF[h]	$\log(\frac{p(x\lvert y)}{p(x\lvert \bar{y})})$
IDF.ICF[i]	$\log(1+\frac{N}{f(x*)})\log(1+\frac{M}{f_c(x)})$
IDF.ICS$_\delta$F[i]	$\log(1+\frac{N}{f(x*)})\log(1+\frac{M}{\sum_{c_i\in C} p(x\lvert c_i)})$
Joint probability	$p(xy)$
Conditional probability	$p(y\lvert x)=\frac{p(xy)}{p(x*)}$
Reverse conditional probability	$p(x\lvert y)$
Mutual dependency	$\log \frac{p(xy)^2}{p(x*)p(*y)}$
Log frequency biased	$\log \frac{p(xy)^2}{p(x*)p(*y)} + \log p(xy)$
Normalized expectation	$\frac{2f(xy)}{f(x*)f(*y)}$
Mutual expectation	$\frac{2f(xy)}{f(x*)f(*y)} + p(xy)$
Salience	$\log \frac{p(xy)^2}{p(x*)p(*y)} + \log f(xy)$
t Test	$\frac{f(xy)-\hat{f}(xy)}{\sqrt{f(xy)(1-(f(xy)/N))}}$
z Score	$\frac{f(xy)-\hat{f}(xy)}{\sqrt{\hat{f}(xy)(1-(\hat{f}(xy)/N))}}$
Poisson significance	$\frac{\hat{f}(xy)-f(xy)\log\hat{f}(xy)+\log f(xy)!}{\log N}$
Log likelihood ratio	$-2\sum_{i,j} f_{i,j} \log \frac{f_{i,j}}{\hat{f}_{i,j}}$
Squared log likelihood ratio	$-2\sum_{i,j} f_{i,j} \frac{\log f_{i,j}^2}{\hat{f}_{i,j}}$
Russel–Rao	$\frac{a}{a+b+c+d}$

Table 3 continued

Metric	Mathematical form
Sokal–Michiner	$\frac{a+d}{a+b+c+d}$
Rogers–Tanimoto	$\frac{a+d}{a+2b+2c+d}$
Hamann	$\frac{(a+d)-(b+c)}{a+b+c+d}$
Third Sokal–Sneath	$\frac{b+c}{a+d}$
Jaccard	$\frac{a}{a+b+c}$
First Kulczynski	$\frac{a}{b+c}$
Second Sokal–Sneath	$\frac{a}{a+2(b+c)}$
Second Kulczynski	$\frac{1}{2}(\frac{a}{a+b}\frac{a}{a+c})$
Yulle's ω	$\frac{\sqrt{ad}-\sqrt{bc}}{\sqrt{ad}+\sqrt{bc}}$
Yulle's Q	$\frac{ad-bc}{ad+bc}$
Driver–Kroeber	$\frac{a}{\sqrt{(a+b)(a+c)}}$
Fifth Sokal–Sneath	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$
Pearson	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$
Baroni-Urbani	$\frac{a+\sqrt{ad}}{a+b+c+\sqrt{ad}}$
Braun-Blanquet	$\frac{a}{\max(a+b,a+c)}$
Simpson	$\frac{a}{\min(a+b,a+c)}$
Michael	$\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$
Mountford	$\frac{2a}{2bc+ab+ac}$
Fager	$\frac{a}{\sqrt{(a+b)(a+c)}}$
Unigram subtuples	$\log\frac{ad}{bc} - 3.29\sqrt{\frac{1}{a}\frac{1}{b}\frac{1}{c}\frac{1}{d}}$
U cost	$\log(1+\frac{\min(b,c)+a}{\max(b,c)+a})$
S cost	$\log(1+\frac{\min(b,c)+a}{a+1})^{-\frac{1}{2}}$
R cost	$\log(1+\frac{a}{a+b})\log(1+\frac{a}{a+c})$
T combined cost	$\sqrt{U\text{cost}\times S\text{cost}\times R\text{cost}}$
Phi	$\frac{p(xy)-p(x*)p(*y)}{\sqrt{p(x*)p(*y)(1-p(x*))(1-p(*y))}}$
Kappa	$\frac{p(xy)+p(\bar{x}\bar{y})-p(x*)p(*y)-p(\bar{x}*)p(*\bar{y})}{1-p(x*)p(*y)-p(\bar{x}*)p(*\bar{y})}$
J measure	$\max[p(xy)\log\frac{p(y\|x)}{p(*y)} + p(x\bar{y})\log\frac{p(\bar{y}\|x)}{p(*\bar{y})}, p(xy)\log\frac{p(x\|y)}{p(x*)} + p(\bar{x}y)\log\frac{p(\bar{x}\|y)}{p(\bar{x}*)}]$
One-way J measure	$p(xy)\log(\frac{p(y\|x)}{p(*y)}) + p(x\bar{y})\log(\frac{p(\bar{y}\|x)}{p(*\bar{y})})$
Gini index	$\max[p(x*)(p(y\|x))^2+p(\bar{y}\|x)^2)-p(*y)^2+p(\bar{x}*)(p(y\|\bar{x}))^2+p(\bar{y}\|\bar{x})^2) - p(*\bar{y})^2, p(*)(p(x\|y))^2 + p(\bar{x}\|y)^2) - p(x*)^2p(*)(p(x\|y))^2+p(\bar{x}\|y)^2)-p(x*)^2+p(*\bar{y})(p(x\|\bar{y}))^2+p(\bar{x}\|\bar{y})^2) - p(\bar{x}*)^2]$
One-way Gini index	$p(x*)(p(y\|x))^2 + p(\bar{y}\|x)^2) - p(*y)^2 + p(\bar{x}*)(p(y\|\bar{x}))^2 + p(\bar{y}\|\bar{x})^2) - p(*\bar{y})^2$

Table 3 continued

Metric	Mathematical form
Confidence	$\max[p(y\|x), p(x\|y)]$
Laplace	$\max[\frac{f(xy)+1}{f(x*)+2}, \frac{f(xy)+1}{f(*y)+2}]$
One-way Laplace	$\frac{f(xy)+1}{f(x*)+2}$
Conviction	$\max[\frac{p(x*)p(*\bar{y})}{p(x\bar{y})}, \frac{p(\bar{x}*)p(*y)}{p(\bar{x}y)}]$
One-way conviction	$\frac{p(x*)p(*\bar{y})}{p(x\bar{y})}$
Piatersky–Shapiro	$p(xy) - p(x*)p(*y)$
Certainty factor	$\max[\frac{p(y\|x)-p(*y)}{1-p(*y)}, \frac{p(x\|y)-p(x*)}{1-p(x*)}]$
One-way certainty factor	$\frac{p(y\|x)-p(*y)}{1-p(*y)}$
Added value	$\max[p(y\|x) - p(*y), p(x\|y) - p(x*)]$
One-way added value	$p(y\|x) - p(*y)$
Collective strength	$\frac{p(xy)p(\bar{x}\bar{y})}{p(x*)p(*y)+p(\bar{x}*)p(*\bar{y})} \cdot \frac{1-p(x*)p(*y)-p(\bar{x}*)p(*\bar{y})}{1-p(xy)-p(\bar{x}\bar{y})}$
Klosgen	$\sqrt{p(xy)}\max(p(y\|x) - p(*y), p(x\|y) - p(x*))$
One-way Klosgen	$\sqrt{p(xy)}(p(y\|x) - p(*y))$
GSS coefficient	$p(xy)p(\bar{x}\bar{y}) - p(x\bar{y})p(\bar{x}y)$
Specificity	$p(\bar{y}\|\bar{x})$
Leverage	$p(y\|x) - p(x*)p(*y)$
Relative risk	$p(y\|x)/p(y\|\bar{x})$
One-way support	$p(y\|x)\log_2 \frac{p(xy)}{p(x*)p(*y)}$
Two-way support	$p(xy)\log_2 \frac{p(xy)}{p(x*)p(*y)}$
Two-way support variation	$p(xy)\log_2 \frac{p(xy)}{p(x*)p(*y)} + p(x\bar{y})\log_2 \frac{p(x\bar{y})}{p(x*)p(*\bar{y})} + p(\bar{x}y)\log_2 \frac{p(\bar{x}y)}{p(\bar{x}*)p(*y)+p(\bar{y}\|\bar{x})}\log_2 \frac{p(\bar{x}\bar{y})}{p(\bar{x}*)p(*\bar{y})}$
Loevinger	$1 - \frac{p(x*)p(*\bar{y})}{p(x\bar{y})}$
Sebag–Schoenauer	$\log \frac{p(xy)}{p(x*)p(*y)}$
Least contradiction	$\frac{p(xy)-p(x\bar{y})}{p(*y)}$
Odd multiplier	$\frac{p(xy)p(x\bar{y})}{p(*y)p(x\bar{y})}$
Example and counterexample rate	$1 - \frac{p(x\bar{y})}{p(xy)}$
Zhang	$\frac{p(xy)-p(x*)p(*y)}{\max(p(xy)p(*\bar{y}), p(*y)p(x\bar{y}))}$
Weighted log likelihood ratio	$p(x\|y)\log(\frac{p(x\|y)}{p(x\|\bar{y})})$
Document TFIDF	$\frac{f(xy)}{f(*y)}\log \frac{N}{f(x*)}$
Reverse-way document TFIDF	$\frac{f(xy)}{f(x*)}\log \frac{N}{f(*y)}$
Conditional probability variation 1	$p(y\|x)p(\bar{y}\|\bar{x})$
Conditional probability variation 2	$p(y\|x) + p(\bar{y}\|\bar{x})$
Conditional probability variation 3	$p(y\|x) + p(*y)$
Conditional probability variation 4	$\frac{p(y\|x)+p(\bar{y}\|\bar{x})+p(\bar{x}\|\bar{y})}{3}$

Table 3 continued

Metric	Mathematical form
Conditional probability variation 5	$p(x\|y) - p(x\|\bar{y})$
Relevance frequency variation	$\log_2(2 + \frac{a}{\max(b,1)}) \times \log_2(2 + \frac{a}{\max(c,1)})$
Reverse-way BNS	$\|F^{-1}(p(y\|x)) - F^{-1}(p(y\|\bar{x}))\|$[i]

[a] Debole and Sebastiani [8]
[b] Deng et al. [10]
[c] Forman [14]
[d] F^{-1} is the inverse normal cumulative distribution function.
[e] Liu et al. [21]
[f] Lan et al. [20]
[g] Altinçay and Erenel [3]
[h] Martineau et al. [23]
[i] Ren and Sohrab [27]

5.2 Learning

In multi-label text classification, each document d belongs to one or many of the categories in $C = \{c_1, c_2, \ldots, c_l\}$. In order to learn for multi-label classification, we use the traditional binary relevance (BR) strategy [22,29,32], the well-known one-against-all problem transformation method that learns $|C|$ independent binary classifiers, one for each category. Each binary classifier gives a probability that d belongs or not to the category $y = c_i$.

5.2.1 Weighting

For each binary classifier associated with a category y, every document d is transformed to a vector $W_d = (w(x_1, y, d), w(x_2, y, d), \ldots, w(x_n, y, d))$ where each feature x is weighted by:

$$w(x, y, d) = w_{\mathrm{TF}}(x, d) \times w_{\mathrm{DF}}(x, y)$$

The term frequency weight w_{TF} depends on the frequency of x in the document d. The document frequency weight w_{DF} is one of the metrics described in Table 3.

Each feature x can be either:

- a term feature t in the classical model,
- or a term frequency feature (t, n) or a term position (t, p) feature as defined in Sect. 3.

For the classical term representation, following [20], we experimented three possible term frequency weights (see Table 4). For our model, we use only binary term weights ($w_{\mathrm{TF}}(x, d) = \mathrm{BIN}(x, d)$), because the frequency of the term is already considered in the extended term representation $x = (t, n)$.

5.2.2 SVM classifier

For each category, we have used a SVM binary classifier which learns a linear combination of the features in order to define the decision hyperplane. We adopted the SVMLight tool [17] with a linear kernel and used the default settings. Previous studies show that SVMLight performs well for text classification [16].

Table 4 Experimented term frequency weights as a function of the frequency $tf(x, d)$ of x in d

Term frequency weight	Value	Description
BIN(x,d)	1 if $tf(x, d) > 0$, 0 otherwise	Binary weight
RTF(x,d)	$tf(x, d)$	Raw term frequency
LTF(x,d)	$\log(1 + tf(x, d))$	Term frequency logarithm
ITF(x,d)	$1 - \frac{1}{1-tf(x,d)}$	Inverse term frequency

5.2.3 SVM classifier combination

Classifier combination [30] methods are ensemble techniques that use the predictions of several classifiers to obtain better predictive performance than could be obtained from any of the constituent classifiers. One way to combine several classifiers is to consider the multiple classifier outputs as inputs to a generic classifier called secondary classifier.

In our case for each category, we combine with a SVM classifier the scores given by multiple base SVM binary classifiers, each classifier uses one of the 96 metrics for weighting the features. Base SVM learners are trained with the same set of documents. The classification scores obtained by each training document are used as input features for the secondary SVM learner. We also tried random forest as secondary classifier, but SVM gives better results.

5.3 Results

In order to estimate the performance of both our model and the 96 metrics, we have compared the F_1-Score of SVM classification on Reuters-21578, Ohsumed and 20 Newsgroups documents with classical and extended term representations using different weighting schemes.

5.3.1 Metric comparison

We recall that for a fixed category y the weight $w(x, y, d)$ of a feature x in a document d is:

$$w(x, y, d) = w_{\text{TF}}(x, d) \times w_{\text{DF}}(x, y)$$

where the term frequency weight w_{TF} (see Table 4) depends on the frequency of x in the document d and the document frequency weight w_{DF} is one of the metrics described in Table 3. The feature x represents either a term feature t, a term frequency feature (t, n) or a term position feature (t, p).

For each document frequency weight metric w_{DF} we have experimented 5 weighting schemes:

- raw term frequency weight ($w_{\text{TF}} = \text{RTF}$) for term features t
- term frequency logarithm weight ($w_{\text{TF}} = \text{LTF}$) for term features t
- inverse term frequency weight ($w_{\text{TF}} = \text{ITF}$) for term features t
- binary term frequency weight ($w_{\text{TF}} = \text{BIN}$) for term frequency features (t, n)
- binary term frequency weight ($w_{\text{TF}} = \text{BIN}$) for term frequency features (t, n) and term position features (t, p)

Table 5 reports the microaveraged F_1-Score for Reuters-21578 (10, 52 and 115 categories), Ohsumed and 20 Newsgroups. After calculation of the F_1-Score for each classifier, the metrics are ranked in descending order of the best weighting scheme score. Table shows only the

Table 5 Microaveraged F_1-Scores of SVM method with different term representations and weighting metrics (top-10 metrics and IDF scores) on Reuters (10, 52 and 115 categories), Ohsumed and 20 Newsgroups corpora

Corpus	Term representation	t	t	t	(t, n)	(t, n) and (t, p)
	Term frequency weight	RTF	LTF	ITF	BIN	BIN
Reuters 10 cat.	Document frequency weight					
1	Yulle's ω	0.912	0.921	0.926	0.927	**0.947**
2	Log odds ratio	0.913	0.922	0.927	0.929	**0.947**
3	Forman log odds ratio	0.913	0.922	0.927	0.929	**0.947**
4	Yulle's Q	0.909	0.919	0.924	0.925	**0.946**
5	Pointwise mutual information	0.909	0.920	0.923	0.929	**0.946**
6	Unigram subtuples	0.911	0.922	0.926	0.928	**0.946**
7	Bi-normal separation	0.915	0.922	0.927	0.928	**0.946**
8	δIDF	0.911	0.920	0.926	0.929	**0.945**
9	Zhang	0.908	0.918	0.921	0.923	**0.945**
10	Reverse-way BNS	0.911	0.920	0.924	0.924	**0.944**
...						
30	IDF	0.854	0.886	***0.892***	0.888	**0.932**
Reuters 52 cat.	Document frequency weight					
1	Bi-normal separation	0.843	0.861	0.867	0.878	**0.893**
2	Log odds ratio	0.837	0.855	0.860	0.872	**0.891**
3	Forman log odds ratio	0.837	0.855	0.860	0.872	0.**891**
4	Reverse-way BNS	0.834	0.853	0.855	0.870	**0.890**
5	Probability-based term weight	0.821	0.847	0.853	0.862	**0.889**
6	δIDF	0.833	0.853	0.858	0.869	**0.889**
7	Pointwise mutual information	0.826	0.853	0.856	0.864	**0.889**
8	Yulle's ω	0.830	0.854	0.859	0.868	**0.888**
9	Collective strength	0.820	0.842	0.849	0.858	**0.887**
10	Relevance frequency variation	0.826	0.848	0.854	0.866	**0.886**
...						
76	IDF	0.750	0.797	***0.810***	0.820	**0.864**
Reuters 115 cat.	Document frequency weight					
1	Bi-normal separation	0.852	0.865	0.870	**0.883**	0.877
2	Log odds ratio	0.848	0.865	0.870	0.**880**	0.875
3	Forman log odds ratio	0.848	0.865	0.870	**0.880**	0.875
4	Relevance frequency$_{OR}$	0.834	0.851	0.854	**0.878**	0.869
5	Conviction	0.835	0.847	0.852	**0.876**	0.861
6	Yulle's ω	0.843	0.861	0.865	**0.875**	0.870
7	Relevance frequency	0.828	0.849	0.850	**0.875**	0.865
8	Reverse-way BNS	0.847	0.861	0.865	0.874	**0.875**
9	S cost	0.833	0.850	0.853	**0.874**	0.864
10	One-way conviction	0.831	0.844	0.849	**0.874**	0.854
...						
75	IDF	0.790	0.823	***0.833***	**0.849**	0.845

Table 5 continued

Corpus	Term representation	t	t	t	(t, n)	(t, n) and (t, p)
	Term frequency weight	RTF	LTF	ITF	BIN	BIN
Ohsumed	Document frequency weight					
1	One-way Klosgen	0.587	0.604	0.609	0.631	**0.639**
2	Klosgen	0.586	0.600	0.605	0.626	**0.636**
3	z Score	0.578	0.601	0.605	0.624	**0.634**
4	Pearson	0.577	0.600	0.604	0.624	**0.633**
5	Phi	0.577	0.600	0.604	0.624	**0.633**
6	Squared log likelihood ratio	0.558	0.585	0.593	0.621	**0.631**
7	Odds ratio	0.563	0.582	0.590	0.617	**0.629**
8	One-way Gini index	0.593	0.598	0.600	0.618	**0.629**
9	Pearson's χ^2 test	0.593	0.598	0.600	0.618	**0.629**
10	Sebag–Schoenauer	0.560	0.581	0.588	0.616	**0.628**
...						
81	IDF	0.296	0.363	***0.380***	0.417	**0.444**
20 Newsgroups	Document frequency weight					
1	One-way Klosgen	0.731	0.759	0.764	0.767	**0.790**
2	z Score	0.713	0.755	0.762	0.767	**0.790**
3	Pearson	0.708	0.753	0.761	0.766	**0.788**
4	Phi	0.708	0.753	0.761	0.766	**0.788**
5	Bi-normal separation	0.664	0.737	0.747	0.749	**0.785**
6	Reverse-way BNS	0.666	0.737	0.747	0.750	**0.783**
7	One-way Laplace	0.671	0.733	0.742	0.748	**0.782**
8	Klosgen	0.706	0.742	0.750	0.758	**0.781**
9	Relative risk	0.692	0.743	0.751	0.756	**0.781**
10	Second Kulczynski	0.672	0.736	0.746	0.753	**0.781**
...						
78	IDF	0.396	0.589	***0.617***	0.628	**0.708**

top-10 metrics. It is clearly observed that the proposed representation models (t, n) and (t, n) and (t, p) perform significantly better than the classical representation and achieve the best performances in all experiments in terms of microaveraged F_1-scores for all the metrics. The model (t, n) and (t, p) performs better than the model (t, n), which means that using the position improves the performances. The only exception is Reuters-21578 with 115 categories. We think this is due to the fact that a significant number of categories (40/115) are represented by only up to 3 training documents, and the influence of position in the document cannot be learned correctly.

These observations comfort our intuition that including term frequency in document frequency formula as a feature is more relevant than multiplying those quantities. For the classical term representation model, the inverse term frequency (ITF) weight gives better F_1-scores than raw and logarithm term frequency (RTF and LTF). We also notice that the baseline metric TFIDF with the classical representation, precisely ITF.IDF, performs significantly worse than other metrics. Put together using other metrics than TFIDF and using

Table 6 Macroaveraged F_1-Scores of SVM method with different term representations and weighting metrics (top-10 metrics and IDF scores) on Reuters (10, 52 and 115 categories), Ohsumed and 20 Newsgroups corpora

Corpus	Term representation	t	t	t	(t, n)	(t, n) and (t, p)
	Term frequency weight	RTF	LTF	ITF	BIN	BIN
Reuters 10 cat.	Document frequency weight					
1	Unigram subtuples	0.836	0.864	0.874	0.881	**0.898**
2	Bi-normal separation	0.857	0.868	0.876	0.882	**0.897**
3	Log odds ratio	0.851	0.868	0.876	0.878	**0.896**
4	Forman log odds ratio	0.851	0.868	0.876	0.878	**0.896**
5	Pointwise mutual information	0.844	0.864	0.869	0.880	**0.895**
6	Yulle's ω	0.849	0.867	0.874	0.880	**0.894**
7	Yulle's Q	0.840	0.860	0.871	0.873	**0.892**
8	Zhang	0.834	0.857	0.862	0.869	**0.892**
9	δIDF	0.846	0.864	0.871	0.878	**0.891**
10	Collective strength	0.825	0.850	0.856	0.867	**0.890**
...						
73	IDF	0.743	0.807	***0.817***	0.814	**0.862**
Reuters 52 cat.	Document frequency weight					
1	Kappa	0.615	0.645	0.661	0.706	**0.765**
2	Normalized expectation	0.612	0.647	0.657	0.710	**0.763**
3	One-way Klosgen	0.585	0.620	0.639	0.691	**0.762**
4	Bi-normal separation	0.645	0.686	0.686	0.721	**0.762**
5	Jaccard	0.614	0.646	0.655	0.706	**0.761**
6	Poisson significance	0.630	0.660	0.670	0.723	**0.760**
7	Log likelihood ratio	0.627	0.651	0.658	0.692	**0.760**
8	J measure	0.627	0.654	0.659	0.693	**0.759**
9	One-way J measure	0.626	0.647	0.661	0.705	**0.759**
10	One-way Gini index	0.593	0.624	0.637	0.695	**0.759**
...						
86	IDF	0.380	0.500	***0.520***	0.576	**0.625**
Reuters 115 cat.	Document frequency weight					
1	Relevance frequency$_{OR}$	0.443	0.498	0.493	**0.574**	0.489
2	Bi-normal separation	0.474	0.514	0.533	**0.566**	0.508
3	Poisson significance	0.504	0.513	0.522	**0.565**	0.546
4	Log odds ratio	0.465	0.509	0.524	**0.563**	0.495
5	Forman log odds ratio	0.465	0.509	0.524	**0.563**	0.495
6	Pearson	0.482	0.509	0.521	**0.562**	0.539
7	Phi	0.482	0.509	0.521	**0.562**	0.539
8	Conviction	0.465	0.508	0.529	**0.561**	0.510
9	S cost	0.444	0.501	0.510	**0.560**	0.477
10	Mutual expectation	0.481	0.458	0.432	0.512	**0.557**
...						
81	IDF	0.362	0.417	***0.426***	**0.474**	0.380

Table 6 continued

Corpus	Term representation	t	t	t	(t, n)	(t, n) and (t, p)
	Term frequency weight	RTF	LTF	ITF	BIN	BIN
Ohsumed	Document frequency weight					
1	One-way Klosgen	0.538	0.569	0.575	0.595	**0.602**
2	Squared log likelihood ratio	0.540	0.557	0.564	0.589	**0.601**
3	Klosgen	0.536	0.565	0.570	0.591	**0.598**
4	One-way Gini index	0.553	0.562	0.567	0.586	**0.598**
5	Pearson's χ^2 test	0.553	0.562	0.568	0.587	**0.598**
6	Odds ratio	0.520	0.545	0.553	0.576	**0.594**
7	z Score	0.523	0.556	0.562	0.584	**0.592**
8	Pearson	0.522	0.556	0.562	0.583	**0.591**
9	Phi	0.522	0.556	0.562	0.583	**0.591**
10	J measure	0.537	0.552	0.555	0.561	**0.591**
...						
85	IDF	0.185	0.237	***0.255***	0.289	**0.319**
20 Newsgroups	Document frequency weight					
1	One-way Klosgen	0.737	0.764	0.770	0.773	**0.795**
2	z Score	0.718	0.761	0.768	0.773	**0.795**
3	Pearson	0.713	0.758	0.766	0.771	**0.793**
4	Phi	0.713	0.758	0.766	0.771	**0.793**
5	Bi-normal separation	0.669	0.743	0.753	0.754	**0.789**
6	Reverse-way BNS	0.672	0.743	0.752	0.755	**0.788**
7	Klosgen	0.712	0.748	0.757	0.765	**0.787**
8	One-way Laplace	0.676	0.738	0.747	0.754	**0.786**
9	Relative risk	0.697	0.749	0.756	0.761	**0.786**
10	Second Kulczynski	0.678	0.741	0.752	0.759	**0.785**
...						
78	IDF	0.403	0.595	***0.623***	0.633	**0.712**

extended term representation gives significant improvement to the classification. For example, the F_1-Score increased from 0.892 for TFIDF to 0.947 for Yulle's ω with the extended term representation (t, n) and (t, p) in the Reuters-21578 corpus with 10 categories. The best improvement is obtained in Ohsumed corpus as we move from a F_1-Score of 0.380 to 0.639 with one-way Klosgen metric with an extended term representation. However, we notice that the best metrics are very different according to the corpus used and whether the label distribution is balanced or not for Reuters-21578 corpus.

Table 6 provides the macroaveraged F_1-Scores of the top-10 metrics among all weighting schemes. We observe also that the proposed representation models (t, n) and (t, n) and (t, p) achieve better macroaveraged F_1-scores. However, the top-10 metrics when we consider the macroaveraged F_1-scores are generally different from the top-10 metrics considering the microaveraged F_1-scores.

We also notice that lots of metrics proposed in this paper for the first time in term weighting give better results than metrics previously used for this problem.

Table 7 Microaveraged and macroaveraged F_1-Scores of SVM methods and their combination with extended term representation (t, n) and (t, p) for different weighting metrics on Reuters (10, 52 and 115 categories), Ohsumed and 20 Newsgroups corpora

Document frequency weight	MiF	Document frequency weight	MaF
Reuters 10 categories			
Combination	0.952	Combination	0.910
Yulle's ω	0.947	Unigram subtuples	0.898
Log odds ratio	0.947	Bi-normal separation	0.897
Forman log odds ratio	0.947	Log odds ratio	0.896
Yulle's Q	0.946	Forman log odds ratio	0.896
Pointwise mutual information	0.946	Pointwise mutual information	0.895
Unigram subtuples	0.946	Yulle's ω	0.894
Bi-normal separation	0.946	Yulle's Q	0.892
δIDF	0.945	Zhang	0.892
Zhang	0.945	δIDF	0.891
Reverse-way BNS	0.944	Collective strength	0.890
Reuters 52 categories			
Combination	0.905	Combination	0.772
Bi-normal separation	0.893	Kappa	0.765
Log odds ratio	0.891	Normalized expectation	0.763
Forman log odds ratio	0.891	One-way Klosgen	0.762
Reverse-way BNS	0.890	Bi-normal separation	0.762
Probability-based term weight	0.889	Jaccard	0.761
δIDF	0.889	Poisson significance	0.760
Pointwise mutual information	0.889	Log likelihood ratio	0.760
Yulle's ω	0.888	J measure	0.759
Collective strength	0.887	One-way J measure	0.759
Relevance frequency variation	0.886	One-way Gini index	0.759
Reuters 115 categories			
Combination	0.888	Combination	0.570
Bi-normal separation	0.877	Mutual expectation	0.557
Log odds ratio	0.875	One-way Gini index	0.554
Forman log odds ratio	0.875	Second Sokal–Sneath	0.553
Reverse-way BNS	0.875	Pearson's χ^2 test	0.553
Probability-based term weight	0.873	Jaccard	0.552
Pointwise mutual information	0.873	First Kulczynski	0.551
Relevance frequency variation	0.872	*R* cost	0.551
z Score	0.872	*T* combined cost	0.551
Pearson	0.871	One-way Klosgen	0.551
Phi	0.871	One-way J measure	0.547

Table 7 continued

Document frequency weight	MiF	Document frequency weight	MaF
Ohsumed			
Combination	0.679	Combination	0.647
One-way Klosgen	0.639	One-way Klosgen	0.602
Klosgen	0.636	Squared log likelihood ratio	0.601
z Score	0.634	Klosgen	0.598
Pearson	0.633	One-way Gini index	0.598
Phi	0.633	Pearson's χ^2 test	0.598
Squared log likelihood ratio	0.631	Odds ratio	0.594
Odds ratio	0.629	z Score	0.592
One-way Gini index	0.629	Pearson	0.591
Pearson's χ^2 test	0.629	Phi	0.591
Sebag–Schoenauer	0.628	J measure	0.591

5.3.2 Classifier combination

As no metric gives the best results in all situations, we have tested classifier combination. The classification of a new document is done in two steps: We first compute classification scores with 96 base SVM learners, and each learner uses a different metric for weighting the features, and then, we use these scores as features for classifying the document with the secondary SVM learner.

Table 7 provides the F_1-Scores obtained by different weighting metrics and their combination when we use extended term representation (t, n) and (t, p) on Reuters (10, 52 and 115 categories), Ohsumed and 20 Newsgroups corpora. We can see that the performances of the classifier combination are always better according to all criteria: the microaveraged and the macroaveraged F_1-score for all the corpora. This confirms that by combining the predictions of several classifiers using different metrics one obtains better predictive performance than could be obtained from any of the constituent classifiers that use one metric.

The statistical significance of the achieved performances on 20 Newsgroups corpus are given in Table 8. Besides the fact that the F_1-score obtained by the classifier combination is more (in 20 Newsgroups) or less (in Reuters with 10 categories) significantly better than the best metric for each corpus, combination is the only method that gives good results for all corpora, all number of categories and both type of F_1-Scores (micro and macro).

5.4 Computation time and time complexity

Table 9 gives the number of features considered in the training set according to the term representations we have considered in our experiments. It shows a moderate growth in the number of features, by a factor of 3, when we consider term frequency and position features compared to traditional term features.

It is interesting to note in the same table that SVM learning computation time (on one processor of an Intel(R) Core(TM) i7-3520M at 2.90 GHz) is almost proportional to the number of features. Indeed, Thorsten Joachims showed that training linear SVM can be achieved in time $O(ns)$, where s is the number of nonzero features (terms) in each example

Table 8 Fivefold cross-validation performances (mean and standard deviation) of SVM methods and their combination with extended term representation (t, n) and (t, p) for different weighting metrics on 20 Newsgroups corpora

Document frequency weight	MiF	Document frequency weight	MaF
20 Newsgroups			
Combination	0.861±0.008	Combination	0.866±0.006
One-way Klosgen	0.790 ± 0.007	One-way Klosgen	0.795 ± 0.006
z Score	0.790 ± 0.008	z Score	0.795 ± 0.005
Pearson	0.788 ± 0.009	Pearson	0.793 ± 0.006
Phi	0.788 ± 0.009	Phi	0.793 ± 0.006
Bi-normal separation	0.785 ± 0.012	Bi-normal separation	0.789 ± 0.010
Reverse-way BNS	0.783 ± 0.010	Reverse-way BNS	0.788 ± 0.009
One-way Laplace	0.782 ± 0.009	Klosgen	0.787 ± 0.004
Klosgen	0.781 ± 0.006	One-way Laplace	0.786 ± 0.007
Relative risk	0.781 ± 0.009	Relative risk	0.786 ± 0.007
Second Kulczynski	0.781 ± 0.007	Second Kulczynski	0.785 ± 0.006

Table 9 Number of features considered in the training set of Reuters-21578, Ohsumed and 20 Newsgroups corpora and average SVM learning computation time for one metric and one category in each corpus

	Feature type	Number of features	Time (in s)
Reuters-21578			
Term features	t	20 767	0.203
Term frequency features	(t, n)	32 690	0.283
Term frequency and position features	(t, n) and (t, p)	61 096	0.627
Ohsumed			
Term features	t	175 803	2.254
Term frequency features	(t, n)	223 589	3.236
Term frequency and position features	(t, n) and (t, p)	500 474	8.082
20 Newsgroups			
Term features	t	393 797	8.306
Term frequency features	(t, n)	462 370	21.083
Term frequency and position features	(t, n) and (t, p)	861 749	33.028

(document) and n the number of examples [18], with the assumption that $s \ll N$, N being the number of features in the entire corpus. This last assumption is verified for a corpus of text documents. As we have already discussed in Sects. 3.2 and 3.3, the number of frequency and position features associated with a term t in each document grows with the logarithm of s. This means that the time complexity for training a corpus with documents containing at most s terms is $O(ns \log s)$. In practice, $\log s$ is small as it is confirmed by both the number of features and the computation time presented in Table 9.

Our classifier combination method implies the use of 96 SVM learners in the first step, then another SVM learner for the final classification. The theoretical time complexity is not affected because 96 is a constant value. However, in practice, it means that it multiplies the computation time by 96. In many text classification problems, one can afford such compu-

tation time constant growth in order to obtain significant improvement in the quality of the classification (from a microaveraged F_1-Score of 0.444 with IDF to 0.679 by combining 96 metrics in the case of Ohsumed corpus).

6 Conclusion

In this paper, we have studied 96 term-weighting metrics, and among them 80 metrics have not been used for this problem in the literature. Many of them provide better results than those already used for term weighting. We have also proposed an extended term representation where the term frequency and the term position in the document are adequately integrated to the document frequency. As no metric gives the best results according to whether the label distribution is balanced or not, we have proposed a classifier combination method with different metrics that performs well for both macroaveraged and microaveraged F_1-scores for different cases of label distribution.

Future work includes searching for superior weighting metrics, using other learning methods (Naives Bayes, centroid, etc.) and testing on large-scale benchmark data sets. In particular, it would be interesting to improve the computation time in order to apply our ideas on MEDLINE and Wikipedia corpora.

References

1. Aggarwal CC, Zhai C (2012) A survey of text classification algorithms. In: Aggarwal CC, Zhai C (eds) Mining text data. Springer, New York, pp 163–222
2. Altinçay H, Erenel Z (2010) Analytical evaluation of term weighting schemes for text categorization. Pattern Recognit Lett 31(11):1310–1323
3. Altinçay H, Erenel Z (2012) Using the absolute difference of term occurrence probabilities in binary text categorization. Appl Intell 36(1):148–160
4. Badawi D, Altinçay H (2014) A novel framework for termset selection and weighting in binary text classification. Eng Appl Artif Intell 35:38–53
5. Batal I, Hauskrecht M (2009) Boosting KNN text classification accuracy by using supervised term weighting schemes. In: Cheung DW-L, Song I-Y, Chu WW, Hu X, Lin JJ (eds), Proceedings of the 18th ACM conference on information and knowledge management, CIKM 2009. Hong Kong, China, November 2–6, 2009. ACM, pp 2041–2044
6. Bouillot F, Poncelet P, Roche M (2014) Classification of small datasets: why using class-based weighting measures?. In: Andreasen T, Christiansen H, Talavera JCC, Ras ZW (eds), Foundations of intelligent systems–21st international symposium, ISMIS 2014, Roskilde, Denmark, June 25–27, 2014. Proceedings, vol 8502 of Lecture notes in computer science, Springer, pp 345–354
7. Debole F, Sebastiani F (2002) Supervised term weighting for automated text categorization, Technical Report Technical Report 2002-TR-08. Istituto di Scienza e Tecnologie dellInformazione, Consiglio Nazionale delle Ricerche, Pisa, IT
8. Debole F, Sebastiani F (2003) Supervised term weighting for automated text categorization. In: Proceedings of the 2003 ACM symposium on applied computing (SAC), March 9–12, 2003. Melbourne, FL, USA. ACM, pp 784–788
9. Deng Z-H, Luo K-H, Yu H (2014) A study of supervised term weighting scheme for sentiment analysis. Expert Syst Appl 41(7):3506–3513
10. Deng Z-H, Tang S, Yang D, Zhang M, Li L, Xie K (2004) A comparative study on feature weight in text categorization. In: Yu JX, Lin X, Lu H, Zhang Y (eds), Advanced web technologies and applications, 6th Asia-Pacific web conference, APWeb 2004, Hangzhou, China, April 14–17, 2004, Proceedings, vol 3007 of Lecture notes in computer science, Springer, pp 588–597
11. Escalante HJ, García-Limón MA, Morales-Reyes A, Graff M, Montes-y-Gómez M, Morales EF, Martínez-Carranza J (2015) Term-weighting learning via genetic programming for text classification. Knowl Based Syst 83:176–189

12. Fattah MA (2015) New term weighting schemes with combination of multiple classifiers for sentiment analysis. Neurocomputing 167:434–442
13. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305
14. Forman G (2008) BNS feature scaling: an improved representation over tf-idf for svm text classification. In: Shanahan JG, Amer-Yahia S, Manolescu I, Zhang Y, Evans DA, Kolcz A, Choi K-S, Chowdhury A (eds), Proceedings of the 17th ACM conference on information and knowledge management, CIKM 2008, Napa Valley, California, USA, October 26–30, 2008. ACM, pp 263–270
15. Geng L, Hamilton HJ (2006) Interestingness measures for data mining: a survey. ACM Comput Surv 38(3):9
16. Guan H, Zhou J, Guo M (2009) A class-feature-centroid classifier for text categorization. In: Quemada J, León G, Maarek YS, Nejdl W (eds), Proceedings of the 18th international conference on world wide web, WWW 2009, Madrid, Spain, April 20–24, 2009. ACM, pp 201–210
17. Joachims T (1999) Making large-scale SVM learning practical. In: Schölkopf B, Burges C, Smola A (eds) Advances in kernel methods–support vector learning. MIT Press, Cambridge, pp 169–184 (Chapter 11)
18. Joachims T (2006) Training linear SVMs in linear time. In: Eliassi-Rad T, Ungar LH, Craven M, Gunopulos D (eds), Proceedings of the Twelfth ACM SIGKDD international conference on knowledge discovery and data mining. Philadelphia, PA, USA, August 20–23, 2006. ACM, pp 217–226
19. Ko Y (2015) A new term-weighting scheme for text classification using the odds of positive and negative class probabilities. J Assoc Inf Sci Technol 66:2553–2565
20. Lan M, Tan CL, Su J, Lu Y (2009) Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans Pattern Anal Mach Intell 31(4):721–735
21. Liu Y, Loh HT, Sun A (2009) Imbalanced text classification: a term weighting approach. Expert Syst Appl 36(1):690–701
22. Madjarov G, Kocev D, Gjorgjevikj D, Dzeroski S (2012) An extensive experimental comparison of methods for multi-label learning. Pattern Recognit 45(9):3084–3104
23. Martineau J, Finin T, Joshi A, Patel S (2009) Improving binary classification on text problems using differential word features. In: Cheung DW-L, Song I-Y, Chu WW, Hu X, Lin JJ (eds), Proceedings of the 18th ACM conference on information and knowledge management, CIKM 2009. Hong Kong, China, November 2–6, 2009. ACM, pp 2019–2024
24. Nguyen TT, Chang K, Hui SC (2013) Supervised term weighting centroid-based classifiers for text categorization. Knowl Inf Syst 35(1):61–85
25. Pecina P (2010) Lexical association measures and collocation extraction. Lang Resour Eval 44(1–2):137–158
26. Rehman A, Javed K, Babri HA, Saeed M (2015) Relative discrimination criterion–a novel feature ranking method for text data. Expert Syst Appl 42(7):3670–3681
27. Ren F, Sohrab MG (2013) Class-indexing-based term weighting for automatic text classification. Inf Sci 236:109–125
28. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
29. Tsoumakas G, Katakis I, Vlahavas IP (2010) Mining multi-label data. In: Maimon O, Rokach L (eds) Data mining and knowledge discovery handbook, 2nd edn. Springer, New York, pp 667–685
30. Tulyakov S, Jaeger S, Govindaraju V, Doermann DS (2008) Review of classifier combination methods. In: Marinai S, Fujisawa H (eds) Machine learning in document analysis and recognition, vol 90 of Studies in computational intelligence. Springer, New York, pp 361–386
31. Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Fisher DH (eds), Proceedings of the fourteenth international conference on machine learning (ICML 1997), Nashville, Tennessee, USA, July 8–12, 1997. Morgan Kaufmann, pp 412–420
32. Zhang M, Zhou Z (2014) A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng 26(8):1819–1837

Mounia Haddoud is a Ph.D. student in Computer Science from University of Rouen (France) and University of Sciences and Technology Houari Boumediène (USTHB, Algeria). Her research interests include text classification and automatic keyphrase extraction. She is currently Assistant Lecturer at USTHB.

Aïcha Mokhtari is currently Professor of Computer Science at the University of Sciences and Technology Houari Boumediène (USTHB, Algeria). Her research focuses on reasoning about knowledge and uncertainty, and its applications to access control, web semantic, distributed computing and ambient systems. Her work lies at the boundary of a number of fields. She usually teaches databases in an undergraduate course and knowledge representation and reasoning in a postgraduate course.

Thierry Lecroq got his Ph.D. in Computer Science from the University of Orléans (France) in 1992 and his Habilitation from the University of Rouen (France) in 2000. He is currently Professor of Computer Science at the University of Rouen (France). His main interest is string algorithms. He co-authored several books and chapters in collaborative books on this topic.

Saïd Abdeddaïm received his Ph.D. in Computer Science from the University of Pierre et Marie Curie (Paris 6) in 1996. He is currently Associate Professor of Computer Science at the University of Rouen (France). His research interests include text mining, string algorithms and bioinformatics.