Using Kullback-Leibler Distance for Text Categorization

Brigitte Bigi

CLIPS-IMAG Laboratory, UMR CNRS 5524
B.P. 53, 38041 Grenoble cedex 9, FRANCE
brigitte.bigi@imag.fr

Abstract. A system that performs text categorization aims to assign appropriate categories from a predefined classification scheme to incoming documents. These assignments might be used for varied purposes such as filtering, or retrieval. This paper introduces a new effective model for text categorization with great corpus (more or less 1 million documents). Text categorization is performed using the Kullback-Leibler distance between the probability distribution of the document to classify and the probability distribution of each category. Using the same representation of categories, experiments show a significant improvement when the above mentioned method is used. KLD method achieve substantial improvements over the *tfidf* performing method.

1 Introduction

Text Categorization is an important component of many large Information Retrieval or Machine Learning system. It is often defined as the content-based assignment of one or more predefined categories to texts. It is commonly conjectured that it is infeasible to manually classify all of the new documents that are added to a system in a timely manner. Therefore, automatic methods of document classification are needed.

Information processing needs have increased with the rapid growth of textual information sources, such as news media and the World Wide Web. Text retrieval systems find or route texts in response to arbitrary user queries or interest profiles. Text categorization can be used to support Information Retrieval or to perform information extraction, document filtering and routing to topic-specific processing mechanisms.

Recent research has been concerned with scaling-up (e.g. data mining). Text categorization is a domain where large data sets are available and which provides an application field to Machine Learning. Indeed, manual categorization is known to be an expensive and time-consuming task. Machine Learning approaches to classification (text categorization is a classification task) suggest the construction of categorization means using induction over pre-classified samples. They have been rather successfully applied in various studies.

A growing number of statistical classification and machine learning techniques have been applied to text categorization [1], including nearest neighbor classifiers [2], probabilistic bayesian models [3], neural networks [4], etc. Term-frequency/inverse-document-frequency (*tfidf*) [5] is the common term weighting method and a cosine similarity is used for the categorization. In the paper [6], the author presents an analysis of the word weighting scheme based on *tfidf* and the similarity metric. The empirical results suggest that a probabilistically founded modelling is preferable to the heuristic *tfidf* modelling. Moreover, the author says that the probabilistic methods are preferable from a theoretical viewpoint because they are more well founded. The paper [7] presents a controlled study with significance analyses on five text categorization methods: the Support Vector Machine (SVMs), a k-Nearest Neighbor (kNN) classifier, a neural network (NNet) approach, the Linear Least-squares Fit (LLSF) mapping and a Naive Bayes (NB) classifier. It suggests that SVMs and kNN significantly outperform the other classifiers. In the paper [8], the author explores the use of SVMs for learning text classifiers, and this method achieve substantial improvements over others compared methods, including Rocchio algorithm. Moreover, in the paper [9] authors compare the effectiveness of five different automatic learning algorithms for text categorization and observe that SVMs are particularly promising. It is commonly conjectured that SVMs is the best categorization method for small corpus (around 10,000 documents). In [10], SVMs are applied on a corpus made of about 42,000 documents. Nevertheless, the problem that we put forward in the literature is the small size of the corpus. Especially as with the rapid growth of online information: text categorization has become one of the key techniques for handling and organizing numerous data. As an example, one year of the Reuters corpus is composed about 807,000 news stories. We do not know how the SVMs can be applied in this case because the literature do not explore these conditions.

The method proposed in this paper is based on the symmetric Kullback-Leibler divergence, also called Kullback-Leibler distance measure, well known in Information Theory [11]. We propose to perform text categorization using this distance between the probability distribution of the document to classify and the probability distribution of each category. In information retrieval, the Kullback-Leibler divergence is used for query expansion in [12]. The approach is simple and very efficient (tests are made on TREC 7 and 8). Authors introduce a new term-scoring function that is based on the differences between the distribution of terms in relevant documents and the distribution of terms in all documents.

This paper explores and identifies the benefits of Kullback-Leibler distance for text categorization. The size of textual data is itself a challenge: a real-size corpus, composed of several hundred of thousand texts, may include several thousand of words. The organization of the paper is as follows. Section 2 presents the classical *tfidf* classifier. Section 3 is devoted to the KLD-based method we propose. In the last section, performance on a corpus derived from the Reuters is summarized and analyzed. The resulting categorization rates compare favorably for our method with those of the standard method.

2 The reference model based on tfidf

2.1 The TfIdf Term Weighting

One of the most common weighting used is referred to as term-frequency/inverse-document-frequency (*tfidf*) [5]. Documents are represented by term vectors of the form $d = (t_i, t_j, ..., t_p)$ where each t_k identifies a *content term* assigned to some sample document d as is done in the popular vector representation for information retrieval [5]. Typically, each j^{th}document d is represented as a vector of *weights* $\overrightarrow{d_j} = \left(w_{1j}, w_{2j}, ..., w_{|V|j}\right)$ of the content terms selected, where V is the set of terms that occur at least once in at least one document, and w_{kj} represents how much term t_k contributes to the semantics of document d_j. Each element w_{kj} is calculated as a combination of the statistics $tf(t_k, d_j)$ and $idf(t_k)$ [13]. This weighting scheme starts with the frequency of a term in a given document $tf(t_k, d_j)$, and multiplies this by the "inverse document frequency" $idf(t_k)$ of the term in the corpus. The *idf* of a term is lower the more documents appears in. The idea is that the more documents a word apprears in, the less likely it is to be a good measure for distinguishing one document from another. The *tfidf* formula for a term t_k is as follows:

$$w_{kj} = tf(t_k, d_j) \times idf(t_k)$$

where $tf(t_k, d_j)$ is equal to 0 when term t_k is not assigned to document d_j, and equal to $\#(t_k, d_j)$ for the assigned terms. The *idf* term is calculated as follows:

$$idf(t_k) = \log\left(\frac{|T_r|}{df(t_k)}\right)$$

where T_r is the set of training documents ($|T_r|$ is the total number of documents in the training) and $df(t_k)$ is the document frequency for the term t_k.

The *tfidf* word weighting heuristic says that a term t_k is an important indexing term for document d_j if it occurs frequently in it (the term frequency is high). On the other hand, terms which occur in many documents are rated less important indexing terms due to their low inverse document frequency. However, in many cases, the added "information" contained in the *idf* is not needed for a particular algorithm, just the term frequency tf can be used for a weighting scheme. Moreover, calculating the *idf* of a term requires a count across all documents in a corpus.

2.2 The classifier

The construction of a text categorization classifier for category $c_i \in C$ usually consists in the definition of a function that, given a document d_j returns a categorization status value for it. There are various policies for determining this measure, and the most common is defined in [5] as a cosine similarity, which represents the cosine of the angle that separates the two vectors $\overrightarrow{c_i}$ and $\overrightarrow{d_j}$:

$$similarity(\overrightarrow{c_i}, \overrightarrow{d_j}) = \frac{\sum_{k=1}^{|V|}(w_{ki} \times w_{kj})}{\sqrt{\sum_{k=1}^{|V|}(w_{ki})^2 \times \sum_{k=1}^{|V|}(w_{kj})^2}} \qquad (1)$$

where $w_{kj} = tf(t_k, d_j) \times idf(t_k, d_j)$ and $w_{ki} = tf(t_k, c_i) \times idf(c_i)$. All the comparisons between the document and the category vectors provides ranked category output in decreasing order of the computed similarity between $\overrightarrow{c_i}$ and $\overrightarrow{d_j}$. The document is assigned to the category with which its document vector has the highest cosine:

$$H_{tfidf}(d_j) = \arg\max_{c_i \in C} \ similarity(\overrightarrow{c_i}, \overrightarrow{d_j})$$

To compute w_{kj}, $tf(t_k, d_j)$ represents the number of times t_k apprears in $d_j \in T_r$. To compute w_{ki}, a category learning model, as defined previously for documents, is needed. For the vector $\overrightarrow{c_i}$, it is possible to use the *category frequency*. This problem will be discussed in section 4, because unlike in text retrieval, in text categorization the high dimensionality of the term space (i.e. the large value of $|V|$) may be problematic.

3 The KLD Classifier

This model makes use of term sets automatically selected for each category c_i. Let $|C|$ be the number of categories and V the vocabulary made of the union of all terms for all categories. For each topic category, a statistical distribution $P(t_k \mid c_i)$ *made only of the selected terms* is obtained from a training corpus. Such a distribution is compared with the distribution of the content of the document to classify. A word is considered in the document if and only if it belongs to any category-terms list. The document content, which is limited to terms, is compared with each category term probability distribution. The comparison is performed introducing a symmetric Kullback-Leibler (KL) divergence. As the document may contain only a limited number of terms in comparison to categories, the frequency of many terms in the document is zero. This causes problems in the KL distance computation when probabilities are estimated by frequencies of occurrence. In order to avoid them, a special type of back-off scheme is introduced in this paper.

3.1 Kullback-Leibler distance

Kullback and Leiber in 1951 [14] studied a measure of information from statistical aspects of view, involving two probability distributions associated with the same experiment, calling discrimination function, later different authors named as cross entropy, relative information, etc. The Kullback-Leibler divergence - also known as the relative entropy, is a measure of how different two probability distributions (over the same event space) are. The KL divergence of probability distributions P, Q on a finite set χ is defined as:

$$D(P||Q) = \sum_{x \in \chi} P(x) log \frac{P(x)}{Q(x)} \tag{2}$$

The KL divergence between P and Q can also be seen as the average number of bits that are wasted by encoding events from a distribution P with a code based on a not-quite-right distribution Q. This KL divergence is a non-symmetric information theoretic measure of distance of P from Q. The smaller the relative entropy, the more similar the distribution of the two variables, and conversely.

It has to be noted that the measure is asymmetrical. During the past years, various measures have been introduced in the literature generalizing this measure. Since the expression of equation 2 is not symmetric, it is not strictly a distance metric. We therefore use the symmetric Kullback-Leibler divergence i.e. the Kullback-Leibler Distance (KLD) metric as:

$$D(P||Q) = \sum_{x \in \chi} \left((P(x) - Q(x))\, log \frac{P(x)}{Q(x)} \right) \tag{3}$$

KL or KLD have been used in many natural language applications as for query expansion [12]. They have also been used, for example, in natural language and speech processing applications based on statistical language modeling [15], and in information retrieval, for topic identification [16], for choosing among distributed collections [17]. Here, the idea is that categories to be considered for document are those which mostly contribute to the distance defined in the equation 3.

The text categorization model proposed in this paper shares with other commonly used models the assumpion that a document is properly represented by a vector of weights. Each weight corresponds to a word called term and belonging to a vocabulary V. In this model, a document is represented by a term vector of probabilities $\overrightarrow{d_j}$ while a category is represented by a term vector of probabilities $\overrightarrow{c_i}$. The distance measure should be that which maximizes the KLD (the symmetric Kullbach-Leibler divergence defined in Information Theory as equation 3) between the document represented in $\overrightarrow{d_j}$ and the category $\overrightarrow{c_i}$.

3.2 The probability distributions

As mentionned above, the term probability distribution of a document is compared with each category probability distribution. A *back-off model* is proposed in which term frequencies appearing in the document are discounted and all the terms which are not in the document are given an epsilon probability equal to the probability of unknow words. The reason is that in practice, often not all the terms in V appear the documented represented in d_j. Let $V(d_j) \subset V$ be the vocabulary of the terms which do appear in the documents represented in d_j. For the terms not in $V(d_j)$, it is useful to introduce a back-off probability for $P(t_k, d_j)$ when t_k does not occur in $V(d_j)$, otherwise the distance measure will be infinite. The use of a back-off probability to overcome the data sparseness problem has been extensively studied in statistical language modelling (see, for example, [18]).

The resulting definition of document probability $P(t_k, d_j)$ is:

$$P(t_k, d_j) = \begin{cases} \beta P(t_k \mid d_j) & if\; t_k\; occurs\; in\; the\; document\; d_j \\ \varepsilon & else \end{cases} \tag{4}$$

with:

$$P(t_k \mid d_j) = \frac{tf(t_k, d_j)}{\sum_{x \in d_j} tf(t_x, d_j)}$$

where:

- $P(t_k \mid d_j)$ is the probability of the term t_k in the document d_j with $\sum_{x \in d_j} tf(t_x, d_j) = 1$;
- β is a normalisation coefficient which varies according to the size of the document;
- ε is a threshold probability for all the terms not in d_j.

The probability of a term t_k in a category c_i is expressed as:

$$P(t_k, c_i) = \begin{cases} \gamma.P(t_k \mid c_i) & if\ t_k\ occurs\ in\ the\ category\ c_i \\ \varepsilon & else \end{cases} \tag{5}$$

with:

$$P(t_k \mid c_i) = \frac{tf(t_k, c_i)}{\sum_{x \in c_i} tf(t_x, c_i)}$$

where:

- $P(t_k \mid c_i)$ is a category unigram probability of t_k in c_i with $\sum_{x \in c_i} tf(t_x, c_i) = 1$;
- γ is a normalisation coefficient;
- ε is the same probability in equation (5) as in equation (4) for all the terms not in c_i.

3.3 Constraints on the coefficients

β, γ and the ε value have to be chosen in order that the corresponding probabilities sum to 1.

The γ estimation
Equation (5) must respect the following constraint:

$$\sum_{k \in c_i} \gamma.P(t_k \mid c_i) + \sum_{k \notin c_i, k \in V} \varepsilon = 1$$

The γ can be easily estimated as follows:

$$\gamma = 1 - \sum_{k \notin c_i, k \in V} \varepsilon$$

Constraints on ε
ε is a threshold probability given to terms not in the document in equation (4), or given to terms not in the category in equation (5). Thus, this probability must be smaller than the minimum probability of a term in the document, and must be smaller than the minimum probability of a term in a category (i.e. smaller than $P(t_k \mid c_i)$ for each possible term t_k in c_i). Consequently, this value is obtained experimentaly.

The β estimation

Equation (4) must respect the following property:

$$\sum_{k \in d_j} \beta.P(t_k \mid d_j) + \sum_{k \notin d_j, k \in V} \varepsilon = 1$$

β can be easily estimated for a document with the following computation:

$$\beta = 1 - \sum_{k \notin d, k \in V} \varepsilon$$

3.4 Using KLD for text categorization

The categorization method based on the Kullback-Leibler distance computes the distance as follows:

$$KLD(c_i, d_j) = \sum_{k \in V} \left\{ (P(t_k, c_i) - P(t_k, d_j)) \times log \left(\frac{P(t_k, c_i)}{P(t_k, d_j)} \right) \right\} \tag{6}$$

This computation involves four cases:

1. $(t_k \in d_j) \wedge (t_k \in c_i)$, i.e. the term t_k appears in the document d_j and in the category c_i;
2. $(t_k \in d_j) \wedge (t_k \notin c_i)$, i.e. the term t_k appears in the document d_j but not in the category c_i;
3. $(t_k \notin d_j) \wedge (t_k \in c_i)$, i.e. the term t_k appears in the category c_i but not in the document d_j;
4. $(t_k \notin d_j) \wedge (t_k \notin c_i)$, i.e. the term t_k does not appears in the document d_j and in the category c_i.

As mentionned above, any term which is not a category-term, has a probability assigned to ε in $P(t_k, c_i)$ and the same probability in $P(t_k, d_j)$ (case 4) ; thus, its contribution to the KL distance is null. That is the reason why these terms do not need to be represented in the document. It is the case for all unknown terms regarding to V.

For each category, it is necessary to normalize the distance because the categories are very differents. Consequently, we use the following Kullback-Leibler normalized:

$$KLD^\star(c_i, d_j) = \frac{KLD(c_i, d_j)}{KLD(c_i, 0)}$$

where $KLD(c_i, 0)$ represents the distance of equation (6) between a category c_i and an empty document. The distribution probability of an empty document is an ε probability for all words of the vocabulary.

The document d_j is assigned to the category with which its document has the smallest $KLD^\star$ measure:

$$H_{KLD^\star}(d_j) = \arg\min_{c_i \in C} KLD^\star(c_i, d_j)$$

4 Experimental Results

This paper describes the results of experiments run on print news stories to test the categorization method based on the Kullback-Leibler distance. Our primary aim is to apply the KLD method to text categorization and estimate its capability and not to study the category learning. It is the reason why our results are not optimal. In future works, the results will be improved by a rigorously study of the category learning problem.

4.1 Corpus

We carried out our experiments on the Reuters dataset of newswire stories from exactly one year over 1996-1997. A description and some statistics about this corpus are available on the web (http://about.reuters.com/researchandstandards/corpus/statistics/index.asp). The corpus consists of 806,791 XML files in NewsML format (approximatively 3.7 Gb of uncompressed data). We divide this corpus in a learning set, a development set (not used in experiments described in this paper) and a test set as shown in figure 1.

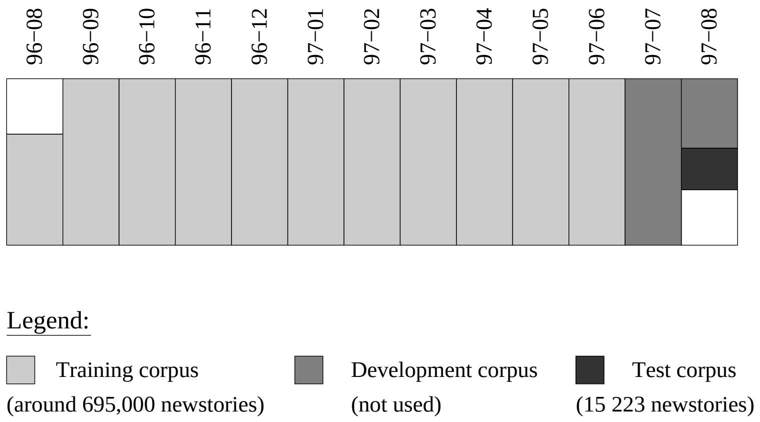


Fig. 1. The Reuters Corpus used in experiments

Each strory is manually indexed by zero to several topics. In figure 2 we report some statistics about this topic indexing distributed by Reuters. These charts are available at the following addresses:
http://about.reuters.com/researchandstandards/corpus/statistics/topic_count.gif
http://about.reuters.com/researchandstandards/corpus/statistics/topics.gif

We work on the set of 126 topics/categories that were provided with the formatted version of the corpus. All the performances were assessed by measuring the ability of the methods to reproduce manual assignments on a given dataset.

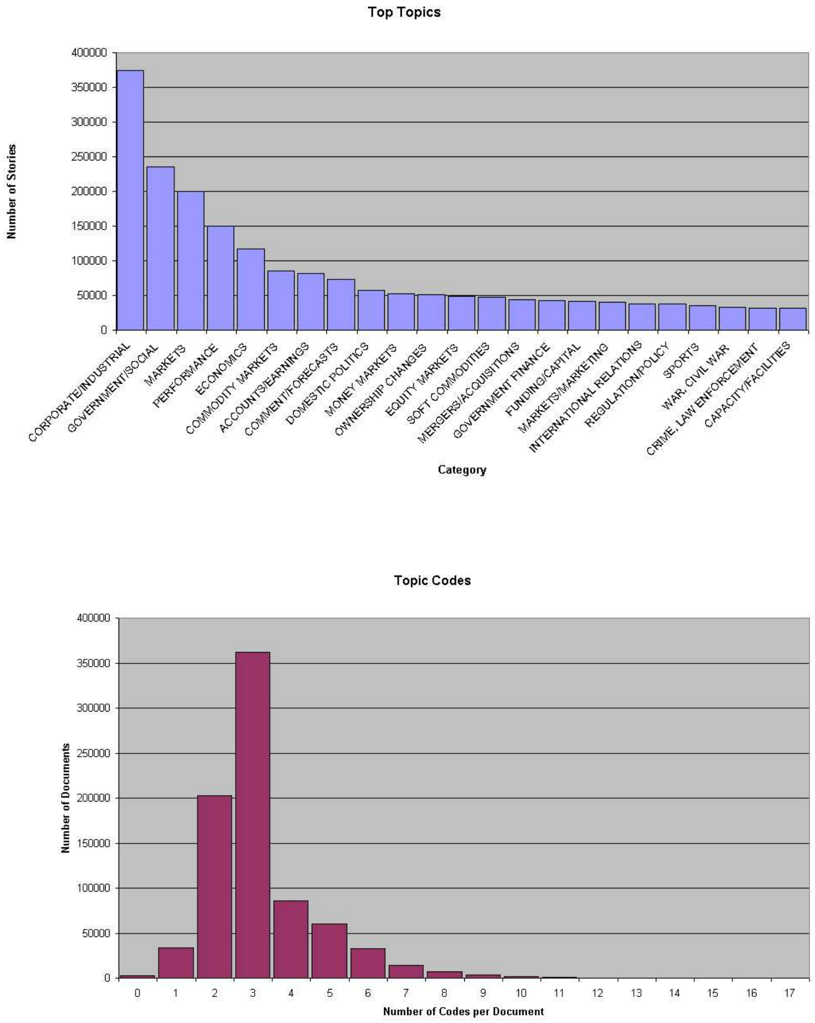


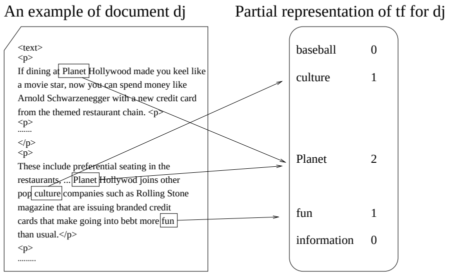
Fig. 2. Top topics and Topic Codes of the Reuters Corpus

4.2 Category learning

In order to make a category decision, a representation of categories must be chosen. As it is commonly made, we introduce one or more intermediate steps between the input representation of documents and the output category representation. The first step is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. All data (training and test) are filtered to extract only the body of newstories (the title, etc. are ignored). In our experiments, we use words without more complicated representation as it is recommanded in [9]. Indeed, authors found that the simplest document representation (using individual words delimited by white spaces with no stemming) was at least as good as representations involving more complicated syntactic and morphological analysis. And, representing documents as binary vectors of words, chosen using a mutual information criterion for each category, was as good as finer-grained coding. Stemming can be avoided as shown in [19]. Consequently, the case of characters has only been lowered. Several techniques are possible to select category-dependent terms. We have chosen to ranking the training corpus words and selecting the first K words of this ordered list. For this purpose, words like numbers or determinants are not good candidates. These words have been identified and placed into a stop-list (318 stop-words for the examples described in this paper). For each category, the ranked words according to their frequency in a training corpus are selected only if they are not in the stop-list. Thus, the text categorization methods make use of sets of K words automatically selected by frequency order. The category parameters are then estimated as:

$$\overrightarrow{c_i} = \sum_{\overrightarrow{d_j} \in C_i} \overrightarrow{d_j}$$

where $\overrightarrow{d_j} = (tf(w_1, d_j), tf(w_2, d_j), ..., tf(w_N, d_j))$ and w_n are words of training documents, as described in the figure 3. When a category is too small (i.e. less than K words), it is ignored in the evaluation (but these categories can appear in the test corpus).


An example of document dj
Partial representation of tf for dj

<text>
<p>
If dining at Planet Hollywood made you keel like
a movie star, now you can spend money like
Arnold Schwarzenegger with a new credit card
from the themed restaurant chain. <p>
<p>
....
</p>
<p>
These include preferential seating in the
restaurants, ... Planet Hollywod joins other
pop culture companies such as Rolling Stone
magazine that are issuing branded credit
cards that make going into bebt more fun
than usual.</p>
<p>
....

baseball	0
culture	1
Planet	2
fun	1
information	0

Fig. 3. Representation of document term frequencies in the training

Our aim is to apply the KLD method (using probabilities term weighting) to a text categorization task and to compare the results obtained using the well-known similarity based method (using *tfidf* term weighting). To do so, we conducted a set of 3 experiments resulting from 3 different term selections:

1. In the first experiment, we use all words of the documents not in the stop-list, not numbers or punctuations.
2. The second experiment uses the same as the previous one and adds a lexicon to select words. This lexicon is composed of 86,000 entries.
3. The third experiment use the same filter as the first one and add a term selection at the document-level. We select a maximum of 50 words by document with the mutual information measure [20]. These measures allow to compute the association degree between a word in the document d_j compared to this word in the corpus and then to make up lists of the most important words for this document. This average mutual information (MI) measure between A and B can be evaluated as:

   $$MI(A:B) = \sum_{a,b} P(a,b) log \frac{P(a,b)}{P(a).P(b)}$$

   A value of MI is computed for each word of a document. These values are then ranked, and the 50 best MI are selected as terms for the document. These features are used as input to learn categories, with a term frequency of 1 for each word by document.

In these 3 experiments, the KLD method uses probabilities from the learned categories as defined previously in Section 3. The *tfidf* method assigns w_{kj} for a term t_k in a category c_i as defined in Section 2, i.e. as follows:

$$w_{kj} = tf(t_k, c_j) \times log\left(\frac{|c_i|}{df(t_k, c_i)}\right)$$

4.3 Evaluation Criteria

For each category, the categorization goal is viewed as a binary classification problem. Given a category, the categorization methods decide whether each document is in or not in this category. With a single category as the focus, let:

- MA be the number of documents assigned to the category both manually and automatically,
- A be the number of documents assigned to the category automatically but not manually,
- M be the number of documents assigned to the category manually but not automatically.

Then the two common measures of Recall (R) and Precision (P) can be defined as:

$$R = \frac{MA}{MA+M}$$
$$P = \frac{MA}{MA+A}$$

Now, theses measures are adapted to the categorization decisions. Given a document and a category, a categorization decision is made to determine whether or not to assign this category to the document. When automatic categorization is conducted, a number of these decisions are made. Out of these decisions, some may match with the manual ones, while others may not. We want to compare the various automated decisions with the manual ones. An "assignment" is defined as the positive decision to assign a category to a document. Let:

- c be the number of correct assignments made automatically,
- a be the number of assignments made automatically,
- m be the number of assignments made manually.

Then, we can define the recall (r) and precision (p) measures as follows:

$$r = \frac{c}{m}$$
$$p = \frac{c}{a}$$

Recall r is the proportion of correctly predicted YESes by the system among the true YESes for all the document category pairs given a dataset. Precision p is the proportion of correctly YESes among all the system predicted YESes. These values consider that the model give a single solution and the notion of precision can be extanded. The precision-at-N is the proportion of correctly YESes in the N first solutions of the model among all the system predicted YESes.

4.4 Overall Performance

Table 1 summarizes our results obtained for the test set. The vocabulary selection of terms for each category is a subset of $K = 2,000$ terms of the category vocabulary. We have not rigorously explore the optimum number of terms for this problem, but this number provided goog results. As we said above, the vocabulary V is the union of all terms of all categories. With this size by category we obtain a vocabulary size $|V|$ of 101,276 in the first experiment, 11,281 in the second experiment and 22,056 in the third experiment.

Table 1. Results of text categorization for the $tfidf$ and KLDmethods

$tfidf$	Recall	Precision-at-1	Precision-at-5	Precision-at-10
First experiment	0.449	0.145	0.475	0.606
Second experiment	0.480	0.155	0.552	0.726
Third experiment	0.613	0.198	0.635	0.783

KLD	Recall	Precision-at-1	Precision-at-5	Precision-at-10
First experiment	0.624	0.221	0.535	0.601
Second experiment	0.649	0.210	0.630	0.790
Third experiment	0.731	0.236	0.734	0.874

We can see in table 1 that KLD method performs best among the conventional method. Compared to the *tfidf* method, all KLD results perform best independently of the choice of category selection features. It is well known that the feature selection is an important goal, and we observe the same in these experiments: the third experiment is significantly better than the first and the second one.

Finally, only for comparison, we carried out some experiments on the well-known Reuters-21578 corpus. We do not detail these experiments in this paper because this work is still in hand. Some results are also available. The corpus consists of a set of 21,578 Reuters newswire stories from 1987 which have been indexed manually using 135 financial topics to support document routing and retrieval for Reuters customers. We divided this corpus into a training set containing 16,300 stories and a test set containing all the other 5,278 stories. Several thousand documents in the data set have no topic assignments and we have chosen to ignore them as we cannot possibly learn from them. The resulting test set is composed of 2,475 stories. The results presented in table 2 refer to experiments made in the same conditions as the previous ones excepted for the K value because the categories are too small. The results remain favorable to KLD.

Table 2. Preliminary results on Reuters-21578

tfidf	Recall	Precision	*K*
First experiment	0.731	0.545	200
Third experiment	0.755	0.564	200
Third experiment	0.723	0.538	50

KLD	Recall	Precision	*K*
First experiment	0.785	0.585	200
Third experiment	0.799	0.597	200
Third experiment	0.749	0.557	50

At last, comparing time of the two methods, the training time is the same for the two methods since we used the same category learning! The time to estimate the appropriate categories for a document is around 20% to 400% faster for KLD.

5 Conclusion and Future works

This paper introduces a new effective method to perform text categorization. It provides that KLD based method is well suited for this task even in the following conditions: high number of documents and high dimentional feature space. The experimental results show that KLD consistently achieve good performance on text categorization task, outperforming the reference method substantially and significantly. All this makes KLD a very promising and easy-to-use method for text categorization.

We think that our research directions are experimental ones. We want to continue to validate the KLD method. We will initially test other solutions issued from the literature to learn categories. We will investigate the role of document length in this step, looking for correspondence between variations in document length and the comparative performances of KLD and $tfidf$. An other solution we want to experiment is proposed in [21] ; learning is achieved by combining document vectors into a prototype vector $\vec{c_i}$ for each category. As described in [6], the vector is calculated as a weighted difference between the normalized document vectors of the positive examples for a category and the normalized document vectors of the negative examples, as follow:

$$\vec{c_i} = \alpha \frac{1}{|c_i|} \sum_{\vec{d} \in c_i} \frac{\vec{d}}{\left\| \vec{d} \right\|} - \beta \frac{1}{|T_r - c_i|} \sum_{\vec{d} \in (T_r - c_i)} \frac{\vec{d}}{\left\| \vec{d} \right\|}$$

where $|c_i|$ is the number of documents assigned to c_i ;$\left\| \vec{d} \right\|$denotes the Euclidian length of a vector $\vec{d}$; α and β are parameters that adjust the relative impact of positive and negative training examples. Because we think it is difficult to use SVMs on our large corpus, after these experiments we will validate the KLD method with the Reuters-21578 dataset even if our method was imagined for larger corpora. Finally, we are motivated to attempt to build a better "meta-classifier" (as for example in [22]) resulting from the combination of KLD and SVMs because they are qualitatively different.

We also plan experiments with varying amounts of training data because we hypothesize that the optimal vocabulary size may change with the size of the training set.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
2. Yang, Y.: Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1994) 13–22
3. Lewis, D., Ringuette, M.: A comparison of two learning algorithms for text categorization. In: Proceedings of the Third Annual Symposium on Document Analsysis and Information Retrieval. (1994) 81–93
4. Wiener, E., Pedersen, J., Weigend, A.: A neural network approach to topic spotting. In: Proceedings of the Fourth Annual Symposium on Document Analsysis and Information Retrieval. (1995)
5. Salton, G., McGill, M.: The smart and sire experimental retrieval systems, McGraw-Hill, New York (1983) 118–155
6. Joachims, T.: A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Fisher, D.H., ed.: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US, Morgan Kaufmann Publishers, San Francisco, US (1997) 143–151

7. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proceedings of ACM Conference on Research and Development in Information Retrieval. (1999) 42–49
8. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of the European Conference on Machine Learning, Springer (1998)
9. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of ACM-CIKM98. (1998) 148–155
10. Kindermann, J., Paass, G., Leopold, E.: Error correcting codes with optimized kullback-leibler distances for text categorization. In Raedt, L., ed.: Principles of data mining and knowledge discovery. (2001) 133–137
11. Cover, T., Thomas, J.: Elements of Information Theory. Wiley (1991)
12. Carpineto, C., De Mori, R., Romano, G., Bigi, B.: An information theoretic approach to automatic query expansion. ACM Transactions On Information Systems **19** (2001) 1–27
13. Salton, G.: Developments in automatic text retrieval. Science **253** (1991) 974–980
14. Kullback, S., Leibler, R.: On information and sufficiency. **22** (1951) 79–86
15. Dagan, I., Lee, L., Pereira, F.: Similarity-based models of word cooccurrence probabilities. Machine Learning **34** (1999) 43–69
16. Bigi, B., De Mori, R., El-Bèze, M., Spriet, T.: A fuzzy decision strategy for topic identification and dynamic selection of language models. Special Issue on Fuzzy Logic in Signal Processing, Signal Processing Journal **80** (2000)
17. Xu, J., Croft, B.: Cluster-based language models for distributed retrieval. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA (1999) 254–261
18. De Mori, R.: SPOKEN DIALOGUES WITH COMPUTERS. Academic Press (1998)
19. Leopold, E., Kindermann, J.: Text categorization with support vector machines: How to represent texts in input spaces? Machine Learning **46** (2002) 423–444
20. Rosenfeld, R.: A maximum entropy approach to adaptive statistical language modeling. Computer, Speech and Language **10** (1996) 187–228
21. Buckley, C., Salton, G., Allan, J.: The effect of adding relevance information in a relevance feedback environment. In: Proceedings of the seventeenth annual international ACM-SIGIR conference on research and development in information retrieval, Springer-Verlag (1994)
22. Bennett, P., Dumais, S., Horvitz, E.: Probabilistic combination of text classifiers using reliability indicators: Models and results. In: Proceedings of ACM International Conference on Research and Development in Information Retrieval. (2002) 207–214