International Journal of Computer Science and Applications,

Vol. 7, No. 1, pp. 94 – 107, 2010

AUTOMATIC LANGUAGE IDENTIFICATION: AN ALTERNATIVE UNSUPERVISED APPROACH USING A NEW HYBRID ALGORITHM

ABDELMALEK AMINE

UTMS University of Saida, Algeria
Saida, Algeria
EEDIS Laboratory, UDL University
Sidi Belabbes, Algeria
amine_abd1@univ-sab.dz

ZAKARIA ELBERRICHI

EEDIS Laboratory, UDL University
Sidi Belabbes, Algeria
elberrichi@univ-sab.dz

MICHEL SIMONET

TIMC-IMAG, IN3S Laboratory
Grenoble, France
michel.simonet@imag.fr

This paper deals with our research on unsupervised classification for automatic language identification purpose. The study of this new hybrid algorithm shows that the combination of the Kmeans and the artificial ants and taking advantage of an n-gram text representation is promising. We propose an alternative approach to the standard use of both algorithms. A multilingual text corpus is used to assess this approach. Taking into account that this method does not require a priori information (number of classes, initial partition), is able to quickly process large amount of data, and that the results can also be visualised. We can say that, these results are very promising and offer many perspectives.

Keywords: Language identification; clustering; multilingual text; AntClass.

1. Introduction

Research in recent years has given a lot of interest to textual data processing and especially to multilingual textual data. This is for several reasons: a growing collection of networked and universally distributed data, the development of communication infrastructure and the Internet, the increase in the number of people connected to the global network and whose mother tongue is not English [1]. This has created a need to organize and process huge volumes of data. The manual processing of these data (expert , or knowledge based systems) is very costly in time and personnel, they are inflexible and generalization to other areas are virtually impossible, so we try to develop automatic methods [2].

One of the major issues raised in any application of automatic processing of digital documents is that of multilingualism, since we want to perform linguistic processing. Any linguistic text processing is completely dependent on the language of the latter. It is therefore essential in a multilingual environment that research tools are able to automatically identify the languages of the documents they have to deal with. A variety of methods for identifying text language of a multilingual corpus have been developed [3].

We propose in this paper to study the effectiveness of the clustering algorithm AntClass [4] for identifying text language of a given multilingual corpus, based on a vector representation focused not on words but on the n-grams for representing the texts.

Section 2 introduce and explain the automatic language identification. Section 3 is devoted to our methodological approach and its stages, and in section 4 we present, evaluate and discuss the obtained results. Finally section 5 will conclude the article.

2. Automatic language identification

Language identification is to assign a textual unit, supposedly monolingual to a language. This identification became important as textual data in different languages, are more and more available on the global network [3] [5].

Automatic language identification is possible because natural languages are extremely non-random, and they each have regularities in the use of characters or character sequences. The alphabet of each language is either unique or highly characteristic of this language. Information on the stability and consistency of the frequency of letters and letter sequences are not new [6]. It is statistically proven that for each language, the number of occurrences of the sequence of two, three, four or five letters are stable and different from language to language. For example, in English, in any text, the frequency of the letter "E" is about 13%, the frequency of the letter "U" is about 3% and the frequency of the letter "Z" is approximately 0.1%. For two sequences of characters or bi-grams, we find for example that the probability of having the string "TH" in English is relatively high, in Spanish and Portuguese, this probability approaches zero. In the same order, the probability of having "SZ" in Hungarian and Polish is great; the string "TION" characterizes the French and English. Based on these probabilities of occurrence of letters and letter sequences, we can design an algorithm capable of identifying the language of a text.

We can distinguish two kinds of approaches: linguistic approaches, and probabilistic and statistical approaches.

The approaches based on linguistic knowledge involve the construction of linguistic resources and require prior knowledge. They are not generalizable to the classification of languages in text categorization.

The statistical approaches use probabilities and knowledge built automatically from a text corpus representative of the language, the goal is to capture using statistical models and probabilities certain regularities of the languages and their associated frequency or probability of occurrence. They generalize the recognition of language classification of

texts. These empirical regularities play the role of linguistic knowledge. The identification is to calculate the probability for a statement to belong to different languages, according to the regularities observed.

A variety of tools have been developed to classify texts based on their respective languages [3] [7] [8] [9]. However, all these approaches work in a supervised manner: given a sample of each language model, parameters are estimated for prediction and texts are classified according to their similarity with the learning text sets. But supervised learning has a major drawback: The languages that are not contained in the training set will not be identified and the text will be assigned to other classes arbitrarily.

We propose in this work, a method that operates on the n-grams of characters as attributes, and clusters together similar texts and discovers the number of languages in a completely unsupervised manner.

3. Methodological Approach

In this section we describe our methodology. With the use of the approach based on the n-grams we construct matrix documents-terms that will be exploited by the AntClass algorithm to group similar documents together. This combination will be examined in several experiments using the Euclidean distance, cosine distance and Manhattan distance as similarity measures for several values of n.

3.1. *Corpus*

Sub Experiments conducted in our work are based on a multilingual corpus composed from texts from different sources, we had to adapt:

- For the French language: the corpus DUP[*], 159 texts in HTML format.
- For the Arabic language: the corpus CCA[†], 415 texts in XML format.
- For the English language: the corpus EDM[‡], 404 texts in HTML and from the Reuters corpus[§] 163 articles in SGML format.

The texts of our corpus come in several formats (HTML, XML and SGML). For each item of the corpus, we removed all tags like: <title> ... </ title> <auteur> ... </ author> <date> ... </ date> ... etc... We took only the text part (written by the author).

In a first step, we transformed the uppercase characters to lowercase characters for English and French, then we have automatically eliminated from the text diacritical characters (punctuation) such as: dot, comma, semicolon, the question mark and exclamation etc.... and the numbers because these characters have no influence on the results of the clustering and do not provide relevant information for the decision making, their elimination reduces the size of the representation space. The corpus texts are saved in UTF-8 encoding. This allows us to handle documents that use different character sets.

[*] Duplessis Project (Dup) (Research Group analysis of political discourse) University of Quebec at Montreal. http://www.chaire-mcd.uqam.ca/ato-mcd/projet_dup.html
[†] Corpus of Contemporary Arabic project, (CCA). http://www.comp.leeds.ac.uk/eric/latifa/research.htm
[‡] World Deliberative Area (EDM). http://www.chaire-mcd.uqam.ca/ato-mcd/projet_edm.html
[§] Reuters-21578 Corpus of English-language news proposed by the Reuters agency

3.2. *A representation based on n-grams of characters*

The term "n-gram" was introduced by [10] in 1948. Since then, the n-grams have been used in several areas, such as speech recognition systems, with typical values of n equal to 3 or 4. They are now also used in systems for automatic processing of language for information retrieval. One of the applications of the n-grams model is the indexing of large corpus [11].

An n-gram may designate both an n-tuple of characters (n-gram character) or an n-tuple of words (n-gram words). This model does not represent documents by a vector of term's frequencies, but by a vector of n-gram's frequencies in the documents.

An n-gram character is a sequence of *n* consecutive characters. For any document, all n-grams that can be generated are the result obtained by moving a window of *n* boxes in the text [12] [13]. This movement is made in stages; one stage corresponds to one character for n-grams of characters, and a word for n-grams of words. Then we count the frequencies of n-grams found. In scientific literature, this term sometimes refers to sequences that are neither ordered nor straight, for example a bigram can be composed of the first letter and third letter of a word; [14] consider an n-gram as a set of unordered *n* words after performing the stemming and the removing of Stopwords.

Techniques based on n-grams have several advantages: they automatically capture the roots of the most frequent words [15] and operate independently of languages [8] and are tolerant of spelling errors and distortions caused when using optical scanners [16] and do not need the removing of Stopwords or the stemming process [17] that improve the performance of words based systems.

In our experiments, n-grams of characters are used, thus an n-gram refers to a string of *n* consecutive characters.

In this approach, we do not need to conduct a linguistic processing of the corpus. For a given document, as we already said, extracting all n-grams (usually $n = (2, 3, 4, 5)$ is the result obtained by moving a window of *n* boxes in the main text. This movement is made by steps of one character at a time, every step we take a "snapshot" and all these 'shots' constitute the set of all n-grams of document.

We cut the texts of the corpus based on the value of *n* chosen. We took $n = 2, 3, 4$ and *5*.

For example, the 5-grams characters of the following text (text from Reuters-21578):

> sandoz ag said it planned a joint venture to produce herbicides in the soviet union the company said it had signed a letter of intent with the soviet ministry of fertiliser production to form the first foreign joint venture the ministry had undertaken since the soviet union allowed western firms to enter into joint ventures two months ago the ministry and sandoz will each have a stake but a company spokeswoman was unable to give details of the size of investment or planned output

are:
"Sando, andoz, ndoz_, doz_a, oz_ag, z_ag, _ag_s, ag_sa, g_sai, _said, said_, aid_i, id_it, d_it_, _it_p, it_pl, t_pla, _sitemap ..., ned_o, ed_ou, d_out, _outp, Outpu, utput"

The character _ represents a space.

We constitute in this way the cross table N_{ij} of occurrences of the n-gram *i* in text *j* so that all the n-gram do not contain spaces and belong absolutely to an index of Arabic, English, French, Spanish and Italian , predefined in advance.

Algorithm n-gram

```
(1) for each text do
(2)  for each n-gram do
(3)    if the n-gram contains a "" (Space) then remove the n-gram
(4)    else if the n-gram belongs to the index then check for an entry in the
            global vector of the n-grams, which corresponds to this n-gram,
             increment the box N_ij where i is the rank of  the n-gram in the
             global vector and j is the  text number
(5)         else create a new entry corresponding to this n-gram in the global
             vector and affect 1 to N_ij where i corresponds to the last n-gram
             and j to the text number
(6)        endif
(7)    endif
(8)  endfor
(9) endfor
```

- Dimension reduction

The objective of reduction methods of terms is to provide a shorter but more meaningful list of terms. The terms are usually ordered from the most important to least important according to some criterion. The question arises in the number of terms to retain in the list [18]. To choose the right number of words, you must know whether the information conveyed by the words at the end of the list is useful, or it is redundant with information provided by the terms of the beginning of the list. There is no evidence that a large number of terms is necessary for good performance, because even with models like Support Vector Machines (SVM) which are in principle suitable for large vectors, the results are contradictory. This is probably due to the fact that the terms are mutually correlated, and to the way different algorithms manage these relationships.

We know that reducing the size by using the frequency-document is immediate, and that its performance is equivalent to other more sophisticated forms despite its simplicity [19]. It eliminates the n-grams that appear in a number of documents below a certain threshold. We chose to eliminate the n-grams that appear in only one document (the chosen threshold is 1), greatly reducing the number of n-grams.

At the end of these steps, we obtain a document-term matrix N_{ij} and an overall n-grams vector (n = 2, 3, 4 and 5).

To calculate the weight (frequency) of each extracted n-gram, we use a combination of local and global weights [20],

$$tf \times idf\left(t_k, d_j\right) = Occ(t_k, d_j) \times Log \frac{Nbre\_doc}{Nbre\_doc(t_k)} \quad (1)$$

which is normalized to correct the influence of the lengths of the texts [20]:

$$w(t_k, d_j) = \frac{tf \times idf\left(t_k, d_j\right)}{\sqrt{\sum_{k=1}^{n}\left(tf \times idf\left(t_k, d_j\right)\right)^2}} = \frac{Occ(t_k, d_j) \times Log \frac{Nbr\_doc}{Nbr\_doc(t_k)}}{\sqrt{\sum_{k=1}^{n}\left(tf \times idf\left(t_k, d_j\right)\right)^2}} \quad (2)$$

where:
-The term t_k is the k^{th} n-gram of document d_j,
-N is the total number of n-grams extracted,
-$Occ(t_k, d_j)$ is the number of occurrences of t_k in d_j,
- Nbr_doc is the total number of documents from the corpus and $Nbr\_doc(t_k)$ is the number of documents of this set in which t_k appears at least once.

Each document will be represented by its normalized vector of n-gram.

$$d_j = (w_{1j}, w_{2j}, ..., w_{nj}) \quad (3)$$

The weight of the sub-strings obtained (n-grams) are placed in a two-dimensional array (matrix), where columns correspond to documents, while the lines are the weights of the n-grams for each document.

3.3. *Clustering multilingual texts*

Several clustering methods have been applied to textual documents [13]. The Ants which possess a range of behaviors very diverse (collective or individual) suggest very interesting heuristics for many problems including clustering.

An early study on this area was conducted by [21] where a population of ant-agents moves randomly on a two dimensional grid and are able to move objects in order to gather them. This method was extended by [22] on simple objects.

An extension of the algorithm "LF" of [22] was presented in [3] where the authors have developed an algorithm called AntClass using the same principles that LF and adding some improvements. In LF each cell can contain only one object, a class is then represented by a cluster of objects. In AntClass several objects can be placed on a single cell (the ants can pill up objects in the same grid cell), forming a pile. In this case, a class corresponds to a pile and a partition is given by all present piles in the grid. Each pile has a representative which is the center of gravity g_i of the elements that constitute it.

This is a hybrid with the K-Means algorithm. This hybridization consists in initializing the K-Means algorithm with the partition obtained by grouping objects by ants. Thus, this new principle allows for automatic interpretation of classes which is done

visually and with more difficulty in LF. Moreover the AntClass algorithm converges faster, as in LF an ant can pass a number of iterations to find an empty slot next to the group of objects close to that it carries.

The grid G is square and its size is determined automatically based on the number of objects to be treated. If N is the number of objects, G contains L cells per side: $L = \lfloor \sqrt{2N} \rfloor$, this formula ensures that the number of cases is at least equal to the number of objects. Initially the A ants $\{a_1, \ldots, a_A\}$ are arranged randomly on the grid by checking that a cell can only accommodate a single ant and come with a carrying capacity $c(a_i)$, a memory of size $m(a_i)$, velocity $v(a_i)$ and a patience $p(a_i)$, knowing that T is the number of moves of each ant.

Algorithm Ants: grouping objects by ants.
Ants(Grid G)

(1) for t = 1 to T do
(2) for k = 1 to A do
(3) Move the ant a_k on one cell unoccupied by another ant
(4) if there is a lot of objects T_i on the same cell that a_k then
(5) if the ant a_k is carrying an object o_i [a lot of objects T_i] then
(6) place the object o_i [the pile T_i] carried by the ant on the pile T_j following the probability $p_d(o_i, T_j)$ [$p_d(T_i, T_j)$]
(7) else Pick up the object o_i the most dissimilar of the pile T_j [until the capacity $c(a_k)$ of the ant is reached or the pile is empty] by the probability $p_p(T_j)$
(8) endif
(9) endif
(10) endfor
(11) endfor return the grid G

$$p_p(T_j) = \begin{cases} 1 & \text{if} \quad |T_j| = 1 \\ \min\left\{\left(\dfrac{\bar{d}_g(T_j)}{\bar{d}(O)}\right)^{k_1}, \quad 1\right\} & \text{if} \quad |T_j| = 2 \\ 1 - 0.9\left(\dfrac{\bar{d}_g(T_j) + \varepsilon}{d_g^*(T_j) + \varepsilon}\right)^{k_1} & \text{else} \end{cases} \tag{4}$$

where ε is a small positive value ($1^{0\text{-}5}$) | T_j | the number of objects in the pile and k_1 is a positive real parameter to control the shape of the density $p_p(T_j)$ when | T_j |> 2 .

$$p_d(o_i, T_j) = \begin{cases} 1 & \text{if} \quad d(x_i, g_j) \le d_g^*(T_j) \\ 1 - 0.9\min\left\{\left(\dfrac{d(x_i, g_j)}{\bar{d}(O)}\right)^{k_2}, \quad 1\right\} & \text{else} \end{cases} \tag{5}$$

where k2 is a real positive parameter to control the shape of the density pd (oi, Tj) when d(xi, gj)> d*g(Tj).

The similarity between objects is estimated by a function calculating the distance between the vectors of those documents.

- The maximum distance between two objects of the set O:

$$d^*(O) = \max_{(i,j)\in\{1,\ldots,N\}^2} \{d(x_i, x_j\} \tag{6}$$

- The average distance between two objects of the set O:

$$\bar{d}(O) = \frac{2}{N(N-1)} \sum_{(i,j)\in\{1,\ldots,N\}^2} \{d(x_i, x_j\} \tag{7}$$

- The maximum distance between objects in a pile T_j and its center of gravity g_j:

$$d_g^*(T_j) = \max_{x_i \in T_j} \{d(x_i, g_j\} \tag{8}$$

- The average distance between objects in a heap T_j and its center of gravity gj:

$$\bar{d}_g(T_j) = \frac{1}{|T_j|} \sum_{x_i \in T_j} \{d(x_i, g_j\} \tag{9}$$

Algorithm AntClass: unsupervised classification by ants and K-means (The K-means algorithm is initialized with the partition obtained by Ants).
AntClass ()

(1) let P_0 the initial partition consisting of N classes
(2) for $t = 1$ to T_{AntClass} do
(3) initialize the grid G from the partition P_{t-1} (one pile per class)
(4) G '← Ants (G)
(5) construct the partition P' associated to the grid G'
(6) P_t ← K-means (P')
(7) endfor
(8) return the partition $P_{T\text{AntClass}}$

The pair (k_1, k_2) = (0.1, 0.1) provides the lowest number of classes which appears to be the best possible initializing for the K-means.

The *oi* objects are the normalized n-gram vectors for each document. We tested for each value of *n* (*n*= 2, 3, 4 and 5), 3 measures of similarity: cosine distance, Euclidean distance and Manhattan distance. The Table 3 below summarizes the results obtained with 150 ants and 1000 iterations.

- Cosine distance:

$$Cos(d_i, d_j) = \frac{\sum_{t_k} (w_{ki} \bullet w_{kj})}{\|d_i\|^2 \bullet \|d_j\|^2} \tag{10}$$

- Euclidean distance:

$$Euclidean(d_i, d_j) = \sqrt{\sum_{1}^{n} (w_{ki} - w_{kj})^2} \tag{11}$$

- Distance of Manhattan:

$$Manhattan(d_i, d_j) = \sum_{1}^{n} |w_{ki} - w_{kj}| \tag{12}$$

3.4. *Languages Identification*

In this step, we assign a label to each class (after the clustering process) matching the dominant language of each pile of the grid.

We proceeded as follows:

- For each pile, we specify the n-grams that appear at least once in the text of this pile.
- We calculate for each pile the percentages of its component languages as follows:
  - For each gram we traverse the Arabic, English, French, Spanish and Italian index predefined in advance and point the language in which the gram appears by incrementing a corresponding counter;
  - For each language we divide the corresponding counter on the total number of terms of this pile;
  - At the end of this stage, we determine the rate of text for each language present in the pile.
- For each pile, we assign a label named after the dominant language in this pile.
- If we find piles of the same labels, we merge them into a single class, to obtain the minimum number of classes with distinct labels.

We thus obtain three classes of languages: English, French and Arabic, see below Table 4, Table 5 and Table 6.

4. Results and evaluation

In this section we present and evaluate the obtained results.

Table 1 shows the number of n-grams obtained from our corpus for the values 2, 3, 4 and 5 of n.

Table 1 Number of linguistic units (n-grams)

Number of texts in the corpus	**Number of linguistic units (n-grams)**			
1141	2	3	4	5
	1347	6385	13858	15735

Table 2 shows the number of n-grams obtained after reducing the size by using the frequency-document reducing method

Table 2 Number of linguistic units (n-grams) in the corpus after reduction

Initial number of linguistic units (n-grams)				**number of linguistic units (n-grams) after reduction**			
2	3	4	5	2	3	4	5
1347	6385	13858	15735	1078	4186	7310	8601

The first step of clustering process and obtaining the number of piles according to the values of n and each similarity measure (Table 3).

Table 3 Result of clustering for each value of n (n = 2, 3, 4 and 5)

Distance	**Cosine**				**Euclidian**				**Manhattan**			
n-gram	2	3	4	5	2	3	4	5	2	3	4	5
Number of piles	197	107	95	185	119	42	24	109	117	19	17	116

Second step: assigning labels to piles named after the dominant language, merging piles having the same labels, and obtaining three classes of languages, English, Arabic and French (Table 4, Table 5 and Table 6).

Table 4 English Class

Distance	**Cosine**				**Euclidian**				**Manhattan**			
n-gram	2	3	4	5	2	3	4	5	2	3	4	5
Number of piles	127	67	17	15	61	1	2	11	3	1	1	17
Number of texts	410	351	81	40	293	7	31	24	24	16	3	27
Arabic text rate (%)	12.35	14.80	18.35	7.22	6.98	29.41	21.25	3.03	10.55	25.31	0.0	4.90
French text rate (%)	23.55	34.56	23.39	20.92	24.62	26.47	26.87	25.30	13.92	21.51	22.22	21.56
English text rate (%)	50.87	35.34	30.07	47.22	51.96	35.29	31.04	51.96	43.76	31.64	44.44	41.17
Spanish text rate (%)	7.57	5.51	1.61	10	10.21	2.94	11.25	7.87	24.24	8.86	11.11	17.64
Italian text rate (%)	5.65	9.76	11.98	14.62	6.21	5.88	9.58	11.81	7.5	12.65	22.22	14.70

Table 5 Arabic Class

Distance	**Cosine**				**Euclidian**				**Manhattan**			
n-gram	2	3	4	5	2	3	4	5	2	3	4	5
Number of piles	69	39	24	63	57	9	8	23	86	9	2	47
Number of texts	213	233	132	205	331	410	143	46	544	467	6	99
Arabic text rate (%)	53.57	41.05	43.59	50.81	61.53	29.79	36.65	52.89	38.88	42.60	50	53.74
French text rate (%)	16.70	25.39	22.74	27.14	13.73	27.13	28.49	28.26	16.87	22.86	27.77	23.76
English text rate (%)	21.20	20.42	11.06	6.98	18.80	27.80	13.28	5.79	32.55	23.72	11.11	10.08
Spanish text rate (%)	5.01	5.42	1.53	5.80	2.87	6.04	10.25	6.52	6.66	4.20	5.55	5.56
Italian text rate (%)	3.50	7.70	7.22	9.24	3.04	9.21	11.30	6.52	5.02	6.60	5.55	6.83

Table 6 French Class

Distance	**Cosine**				**Euclidian**				**Manhattan**			
n-gram	2	3	4	5	2	3	4	5	2	3	4	5
Number of piles	1	1	54	107	1	32	14	75	28	9	14	52
Number of texts	3	22	372	370	2	213	439	552	62	153	626	501
Arabic text rate (%)	0.0	29	17.95	11.30	0.0	15	19.05	18.13	2.36	18.48	20.30	6.05
French text rate (%)	75	39	41.73	58.47	50	39.36	43.13	50.98	41.46	38.25	37.70	66.56
English text rate (%)	12.5	22	17.29	12.36	30	29.75	18.94	10.87	42.90	27.81	13.64	10.40
Spanish text rate (%)	12.5	4	1.30	8.46	20	5.08	10.83	8.37	5.32	4.98	16.56	8.23
Italian text rate (%)	0.0	6	9.91	9.39	0.0	10.78	8.03	11.63	7.93	10.45	11.77	8.74

We note that percentages are not zero for Spanish and Italian languages; this is explained by the existence of n-grams in English and French texts of our corpus which belong also to the Spanish and Italian index. It is also due to the presence of one or more words of these languages included in English and French texts.

The Arabic texts, sometimes also include one or more Latin words, especially English words. The French texts include English words too.

In our experiments, the clustering results of the different algorithms are evaluated and compared using the F-measure which make use of the known classes for each document. This measure is based on two concepts: recall and precision:

$$Pr\ ecision\ (i,k) = \frac{N_{ik}}{N_k} \tag{13}$$

$$Re\ call\ (i,k) = \frac{N_{ik}}{N_{Ci}} \tag{14}$$

where N is the total number of documents, i is the number of classes (predefined), k is the number of clusters in unsupervised classification, N_{Ci} is the number of documents of class i, N_k is the number of documents of cluster C_k, N_{ik} is the number of documents of class i in the cluster C_k.

F-measure F(P) is calculated as follows:

$$F(P) = \sum \frac{NCi}{N} Max_{k=1}^{K} \frac{(1+\beta) \times Recall(i,k) \times Precision(i,k)}{\beta \times Recall(i,k) + Precision(i,k)} \tag{15}$$

Typically $\beta = 1$.

The partition P - considered as most relevant and which best corresponds to the awaited external solution - is that which maximizes the associated F-measure.

Table 7 gives the values of the F-measure, extraction time and clustering time obtained for each approach.

Table 7 F-measure and running time

Distance	**Cosine**				**Euclidian**				**Manhattan**			
n-gram	2	3	4	5	2	3	4	5	2	3	4	5
Time for extracting n-grams (second)	244	204	204	152	244	204	204	152	244	204	204	152
Clustering time (second)	49	272	403	400	54	130	285	263	28	65	99	61
F-measure (%)	51.71	44.35	47.45	51.81	48.65	44.09	40.63	49.58	44.39	40.28	40.40	46.95

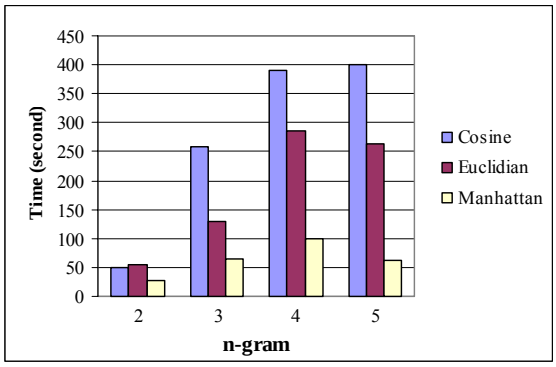

Fig. 1. Clustering time according to n-grams (*n*= 2, 3, 4 and 5) and 3 similarity measurements.

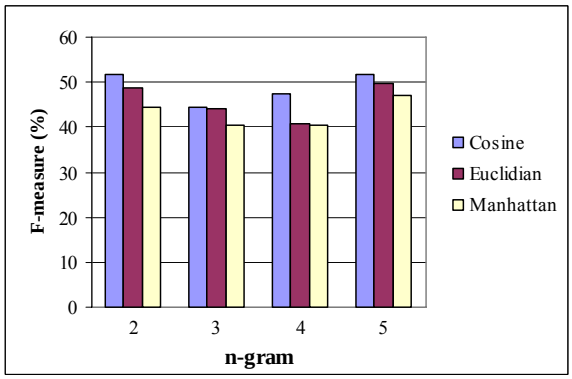

Fig. 2. F-measure according to n-grams (*n*= 2, 3, 4 and 5) and 3 similarity measurements.

We note that the clustering time is smaller when we use the Manhattan distance. We also note that the clustering time increases with the value of n.

The best F-measures and therefore the best partitions are obtained for n = 2 and n = 5. In general the cosine distance produces the best results.

5. Conclusion

The work presented in this paper, shows that it is possible to identify automatically the language in an unsupervised manner and, aims to enhance the unsupervised methods and techniques applied to classification for text language identification of a multilingual corpus.

We presented a method based on the behaviour of real ants having collective and individual characteristics and ability to gather and sort objects. The AntClass algorithm developed on this occasion is hybrid; the search of the number of classes is performed by the artificial ants algorithm and a conventional classification algorithm the K-means, is used to correct the misclassification inherent to stochastic method such as artificial ants. This method is also characterized by the fact that it does not require a priori information

(number of classes, initial partition), and possibly even no parameters, and is able to quickly process a large amount of data. The results provided by our methods can also be visualised. We realized that the choice of a similarity measure is crucial in the process of clustering. Indeed, two different measures can lead to two different results of clustering.

In further work, we will examine how the method performs on other languages. We will investigate the Arabic language which is particularly conducive to the study of dialectal variation. We will also investigate the influence of the number of texts in the corpus.

References

[1] Peters C, Sheridan P. Multilingual Information Access. In M. Agosti, F. Crestani, G. Pasi (eds.). Lectures on Information Retrieval, *Lecture Notes in Computer Science* 1980, pp51-80, Springer Verlag, 2001.

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

[3] Hughes B, Baldwin T, Bird S, Nicholson J, and MacKinlay A. Reconsidering language identification for written language resources. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), 485–488, 2006, Genoa, Italy.

[4] Monmarché N, Slimane M, Venturini G. On improving clustering in numerical databases with artificial ants. In D. Floreano, J.D. Nicoud, et F. Mondala, editors, 5th European Conference on Artificial Life (ECAL'99), *Lecture Notes in Artificial Intelligence,* volume 1674, pages 626–635, Swiss Federal Instituteof Technology, Lausanne, Switzerland, 13-17 September 1999. Springer-Verlag.

[5] Řehûrek R, Kolkus M. Language Identification on the Web: Extending the Dictionary Method. In *Computational Linguistics and Intelligent Text Processing, 10th International Conference, CICLing 2009, Proceedings.* Vyd. první. Mexico City, Mexico: Springer-Verlag, 2009. ISBN 978-3-642-00381-3, pp. 357-368.

[6] Beesley K. Language Identifier: A Computer Program for Automatic Natural Language Identification on On-Line Text. In Proceedings of the 29th Annual Conference of the American Translators Association, 1988, pages 47– 54.

[7] Cavnar W B, Trenkle J M. N-gram-based text categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994, pages 161–175, Las Vegas, US.

[8] Dunning T. Statistical Identification of Languages. Technical Report MCCS, 1994, 94-273, Computing Research Laboratory.

[9] Schütze H, Hull D A, Pedersen J O. A comparison of classifiers and document representations for the routing problem. In Fox, E. A., Ingwersen, P., and Fidel, R., editors, Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval, 1995, pages 229–237, Seattle, US. ACM Press, New York, US.

[10] Shannon C. The Mathematical Theory of Communication. *Bell System Technical Journal*, 1948, 27: 379–423 and 623–656.

[11] Lelu A, Hallab M. Consultation "floue" de grandes listes de formes lexicales simples et composées : un outil préparatoire pour l'analyse de grands corpus textuels. In Rajmann, M. and Chappelier, J. C., editors, *JADT'2000*, 2000, volume 1, pages 317–324, Lausanne.

[12] Rahmoun A, Elberrichi Z. Experimenting N-Grams in Text Categorization. *International Arab Journal of Information Technology*, 2007, Vol 4, N°.4, pp. 377-385.

[13] Amine A, Elberrichi Z, Simonet M, Bellatreche L, Malki M. SOM-Based Clustering of Textual Documents Using WordNet. In: *Handbook of Research on Text and Web Mining Technologies*, Song, M., Wu, YF (eds.). USA : Idea Group Inc., 2008. 189-200, ISBN 978-1-59904-990-8.

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

[15] Grefenstette G. Comparing Two Language Identification Schemes. In Proceedings of the 3rd International Conference on the Statistical Analysis of Textual Data (JADT'95), 1995, Rome, Italy.

[16] Miller E, Shen D, Liu J, Nicholas C. Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System. *Journal of Digital Information*, 1999, 1(5).

[17] Sahami M. Using Machine Learning to Improve Information Access. PhD thesis, 1999, Computer Science Department, Stanford University.

[18] Stricker M. Réseaux de neurones pour le traitement automatique du langage : conception et réalisation de filtres d'information. PhD thesis, 2000, Université Pierre et Marie Curie - Paris VI, Paris.

[19] Yiming Y, Christophere C. A comparative study on feature selection in text categorization. In proc 14th International Conference on Machine Learning, 1997, page 412-420.

[20] Ying L. On Document Representation and Term Weights in Text Classification. In: *Handbook of Research on Text and Web Mining Technologies*, Song, M., Wu, YF (eds.). USA : Idea Group Inc., 2008. 1-22, ISBN 978-1-59904-990-8.

[21] Deneubourg J-L, Goss S, Franks N R, Sendova-Franks A, Detrain C, Chretien L. The dynamics of collective sorting: robot-like ant and ant-like robots. In Proceedings of the First International Conference on Simulation of Adaptive Behavior, 1990, pages 356–365.

[22] Lumer E D, Faieta B. Diversity and adaptation in populations of clustering ants. In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, 1994, pages 501–508.