APPLICATION OF A NEW SET OF PSEUDO-DISTANCES IN DOCUMENTS CATEGORIZATION

S. Gadri, Moussaoui A.*

Abstract: Automatic text classification is a very important task that consists in assigning labels (categories, groups, classes) to a given text based on a set of previously labeled texts called training set. The work presented in this paper treats the problem of automatic topical text categorization. It is a supervised classification because it works on a predefined set of classes and topical because it uses topics or subjects of texts as classes. In this context, we used a new approach based on k-NN algorithm, as well as a new set of pseudo-distances (distance metrics) known in the field of language identification. We also proposed a simple and effective method to improve the quality of performed categorization.

Key words: N-grams, language identification, text categorization, text mining, machine learning, Kullback-Leibler distance, \( \chi^2 \) distance, Cavnar-Trenkle distance

Received: October 12, 2014 DOI: 10.14311/NNW.2017.27.011
Revised and accepted: April 29, 2017

1. Introduction

Text classification has as objective to group thematically similar texts in a single set (group/class). The importance of such categorization approach is to organize knowledge so that some specific treatments can be performed, including: information retrieval and efficient information extraction. The increasing volume of digital documents available on networks, the need for automatic classification was felt both on the internet (search engines) and within companies (classification of internal documents, dispatches of news agencies, ...). There exist two approaches of automatic text classification: unsupervised classification (clustering) in which groups of documents are formed automatically by the machine during the treatment according to similarity criteria [12, 25] and supervised classification (categorization) in which these groups are defined in advance by an expert [15, 27, 31]. In this paper, we study the supervised classification. We use a k-NN approach based on similarity metrics known in the field of language identification. We also propose a new simple and effective method to improve the obtained results. The

*Said Gadri – Corresponding author; Abdelouahab Moussaoui; Department of Computer Science, Faculty of Sciences, University of Setif, Setif 19000, Algeria, E-mail: kadri.said21@gmail.com, moussaoui.abdel@gmail.com

©CTU FTS 2017 231
paper is organized as follows: in Section 2 we present a state of the art about
the achieved domain, in particular: text representation approaches, categorization
methods and similarity metrics used in the following areas: text categorization,
language identification. Section 3 presents clearly our contribution in this paper,
in particular, the application of similarity metrics used in language identification
in the field of text categorization, the proposal of a new simple and flexible method
for text categorization. Section 4 shows the experiments carried out on training
and test corpora and the obtained results. In Section 5 we try to evaluate the
obtained results and compare them to existing works. We conclude our paper by
Section 6 which summarizes the realized work and proposes some ideas for possible
improvements and future work in the same context.

2. State of the art

2.1 Language identification and documents categorization

2.1.1 Language identification

Language identification can be defined as the assignment of a text to a given lan-
guage. So, this is a kind of automatic classification where classes are languages
\( \text{Ar}, \text{En}, \text{Fr}, \ldots \). This identification becomes important because of the increasing
availability of textual data expressed in different languages on the web. A real
recognition of the text language is not possible if we just consider the word as a
basic unit of information, it could be possible for some languages such as French
or English, but very difficult for some other languages such as Arabic, German or
Chinese. One alternative approach consists in segmenting texts into characteristic
\( N \)-grams.

2.1.2 Automatic text categorization

Automatic text classification is a complex process whose objective is to find an
efficient algorithm that permits to assign a text to one or many classes (categories,
groups, labels, topics) with the highest success rate. We distinguish two types of
categorization; single-label categorization in which each document belongs to ex-
actly one category and multi-label categorization in which a document may belong
to any number of categories. In the present work we focused on the single-label
categorization. Hence, a similarity measure is required to find documents relevant
to a given query in information retrieval (IR) and to find the category closest to a
given document in text categorization (TC). One well-known application of TC is
the assignment of an article to one of yahoo groups.

2.2 Approaches of text representation

In the field of text categorization, learning algorithms are not able to treat texts
directly. This is why a very important preliminary phase known as representation
phase is necessary. This phase consists in representing each document to categorize
by a vector, whose components are: words, sentences, or other lexemes in order to
make it exploitable by learning algorithms.

232
2.2.1 Representation with bag of words

It is the simplest text representation model which has been used in the field of text categorization [25, 26]. In this representation, all texts are transformed into vectors in which each component represents a term. Thus, terms are the words which constitute the text. One problem of this representation that the size of the vector representing the document is equal to the size of the vocabulary which is generally very large and often consists of several tens of thousands of words. However, the great dimension of these data lets the majority of classification algorithms difficult to apply, in addition, the representation of textual data is typically hollow [32].

2.2.2 Representation with sentences

Some researchers propose the use of sentences as units to represent texts instead of words as it was seen with the “bag of words” representation, because sentences are more informative than words only, for example sentences like: “Search for Information”, “World Wide Web” have a smaller degree of ambiguity than the separated words. In addition, the sentences have the advantage of preserving the information relative to the position of words in the text [7, 10].

2.2.3 Representation with lexical roots (Stems)

In the representation “Bag of Words” each inflection (derived word) is regarded as a different term; in particular the various forms of a verb are considered as different words (cross, crosses, crossed) although they are inflected forms of the same verb. So, that can increase the dimension of the vector space representing different texts. To deal with this problem, it is necessary to consider only the roots of words (stems) than the whole words. Several algorithms were proposed to substitute the words by their roots; the most known for English is the Porter algorithm [20].

2.2.4 Representation with lemmas

The lemmatization consists in using the grammatical analysis in order to replace verbs by their infinitive forms and nouns by their forms in singular. The lemmatization is thus more complicated to implement than the search of roots because it requires a grammatical analysis of texts. An effective algorithm named “TreeTagger” was developed for many languages: English, German, and Italian. This algorithm uses the decision trees to carry out the grammatical analysis with files of parameters specific to each language [17].

2.2.5 Conceptual representation

Is based on the vector formalism, but the elements of the vector are not directly associated any more to index terms but rather with concepts. The idea is to gather the synonymous words and to associate an under-adjacent lexical concept to them which requires the construction of a lexical base for each language (e.g., Wordnet ontology). For example we associate with the synonyms (summit, top, peak) the concept “peak”. The advantage of the conceptual representation is to reduce the representation vector space by gathering the synonymous words and
attributing to them a common concept. Contrary to the representation “Bag of Words” which associates to each word a dimension in the vector. However, the major disadvantage of the conceptual representation is the absence of lexical bases for all languages which allow such representations [24].

2.2.6 Representation Based on N-grams

An N-gram of X is defined as a sequence of N consecutive X. X can be a character or a word [3]. An N-gram of character is thus a consecutive sequence of N characters [8, 28] which cannot be ordered (e.g., the 3-grams of the sentence “Hello Sir” are: “Hel”, “ell”, “llo”, “lo”, “o”, “Si”, “Sir” [5, 13, 18]). The N-gram profile of a document consists of the list of the most frequent N-grams in the reverse order of their frequencies. The approach of text segmentation into characteristic N-grams has several advantages, in particular:

- Tolerant to spelling, typing, and OCR mistakes.
- This approach is language independent.
- Avoid the use of lemmatization and stemming on the text which requires an algorithmic and linguistic effort.
- Segmentation into words is difficult for some languages e.g., in Arabic the names and additional subjects are in some cases attached to verbs and the string is thus a sentence like:

  (katabtouhou) (I wrote it).

In our work we used the two representations: “bag of words” and “N-grams of characters”.

2.3 Text categorization methods

2.3.1 Conventional methods

Several methods exist in the field of text categorization. Their common difficulty is the very large dimension. Among these methods, we can note: decision trees (ID3, C4.5, CART, . . . ) [23], neural networks with back propagation, SVM and RBF methods [9, 15].

2.3.2 Nearest Neighbors methods

Many text categorization algorithms are based on the concept of distance (similarity). The principal idea is to find the text of the training set, which is nearest in distance to the new text to classify, and to assign its category to the new text. We can also increase the number k of texts which are nearest to the new text if that is necessary. In this case, the category of the new text is the same as the majority of their k nearest neighbors (the majority category). The challenge for these methods is how to define a metric of similarity. Practically, there exist several distances, the most used are: The dot product distance (Inner product), Euclidean distance, Cosine distance, Manhattan, Dice, Jaccard, and others.
2.4 Similarity metrics used in language identification

The majority of the learning algorithms used in the field of language identification are based on the concept of distance or similarity metric. The choice of a distance is a common difficulty for these algorithms. Practically there exist several pseudo-distances, in particular: the distance of Beesley, the distance of Cavnar and Trenkle, the distance of Kullback-Leibler, the distance of Khi-2 \( \chi^2 \), etc.

2.4.1 Distance of Beesley

In the method of Beesley [2], the identification consists of two phases: the learning phase that consists in segmenting texts of each language \( L \) into words, then segmenting each word into bi-grams without repeating the letter more than once, building bi-grams profile for each language and use it as a reference profile, then calculate the frequency or the probability of occurrence of each bi-gram in this profile. The diagnostic phase that consists in establishing the bi-grams profile of the new text \( T \), then find the nearest reference profile using the distance with the profile of the language \( L \) by calculating the product of bi-grams probabilities of the new text \( T \) found in the profile of the language \( L \) (Naive Bayes algorithm). This method assumes the possibility of segmenting texts into words which is not the case for some other languages. In addition, it is based on bi-grams, then it is necessary to not neglect working with trigrams or quad-grams to maintain the specificity of each language. For example, the 4-gram “tion” characterizes French and English, if you segment into bi-grams as follows: “ti” and “on”, the system finds some difficulties to differentiate with “ti” and “on” of Spanish and Portuguese [6,13,29,33].

2.4.2 Distance of Cavnar and Trenkle

The method of Cavnar and Trenkle [8] consists of two phases: the acquisition phase which consists in establishing a tri-gram profile for each language \( L \), then use it as a reference profile. The diagnostic phase which involves building a tri-gram profile for the new text \( T \), then calculate distances between this profile and reference profiles of the different languages. The distance to be calculated is based on the sum of position errors (difference in ranks or “out-of-place” measurement) between each tri-gram in the new text profile \( P_T \) and the same tri-gram in the reference profile of each language \( P_L \) if the tri-gram is present. Otherwise, it takes a maximum value of rank. The language of the new text is that for which the distance is minimal.

Formally, the distance between the new text profile \( P_T \) and the language profile \( P_L \) is calculated as follows:

\[
CT (P_T, P_L) = \min \left\{ \frac{\sum_{g \in P_T} | \pi_{P_L}(g) - \pi_{P_T}(g) |}{D_{\max}} \right\}
\]

if \( g \) is present,

\[
D_{\max} \text{ if } g \text{ is absent.}
\]

(1)

Here, \( g \) is a tri-gram, \( P_T \) profile of the new text \( T \), \( P_L \) profile of the language \( L \), \( \pi_{P_T}(g) \), \( \pi_{P_L}(g) \) positions of the tri-gram \( g \) in the profiles \( P_T, P_L \) if \( g \) belongs to the language profile.
2.4.3 Distance of Kullback-Leibler (KL)

Is based on the relative entropy of Kullback and Leibler [29] as distance measurement. A relative entropy between two probability distributions is the additional amount of information needed to code the second distribution using a code generated by the first [13]. Formally this distance measurement is calculated by the following equation:

$$\text{KL}(T_1, T_2) = \sum_g f_2(g) \log \left( \frac{f_2(g)}{f_1(g)} \right),$$

(2)

where $T_1, T_2$ are texts, $f_1(g), f_2(g)$ frequencies of $N$-grams $g$ in texts $T_1, T_2$ successively. If the $N$-gram $g$ is absent from the text $T_i$ a half-frequency is added to prevent the score from falling to $-w$.

2.4.4 Distance of Khi-2 ($\chi^2$)

Presented by [4, 29] and re-used by [22]. This distance is characterized by the distributional equivalence property indicated by [4]. I.e., if two texts have the same $N$-gram profile, we can merge them into a single text without changing the distance, neither between texts nor between $N$-grams. This distance is formally presented as follows:

$$\chi^2(T_1, T_2) = \sum_g \frac{(f_1(g) - f_2(g))^2}{f_2(g)}$$

(3)

with $f_1(g), f_2(g)$ frequencies of $N$-gram $g$ in texts $T_1, T_2$

$$f_i(g) = \frac{\text{Nb. Occurrences of } g \text{ in } T_i}{\text{Total tri-grams in } T_i}. \quad (4)$$

3. Our contribution

3.1 Application of language identification metrics in text categorization

Research on text categorization often uses the similarity measurements mentioned in Section 2.3.2. In our work we used a new panoply of distances which are used particularly in the field of language identification. The experimental results obtained in this field show the effectiveness of this projection in term of rate success and simplicity of implementation.

3.2 Proposition of new method

Our method is inspired from the method of [8] with the following differences and improvements:

- [8] have applied their method to identify the language of a text. In our work, we applied it in contextual categorization of texts (topical TC).
– [8] require sorting profiles of different languages (categories in our case), as well as the profile of the new text to classify according to the reverse order of frequencies before any calculation, this is not necessary in our method, which permits to save a considerable time required by sorting.

– [8] works only with trigrams \((n = 3)\) contrary to our method which is more general \((n = 3, 4, 5)\).

– The calculation of distance used in [8] is based on the sum of position errors between the new text trigram profile and the same trigram in the reference profile of each language if the trigram is present. If this is not the case, the distance takes a maximum value of position error. Here, we can note two disadvantages: The first is the calculation of the sum of the position error requires a huge computational effort, especially when we use a corpus of a large size. The second problem is in the choice of the maximum position error when the trigram is absent. Here, no method was specified. Thus to overcome the two disadvantages we propose the following method: we take each \(N\)-gram \((n = 3, 4, 5)\) of the new text profile, and we look in the profile of each language. If this \(N\)-gram exists, we assign the value 1 to it, otherwise we assign the sum of the frequencies of all \(N\)-grams in the corpus, and then we calculate the sum which represents the distance. The text will be assigned to the language whose distance is minimal.

\[
\text{Category profile} \quad \text{New text profile} \quad \text{Index of } N \text{gram}
\]

\[
\begin{array}{ccc}
\text{TH} & \text{TH} & 1 \\
\text{ING} & \text{ER} & 1 \\
\text{ON} & \text{ON} & 1 \\
\text{ER} & \text{LE} & \text{Sum freq} \\
\text{AND} & \text{ING} & 1 \\
\text{ED} & \text{AND} & 1 \\
\text{...} & \text{...} & \text{Sum distance measure}
\end{array}
\]

**Fig. 1** Principle of the proposed method [8].

Formally our distance measure is calculated as follows:

\[
CT(P_T, P_L) = \min \left\{ \sum_{i=1}^{\|P_L\|} a_i, \quad a_i = 1 \text{ if } g \text{ is present, } \sum \text{freq } \text{if } g \text{ is absent.} \right\}
\]

\[(5)\]
4. Experiments

4.1 Training and test corpus

We used in our experiments the Reuters\textsuperscript{21578} corpus whose documents are issued of the dispatches of international newspapers [16]. The original corpus contains 116 categories (93 for another version of the same corpus) for which the first ten categories are bulkiest [1, 11, 14, 19, 21, 30]. For this purpose, and in order to facilitate the implementation of learning algorithms and to reduce the computing times, we used only the first 10 categories (\textsc{Acq, Corn, Crude, Earn, Grain, Interest, Moneyfx, Ship, Trade, Wheat}), with a total of 7193 documents for the training set and 2747 documents for the test set. Another correction on the corpus which was also necessary is the elimination of documents having big sizes. We have to leave only reasonable size documents (1–3ko). Fortunately, the number of these documents is not important (a few tens) and thus does not influence the obtained results. The training and test corpora are summarized in Tab. I.

\begin{table}[h]
\centering
\begin{tabular}{lcc}
\hline
Categories & Training Corpus & Test Corpus \\
\hline
Acq & 1650 & 719 \\
Corn & 181 & 53 \\
Crude & 389 & 187 \\
Earn & 2877 & 1086 \\
Grain & 433 & 146 \\
Interest & 347 & 121 \\
Moneyfx & 538 & 166 \\
Ship & 197 & 88 \\
Trade & 369 & 112 \\
Wheat & 212 & 69 \\
Total & 7193 (72 \%) & 2747 (28 \%) \\
\hline
\end{tabular}
\caption{Training and test corpus used in experiments.}
\end{table}

4.2 Pre-processing performed on training and test corpus

Before proceeding to the phase of categorization itself, another phase of pre-processing on training and test corpora is very important. It comprises the following tasks:

- Elimination of unnecessary characters (punctuation, digits, abbreviations, etc).
- Conversion of uppercase to lowercase.
- Morphosyntactic pre-processing on the text (text standardization).
- Segmentation of texts into words and \textit{N}-grams ($n = 3, 4, 5$).
– Setting a minimum frequency threshold \( s \) and eliminate the \( N \)-grams whose frequencies are less than \( s \).

### 4.3 Performed processing

After finishing the phase of pre-processing on the training and test corpus, we proceed to the following treatments:

– For the segmentation, we used two approaches: Bag of words, \( N \)-grams of characters.

– The results were tested for varying thresholds \( (s = 2, 3, 4) \).

– For text segmentation into \( N \)-grams we tested results for varying values of \( n \) \((n = 3, 4, 5)\).

– For the applied learning algorithm, we chose the algorithm of \( k \)-NN with \( k \), SVM with RBF and sigmoid kernels (LIBSVM), as well as NB algorithm.

– We applied the 1-NN algorithm with a variety of distances such as: CT \([8]\), KL \([13]\), Khi-2 \((\chi^2)\) \([13]\), then our suggested method with the new associated distance.

### 5. Evaluation of the obtained results

After the automatic pre-processing performed on both training and test corpus as it is indicated in Section 4.2, we conducted the phase of texts’ segmentation into basic units, we have used for this purpose, words, 3-grams, 4-grams, and 5-grams. The Tabs. II, III, IV, and V summarize the obtained results in the segmentation phase. We note here that learning is applied to the most frequent tokens as it is shown in Tabs. II, III, IV, V (column 5) because of their small number and because they always give the best success rate in text categorization. For learning, we have also applied NB algorithm which is the best known algorithm

<table>
<thead>
<tr>
<th>Categories</th>
<th>#Texts</th>
<th>#gross words</th>
<th>#purified words</th>
<th>#Most freq.words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acq</td>
<td>1650</td>
<td>131214</td>
<td>15321</td>
<td>5998</td>
</tr>
<tr>
<td>Corn</td>
<td>181</td>
<td>20854</td>
<td>4077</td>
<td>1406</td>
</tr>
<tr>
<td>Crude</td>
<td>389</td>
<td>53214</td>
<td>7833</td>
<td>3147</td>
</tr>
<tr>
<td>Earn</td>
<td>2877</td>
<td>159900</td>
<td>14091</td>
<td>4545</td>
</tr>
<tr>
<td>Grain</td>
<td>433</td>
<td>47166</td>
<td>6945</td>
<td>2761</td>
</tr>
<tr>
<td>Interest</td>
<td>347</td>
<td>33441</td>
<td>4842</td>
<td>1947</td>
</tr>
<tr>
<td>Moneyfx</td>
<td>538</td>
<td>60585</td>
<td>6947</td>
<td>3032</td>
</tr>
<tr>
<td>Ship</td>
<td>197</td>
<td>21337</td>
<td>5123</td>
<td>1763</td>
</tr>
<tr>
<td>Trade</td>
<td>369</td>
<td>56513</td>
<td>7250</td>
<td>3058</td>
</tr>
<tr>
<td>Wheat</td>
<td>212</td>
<td>22738</td>
<td>4395</td>
<td>1575</td>
</tr>
</tbody>
</table>

**Tab. II** Segmentation of the training corpus into words \((S = 3)\).
<table>
<thead>
<tr>
<th>Categories</th>
<th>#Texts</th>
<th>#3g gross</th>
<th>#3g purified</th>
<th>#3g Most.Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acq</td>
<td>1650</td>
<td>927168</td>
<td>20034</td>
<td>12635</td>
</tr>
<tr>
<td>Corn</td>
<td>181</td>
<td>118180</td>
<td>8920</td>
<td>4731</td>
</tr>
<tr>
<td>Crude</td>
<td>389</td>
<td>264286</td>
<td>11976</td>
<td>7177</td>
</tr>
<tr>
<td>Earn</td>
<td>2877</td>
<td>1180274</td>
<td>20653</td>
<td>13186</td>
</tr>
<tr>
<td>Grain</td>
<td>433</td>
<td>282329</td>
<td>12205</td>
<td>7194</td>
</tr>
<tr>
<td>Interest</td>
<td>347</td>
<td>180078</td>
<td>8817</td>
<td>5213</td>
</tr>
<tr>
<td>Moneyfx</td>
<td>538</td>
<td>310976</td>
<td>10411</td>
<td>6542</td>
</tr>
<tr>
<td>Ship</td>
<td>197</td>
<td>134944</td>
<td>10025</td>
<td>5441</td>
</tr>
<tr>
<td>Trade</td>
<td>369</td>
<td>260005</td>
<td>10249</td>
<td>6400</td>
</tr>
<tr>
<td>Wheat</td>
<td>212</td>
<td>137411</td>
<td>9285</td>
<td>5051</td>
</tr>
</tbody>
</table>

Tab. III Segmentation of the training corpus into $N$-grams ($N = 3$, $S = 3$).

<table>
<thead>
<tr>
<th>Categories</th>
<th>#Texts</th>
<th>#4g gross</th>
<th>#4g purified</th>
<th>#4g Most.Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acq</td>
<td>1650</td>
<td>844884</td>
<td>60343</td>
<td>28259</td>
</tr>
<tr>
<td>Corn</td>
<td>181</td>
<td>111600</td>
<td>19936</td>
<td>7357</td>
</tr>
<tr>
<td>Crude</td>
<td>389</td>
<td>252769</td>
<td>30632</td>
<td>13456</td>
</tr>
<tr>
<td>Earn</td>
<td>2877</td>
<td>1009809</td>
<td>63970</td>
<td>30989</td>
</tr>
<tr>
<td>Grain</td>
<td>433</td>
<td>264300</td>
<td>31274</td>
<td>13240</td>
</tr>
<tr>
<td>Interest</td>
<td>347</td>
<td>165644</td>
<td>20120</td>
<td>8801</td>
</tr>
<tr>
<td>Moneyfx</td>
<td>538</td>
<td>290819</td>
<td>25862</td>
<td>12213</td>
</tr>
<tr>
<td>Ship</td>
<td>197</td>
<td>126412</td>
<td>25969</td>
<td>8990</td>
</tr>
<tr>
<td>Trade</td>
<td>369</td>
<td>251595</td>
<td>23069</td>
<td>12191</td>
</tr>
<tr>
<td>Wheat</td>
<td>212</td>
<td>127626</td>
<td>21052</td>
<td>8032</td>
</tr>
</tbody>
</table>

Tab. IV Segmentation of the training corpus into $N$-grams ($N = 4$, $S = 3$).

<table>
<thead>
<tr>
<th>Categories</th>
<th>#Texts</th>
<th>#5g gross</th>
<th>#5g purified</th>
<th>#5g Most.Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acq</td>
<td>1650</td>
<td>834098</td>
<td>122143</td>
<td>41571</td>
</tr>
<tr>
<td>Corn</td>
<td>181</td>
<td>110615</td>
<td>33084</td>
<td>8544</td>
</tr>
<tr>
<td>Crude</td>
<td>389</td>
<td>251161</td>
<td>55801</td>
<td>18036</td>
</tr>
<tr>
<td>Earn</td>
<td>2877</td>
<td>987806</td>
<td>129798</td>
<td>42163</td>
</tr>
<tr>
<td>Grain</td>
<td>433</td>
<td>261712</td>
<td>57414</td>
<td>17459</td>
</tr>
<tr>
<td>Interest</td>
<td>347</td>
<td>163864</td>
<td>34901</td>
<td>11303</td>
</tr>
<tr>
<td>Moneyfx</td>
<td>538</td>
<td>288262</td>
<td>47448</td>
<td>17233</td>
</tr>
<tr>
<td>Ship</td>
<td>197</td>
<td>125227</td>
<td>38675</td>
<td>10604</td>
</tr>
<tr>
<td>Trade</td>
<td>369</td>
<td>250443</td>
<td>48131</td>
<td>160855</td>
</tr>
<tr>
<td>Wheat</td>
<td>212</td>
<td>126360</td>
<td>35281</td>
<td>9349</td>
</tr>
</tbody>
</table>

Tab. V Segmentation of the training corpus into $N$-grams ($N = 5$, $S = 3$).
in the field, and because it gives good results, SVM with RBF and sigmoid kernels (LIBSVM) plus a 1-NN algorithm referring to several pseudo-distances. And finally, we have applied our new method with its own pseudo-distance. The obtained results show that our method always gives better results compared to the other methods (Suc\_Rate > 70\%) as it is shown in Tabs. VI, VII and Figs. 1, 2, 3, 5.

Tab. VI Success rate, error rate for all learning algorithms (Approach bag of words).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Suc_Rate</th>
<th>Err_Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>N.Bayes</td>
<td>83.76</td>
<td>16.24</td>
</tr>
<tr>
<td>1NN with CT</td>
<td>82.92</td>
<td>17.08</td>
</tr>
<tr>
<td>1NN with KL</td>
<td>83.87</td>
<td>16.13</td>
</tr>
<tr>
<td>1NN with $\chi^2$</td>
<td>82.81</td>
<td>17.19</td>
</tr>
<tr>
<td>SVM (RBF)</td>
<td>55.36</td>
<td>44.64</td>
</tr>
<tr>
<td>SVM (SIG)</td>
<td>47.57</td>
<td>52.43</td>
</tr>
<tr>
<td>The new method</td>
<td>87.57</td>
<td>12.43</td>
</tr>
</tbody>
</table>

Tab. VII Success rate, error rate for all learning algorithms (Approach N-grams of characters).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>N = 3</th>
<th>N = 4</th>
<th>N = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Suc_Rate</td>
<td>Err_Rate</td>
<td>Suc_Rate</td>
</tr>
<tr>
<td>N.Bayes</td>
<td>80.27</td>
<td>19.73</td>
<td>84.91</td>
</tr>
<tr>
<td>1NN with CT</td>
<td>66.12</td>
<td>33.88</td>
<td>72.05</td>
</tr>
<tr>
<td>1NN with KL</td>
<td>67.23</td>
<td>32.77</td>
<td>74.19</td>
</tr>
<tr>
<td>1NN with $\chi^2$</td>
<td>68.27</td>
<td>31.73</td>
<td>77.69</td>
</tr>
<tr>
<td>SVM (RBF)</td>
<td>48.34</td>
<td>51.66</td>
<td>55.84</td>
</tr>
<tr>
<td>SVM (SIG)</td>
<td>52.74</td>
<td>47.26</td>
<td>55.4</td>
</tr>
<tr>
<td>The new method</td>
<td>73.45</td>
<td>26.55</td>
<td>82.46</td>
</tr>
</tbody>
</table>

Fig. 2 Suc-rate (App.bag of words).
Fig. 3 Suc-rate (App. N-grams, N=3).

Fig. 4 Suc-rate (App. N-grams, N=4).

Fig. 5 Suc-rate (App. N-grams, N=5).
For segmentation into 5-grams, the success rate in categorization is 88% (Tab. VII, columns 4, 5). Considering the sizes of corpora used in experiments, in addition to the implementations that we made for some learning algorithms, we believe that the results are very significant and can be improved in other future works.

6. Conclusion and perspectives

In this paper we treated the problem of text categorization in a heterogeneous textual corpus. We explained the two most known approaches in the field to segment a text into basic units, including: “bag of words” and “N-grams” approaches. The realized implementations show the limitation of the “bag of words” approach compared to the “N-grams” approach which is more general, independent of languages and always gives the best results. For the thematic categorization, we implemented several algorithms based on different pseudo-distances, namely: CT, KL, KHI2. In the light of the obtained results, we proposed a new method equipped with its own distance. This new method is inspired from the CT method with the advantage that this last one does not require sorting, and it’s based on a simple and less expensive distance especially for corpus with large sizes. The obtained results by applying our new method are very significant, in term of computation time and in accuracy when categorizing texts. Our perspective is to apply the new method on a corpus of semi-structured documents, and generalize it in other tasks of text categorization like: text summarizing, author recognizing, contextual categorization and author recognizing in the same time.

References


[24] RÉHEL S. Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés. Laval, Canada, 2005. Université Laval Faculté des Sciences et de Génie. [In French].


Gadri S., Moussaoui A.: application of a new set of pseudo-distances in...