Journal of Computing and Information Technology - CIT 20, 2012, 2, 125–140
doi:10.2498/cit.1001770

Arabic Text Classification Framework Based on Latent Dirichlet Allocation

Mounir Zrigui[1], Rami Ayadi[2], Mourad Mars[3] and Mohsen Maraoui[1]

[1] University of Monastir, Tunisia
[2] Faculty of Economics and Management, University of Sfax, Tunisia
[3] Stendhal University, Grenoble, France

In this paper, we present a new algorithm based on the LDA (Latent Dirichlet Allocation) and the Support Vector Machine (SVM) used in the classification of Arabic texts.

Current research usually adopts Vector Space Model to represent documents in Text Classification applications. In this way, document is coded as a vector of words; *n*-grams. These features cannot indicate semantic or textual content; it results in huge feature space and semantic loss. The proposed model in this work adopts a "topics" sampled by LDA model as text features. It effectively avoids the above problems. We extracted significant themes (topics) of all texts, each theme is described by a particular distribution of descriptors, then each text is represented on the vectors of these topics. Experiments are conducted using an in-house corpus of Arabic texts. Precision, recall and F-measure are used to quantify categorization effectiveness. The results show that the proposed LDA-SVM algorithm is able to achieve high effectiveness for Arabic text classification task (Macro-averaged F_1 88.1% and Micro-averaged F_1 91.4%).

Keywords: LDA, Arabic, stemming algorithm, text classification, SVM

1. Introduction

With the development of internet, media storage (containing wide textual corpus) and digital encyclopedias, it is becoming difficult or impossible to analyze the huge amount of information. Hence, we need to explore new approaches using automatic text analysis. Collecting and organizing various types of information have created new challenges and new opportunities within the field of computing. When documents are classified by an automated system, people can find required information and knowledge more rapidly. Therefore, constructing an effective text classification system is very necessary. Nowadays, Text Classification (TC) is widely used in various fields [34].

This paper aims at designing a text representation and an indexing method, reflecting more semantics and constructing effective classification of algorithms in order to improve Arabic Text Classification performance. Indeed, the representation model used by most researchers is the vector space model (VSM), when the document is represented as a vector of terms (terms are simple words (vocabulary), *n*-gram, keywords or longer sentences). This model has some limitations such as the high vectors dimensionality, the loss of the order, documents containing similar contexts, but different vocabulary terms are not classified in the same category.

Recently, there has been a progress in the models of document description; this progress is based on techniques which embed more and more semantics. These models are known for the generative aspect, they can provide a method to achieve correct syntactic and semantic description of texts. Among these models we quote the LDA (Latent Dirichlet Allocation); the basic idea is that a document is a mixture probability of (hidden) latent themes (topics). Then, every topic is characterized by a probability distribution of words which are associated with it. We thus see that the key element is the notion of theme i.e. that semantics is prioritized over the notion of term or word.

In this paper, we have used statistic topic model (LDA) to index and represent the Arabic texts,

we extracted significant topics from all texts, each theme is described by a particular distribution of descriptors (probability distribution of words) then each text is represented on the vectors of these topics, so we have reduced the dimentionality of the descriptor vector and we have introduced more semantic information in the text representation using the notion of topic in coding instead of the terms or words.

The classification does not represent an end in itself; the classification treatment is a step in a much more complex cognitive process. That is why we propose in our research, as part of an OREILLODULE project [25], a real time synthesis, recognition and translation system of the Arabic language developed by the UTIC project (the Monastir's faculty of Science unit), to establish a framework for indexation and Arabic text classification: Indeed, the project "oreillodule" has three sub-systems. The subsystem of translation can sometimes use the classification to disambiguate semantics on some words.

The rest of this paper is organized as follows. Section 2 summarizes the Arabic text classification and text representation related work. Section 3 describes our Arabic text classification framework. Experimental results are shown in Section 4. Section 5 draws some conclusions and outlines future work.

2. Related Works

Text Classification (TC) may be defined as the task of assigning a Boolean value to each pair

$$\langle d_i, c_j \rangle \in D * C$$

where D is a domain of documents,
$C = \{c_1, c_2, \ldots, c_{|c|}\}$ is a set of pre-defined categories.

A value of *True* assigned to $\langle d_i, c_i \rangle$ indicates a decision to file d_i under c_j, while a value of *False* indicates a decision not to file under. More formally, the task is to define the function f (call the classifier) that describes how documents should be classified.

Most of the TC research is designed and tested for English languages articles. However, some TC approaches were carried out for other European languages such as German, Italian and Spanish [1], and others were carried out for Chinese and Japanese [2, 3]. Compared to English, Arabic language has an extremely rich morphology and a complex orthography; this is one of the main reasons [4, 6] behind the lack of research in the field of Arabic TC. Some machine learning approaches have been proposed to classify documents: SVMs with CHI square feature extraction method [7], SVM with HMM feature extraction method for Web News Classification [31], Naïve Bayesian method [8], *k*-nearest neighbors (kNN) [9, 30], maximum entropy [10], distance based classifier [11, 12], decision trees and Associative Classification [28, 27], Rocchio algorithm [13] and WorldNet knowledge based [14].

It is quite hard to fairly compare the effectiveness of these approaches because of the following reasons:

(i) Their authors have used different corpora (because there is no publicly available Arabic TC corpus).

(ii) Even those who have used the same corpus, it is not obvious whether they have used the same documents for training/testing their classifiers or not.

(iii) Authors have used different evaluation measures: accuracy, recall precision and F measures.

Recently, methods based on Machine Learning have been the most studied in Text Classification. The documents are coded as the so-called Bag of Words (BoW). The prediction of document category is obtained on the basis of feature selection and statistical learning algorithms. Studies on the text representation and classification by using statistical theory and linguistic knowledge have been expanded and developed. The following is a brief summary of the steps necessary to build a classification system (Figure 1).

Text representation: includes preprocessing documents in a training set (removes stop words); selects features and represents documents.

Training classifier: partitions sample space according to text distribution and trains the classifier in each space separately.

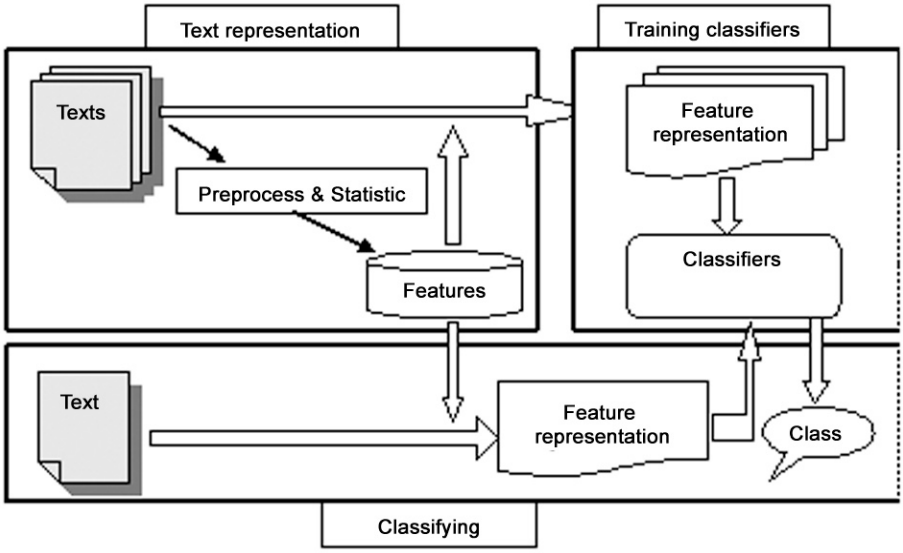

Figure 1. TC process.

Classifying: represents new documents with feature word; computes the space that each document belongs to and classifies each document by adopting corresponding classifier.

3. Arabic Language Structure

Arabic is the mother tongue of more than 300 million people [32]. Unlike Latin-based alphabets, the orientation of writing in Arabic is from right to left, the Arabic alphabet consists of 28 letters. Arabic language is a highly inflected language, it has much richer morphology than English. Arabic words have two genders, feminine and masculine; three numbers, singular, dual, and plural; and three grammatical cases, nominative, accusative, and genitive. A noun has the nominative case when it is the subject; accusative when it is the object of a verb; and the genitive when it is the object of a preposition.

Arabic grammarians describe Arabic in terms of noun, verb, and particle:

— A noun is a name or a word that describes a person, a thing, or an idea.

— Similar to English verbs, verbs in Arabic are classified into Perfect, Imperfect, and Imperative.

— Arabic particles include prepositions, adverbs, conjunctions, interrogative particles, exceptions, and interjections.

All verbs and some nouns are morphologically derived from a list of roots. Words are formed by following fixed patterns, the prefixes and suffixes are added to the word to indicate its number, gender and tense [36].

Most of Arabic words are derived from the pattern *Fa'ala* (فعل), all words following the same pattern have common properties and states. For example, the pattern *Faa'el* (فـاعل) indicates the subject of the verb, the pattern *maf'ool* (مفعول) represents the object of the verb. Table 1 shows different derivations for the root word *kataba* (كتب), its pattern, its pronunciation and the translation of the word in English to show the effect of these derivations on the meaning. The letters that have been added to the main root of the word are underlined.

Arabic word	Pattern	Pronunciation	English meaning
كتب	*Fa'ala* (فعل)	*Kataba*	Wrote
كـاتب	*Faa'el* (فـاعل)	*Kateb*	Writer
مكتـوب	*maf'ool* (مفعول)	*Maktoob*	Is written

Table 1. Different derivations for the root word kataba كتب.

In addition to the different forms of the Arabic word that result from the derivational process, most connectors, conjunctions, prepositions, pronouns, and possession forms are attached to the Arabic surface form as prefixes and suffixes. For instance, the definitive nouns are formed by attaching the article (ال) to the immediate front of the nouns (acts as "The"). The conjunction word (و) (and) is often attached to the word. The letters (ف ، ل ، ب ، ك) can be added to the front to the word as prepositions. The suffix (ة) is attached to represent the feminine gender of the word, (ان) is for dual masculine in the nominative case, (ين) is for dual masculine in both the accusative and the genitive cases, (ون) is for plural masculine in the nominative case. The plural suffix (ات) is used in case of feminine gender for the three grammatical cases. Also some suffixes are added as possessive pronouns, the letter (ه) is added to represent the possessive pronoun (His), (ها) for (Her), (ي) for (My), and (هم, هن) for (Their). Table 2 shows different affixes that may be added to the word (معلم) (Teacher), the affixes attached to the word are underlined, also the table shows the corresponding meaning of the word in English along with its gender and number state.

Arabic word	English meaning	Gender	Number
معلم	Teacher	Masculine	Singular
معلمة	Teacher	Feminine	Singular
معلمات	Teachers	Feminine	Plural
معلمهم	Their teacher	Masculine	Singular

Table 2. Different affixes that may be added to the word معلم.

3.1. Text Representation

Modeling text corpora aims at finding short descriptions of the members of a collection that enable efficient processing on large collections while preserving the essential statistical relationships that are useful for classification.

3.1.1. Vector Space Model

The representation model best known by most researchers is the vector space model (VSM) which is proposed by Salton, Wong and Yang and is an algebraic model for representing text documents as vectors of identifiers, such as, for example, index terms. It is used in Text Classification, Information Filtering and Information Retrieval etc [15].

In VSM, a document is represented as a vector. Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is not null. Several methods to calculate these values are known, (term) weights $\boldsymbol{w}$ were developed. Among these weighting methods, the best known is $tf * idf$ where tf is frequency of term t in document d and idf is inverse document frequency.

$$w_{t,d} = tf * idf = tf_t * \log \frac{|D|}{|\{t \in d\}|} \qquad (1)$$

Where: $|D|$: total number of documents in the data set
$|\{t \in d\}|$: number of documents containing the term t.

The promising definition of term depends on the application. Terms are words (vocabulary), n-gram, keywords or longer sentences. If words are chosen to be the terms, the dimension of the vector is the number of words in the vocabulary. The similarity between documents can be calculated by computing the cosine of the angle between the vectors of the document.

In Text Classification task, the vector space model has the following limitations:

— In the vector representation we lose the order in which these terms appear in the document.

— Documents containing similar contexts but different terms vocabulary are not classified as the same category.

Statistical topic models have been successfully applied in many tasks, including Information Classification, Information Retrieval and Data Mining, etc[16, 17]. These models can capture the word correlations in the corpus with a low-dimensional set of multinomial distribution, called "topics", and find a relatively short description for the documents. LDA is a widely

used generative topic model. In LDA, a document is viewed as a distribution over topics, while a topic is a distribution over words.

3.1.2. Latent Dirichlet Allocation Model

Formally, we define the following terms [17]:

— A ***word*** is the basic unit of discrete data, defined to be an item from a vocabulary indexed by $\{1,\ldots,V\}$. We represent words using unit-basis vectors that have a single component equal to one and all other components equal to zero. Thus, using superscripts to denote components, the vth word in the vocabulary is represented by a $\boldsymbol{V}$-vector $\boldsymbol{w}$ such that $\boldsymbol{w}^v = \boldsymbol{1}$ and $\boldsymbol{w}^u = \boldsymbol{0}$ for $u \neq v$.

— A ***document*** is a sequence of N words denoted by $\boldsymbol{w} = (w_1; w_2, \ldots, w_N)$, where w_n is the nth word in the sequence.

— A ***corpus*** is a collection of M documents denoted by $\boldsymbol{D} = \{\boldsymbol{w}_1; \boldsymbol{w}_2; \ldots; \boldsymbol{w}_M\}$.

LDA is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

LDA assumes the following generative process for each document $\boldsymbol{w}$ in a corpus $\boldsymbol{D}$:

1. Choose $N \sim Poisson(\xi)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the N words w_n:
   a) Choose a topic $Z_n \sim Multinomial(\theta)$.
   b) Choose a word w_n from $p(w_n|z_n,\beta)$, a multinomial probability conditioned on the topic z_n.

Several simplifying assumptions are made in this basic model, some of which we remove in subsequent sections. First, the dimensionality k of the Dirichlet distribution (and thus the dimensionality of the topic variable z) is assumed known and fixed. Second, the word probabilities are parameterized by a $k \times V$ matrix β where $\beta_{i,j} = p(w_j = 1|z_i = 1)$, which for now we treat as a fixed quantity that is to be estimated. Finally, the Poisson assumption is not critical to anything that follows and more realistic document length distributions can be used as needed. Furthermore, note that N is independent of all the other data generating variables (θ and z). It is thus an ancillary variable and we will generally ignore its randomness in the subsequent development.

A k-dimensional Dirichlet random variable θ can take values in the $(k-1)$-simplex (a k-vector θ lies in the $(k-1)$-simplex if $\theta_i \geq 0$, $\sum_{i=1}^{k} \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta/\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1} \qquad (2)$$

Where the parameter α is a k-vector with components $\alpha_i > 0$, and where $\Gamma(x)$ is the Gamma function. The Dirichlet is a convenient distribution on the simplex – it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w are given by:

$$\begin{aligned} &p(\theta, z, w/\alpha, \beta) \\ &= p(\theta/\alpha) \prod_{n=1}^{N} p(z_n/\theta) p(w_n/z_n, \beta) \end{aligned} \qquad (3)$$

Where $p(z_n/\theta)$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating over θ and summing over z, we obtain the marginal distribution of a document:

$$\begin{aligned} &p(W/\alpha, \beta) = \int p(\theta/\alpha) \\ &\left(\prod_{n=1}^{N} \sum_{2z} p(z_n/\theta) p(w_n/z_n, \beta) \right) d\theta \end{aligned} \qquad (4)$$

Finally, taking the product of the marginal probabilities of single documents, we obtain:

$$\begin{aligned} &p(D/\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d/\alpha) \\ &\left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}/\theta_d) p(w_{dn}/z_{dn}, \beta) \right) d\theta_d \end{aligned} \qquad (5)$$

The parameters α and β are corpus level parameters, assumed to be sampled once in the process of generating a corpus. The variables θ_d are document-level variables, sampled once per document. Finally, the variables z_{dn} and w_{dn} are word-level variables and are sampled once for each word in each document.

3.2. Text Classifier (SVM)

Support Vector Machine (SVM) classifiers are binary classifiers, which were originally proposed by Vapnik [20]; SVM are a set of related supervised learning methods used for classification. A support vector machine constructs a hyper plane or set of hyper planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since, in general, the larger the margin the lower the generalization error of the classifier [18].

We are given some training data D, a set of n points of the form as Formula

$$D = \{(x_i, c_i)/x_i \in R^p, c_i \in \{-1, 1\}\}_{i=1}^{n} \quad (6)$$

Where, the c_i is either 1 or -1, indicating the class to which the point x_i belongs. Each x_i is a p-dimensional real vector. We want to find the maximum-margin hyper plane that divides the points having $c_i = 1$ from those having $c_i = -1$. Maximum-margin hyper plane and margins for a SVM trained with samples from two classes. Samples on the margin are called the support vectors. Any hyper plane can be written as the set of points x satisfying:

$$x.w - b = 0 \quad (7)$$

Where, "$x.w$" denotes the dot product. The vector w is a normal vector: it is perpendicular to the hyper plane. The parameter $b/||w||$ determines the offset of the hyper plane from the origin along the normal vector w.

Note that if the training data are linearly separable, we can select the two hyper planes of the margin in a way that there are no points between them and then try to maximize their distance. By using geometry, we find the distance between these two hyper planes is $2/||w||$,

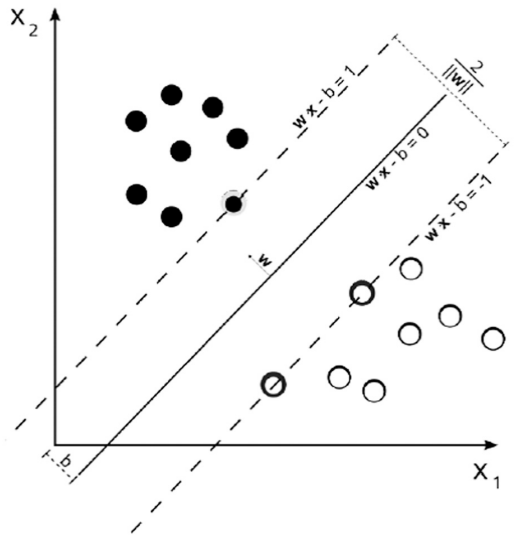

Figure 2. Principle of SVM classifier.

so we want to minimize $||w||$. As we also have to prevent data points falling into the margin, we add the following constraint:

$$c_i(x_i.w - b) \geq 1, \text{ for all } 1 \leq i \leq n \quad (8)$$

We can put this together to get the optimization problem: Minimize in (w, b), $||w||$ subject to formula 8.

Writing the classification rule in its unconstrained dual form reveals that the maximum margin hyper plane and therefore the classification task is only a function of the support vectors, the training data that lie on the margin.

The dual of the SVM boils down to the following optimization problem that is maximizing (in α_j).

$$Q(a) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j c_i c_j (x_i.x_j) \quad (9)$$

It is subject to $\alpha_i \geq 0$, $i = 1, \ldots, n$ and

$$\sum_{i=1}^{n} a_i c_i = 0 \quad (10)$$

The α terms constitute a dual representation for the weight vector in terms of the training set:

$$w = \sum_{i} a_i c_i x_i \quad (11)$$

After solving the above problem, the optimized classification function is as follows:

$$f(x) = \text{sgn}\{(w.x) + b\} = \text{sgn}\left\{\sum_{i=1}^{n} a_i^* c_i (x_i.x) + b^*\right\} \quad (12)$$

Why do we use Support Vector Machine Classifiers for TC Tasks?

Empirical results in Text Classification have shown that SVM classifiers perform well, simply because of the following text properties [19]:

- High dimensional input space: In text documents, we are dealing with a huge number of features. Since SVM classifiers use overfitting protection, which does not necessarily depend on the number of features, SVM classifiers have the potential to handle a large number of features.
- Few irrelevant features: we assume that most of the features are irrelevant to avoid these high dimensional input spaces. Feature subset selection methods try to determine these irrelevant features (in TC tasks, there are many relevant features).
- Document vectors are sparse: For each document, the corresponding document vector contains only a few entries that are not zero.
- Most text classification problems are linearly separable: the idea of SVMs is to find such linear (or polynomial, RBF, etc...) separators [19].

These arguments give theoretical evidence that SVMs should perform well for text categorization. SVM is well suited for problems with dense concepts and sparse instances.

One of the main advantages of SVM classifiers over other conventional methods is their robustness.

4. Arabic Text Classification Framework

4.1. Arabic Text Representation

4.1.1. Arabic Dataset Preprocessing

Arabic documents are processed according to the following algorithm:

Normalized Algorithm:

Begin

1. Each article in the Arabic dataset is processed to remove digits and punctuation marks $\{., :, /, !, \S, \&, ', [, (, \_, -, |, -, \hat{},),], \}, =, +, \$, *, ...\}$.
2. Remove all vowels except "ّ" (الشدة).
3. Duplicate all the letters containing the symbols "ّ" (الشدة).
4. Convert letters "ء" (hamza), "آ" (aleph mad), "أ" (aleph with hamza on top), "ؤ" (hamza on w), "إ" (alef with hamza at the bottom), and "ئ" (hamza on ya) to "ا" (alef). The reason for this conversion is the fact that all forms of hamza are represented in dictionaries as one form and people often misspell different forms of aleph.
5. Convert the letter "ى" to "ي" and the letter "ة" to "ه". The reason behind this normalization is the fact that there is not a single convention for spelling "ى" or "ي" and "ة" or "ه" when they appear at the end of a word.
6. All the non Arabic words were filtered.
7. Arabic function words, such as

   من, في, الى, يلي , ضد, بعد, ان, فما, فسوف
   , وكان, على, احد, وليس, به, يكون, وهو, حتى
   ,وعلى, ان, عليها, فيها, وبين, التي, كذلك, تلك
   ,حين, اما , الذي, منذ, ليس , مساء, عن, لكن
   ..., حول, عنه, ما, اي, وكانت, ليسب, لا, ومن

   etc., were removed. The Arabic function words (stop words) are the words that are not useful in documentary research system e.g. pronouns and prepositions.
8. Applied stemming algorithm (described in the next section) for each article in Arabic data set to obtain a stemmed text.

End algorithm.

4.1.2. A Generic Hybrid Stemmer Algorithm Based Learning for Arabic

We have built our stemmer [11] inspired by the various existing approaches [29] (recognition of form, stemming algorithm and manual stemming). We tried to integrate them to set up a successful tool; also we exploited the wealth of the Arabic language grammar to incorporate the concept of grammatical classes in the lemmatizer. Here is an example of lemmatized text (Figure 3).

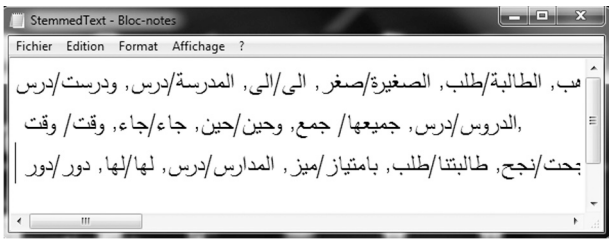

Figure 3. Example of stemmed text.

• Form recognition

Each word in process undergoes a search in the reference database to see if it would not be:

— A particular case (name, number, preposition, adverb ...)

— A root or lexicon

— A case already handled by the lemmatizer

These elements are classified in the database as templates. By comparison, it is possible to attach a word to a model and extract its decomposition.

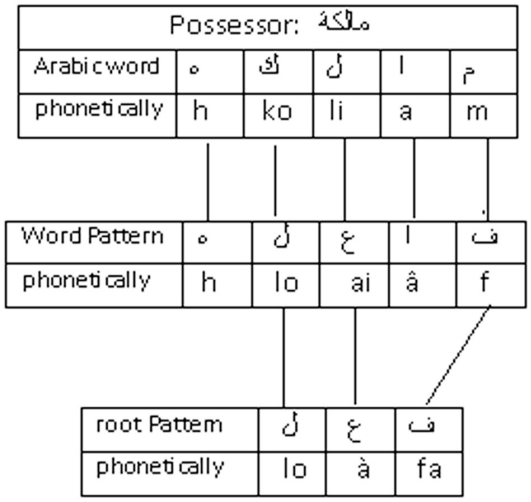

Figure 4. Extraction of a form's matching root.

• Algorithm stemming

We note that the structure of a language determines the algorithm to use. In the case of Arabic, we are inspired by Tim Buckwalter algorithm [26] to perform morphological analysis on transcripts of texts in Arabic. Our algorithm is based on the principle of generating the various possibilities of prefixes, suffixes and schematic body from a word and keeping only the trio which belongs to a reference base, among which is the prefix, the suffix and the simplistic body belonging to the language.

The principle for texts written in Arabic is more or less the same with some changes and enrichments which we introduced at the prefixes and suffixes; in fact, after consulting an expert we could rebuild all prefixes and suffixes of Arabic. We implemented them in Arabic character to allow the recognition compared to untranscribed texts. The most common structure of Arabic words is devised into five components (i.e. antefixes, prefix, body schematic, suffix, postfix). We hypothesize in our algorithm that: the prefix refers to the combination (prefix + antefixes) and the suffix refers to the combination (suffix + postfix).

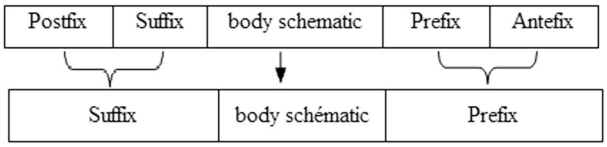

Figure 5. Combined structure.

This fusion will minimize the number of possible decompositions without diminishing accuracy. The merging of elements reduces the number of possibilities generated from a single word; the more we know that merged elements can after their detection be decomposed more easily. The example in Figure 6 illustrates this set of combinations.

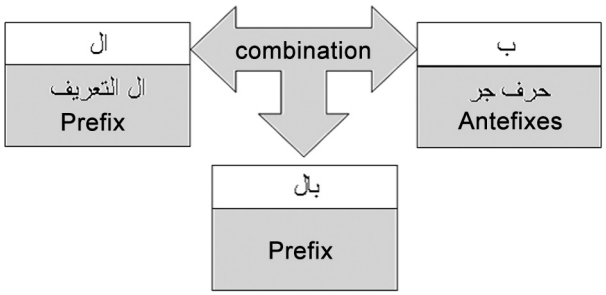

Figure 6. Set of combinations.

Stemming algorithm:

The algorithm is based on the following assumptions:

— *A word is broken down into: prefix, suffix and body schematic.*

— *The prefix varies from 0 to 4 characters.*

— *The body has a size schematic rated X is greater than 2.*

— *The suffix varies from 0 to 6 characters.*

— *The analyzer treats the texts word by word.*

Step 1: Generate all possible decompositions.

Step 2: Validate the prefix for every decomposition by referring to the database.

Step 3: Validate the suffix for the decompositions whose prefixes are validated, by referring to the database.

Step 4: Validate the body schematic for the decomposition with the prefix and suffixes validated.
***If** the body schematic exists in the database it's Validated*
***Else** confirm with the user*
***If** a new body schematic is validated by the user, it is automatically inserted with its decomposition in the table.*

Step 5: Provide, where there are several possible decompositions, the user choosing the right one.

Step 6: In the case of non-validation of the user, the user can manually edit an entry.

• The manual stemming

It is very difficult to automate the stemmer process for the words in Arabic language [25]; the human intervention remains inevitable whether it is at the level application software or the level of the check.

For that purpose, we do not miss to note that the manual lemmatization is the last resort in our system. One of the problems posed by the approaches mentioned above is that their treatment can generate several possible decompositions. At this level there is human intervention to guide the resolution either by choice or by calling one of the other approaches integrated in our system.

The used lemmatization methods (a based lemmatization models, manual and based algorithm) although different in appearance, they represent the main points of our system.

We have implemented interfaces allowing the communication between them and we have created an organization to optimize their use. The integration of three approaches is done sequentially, one after another, when one of them jams, the other takes over. Note that each stage interacts with the knowledge base, whether in consultation or updated.

• The database references

By using the data needs expressed in the system previously detailed, we have established the structure of the reference database.

The basics: This database is the core of the system that we have achieved in collaboration with an expert of the Arabic language, it includes:

— A lexicon: it encompasses two entities, namely word split into prefix, suffix and body schematic and roots (or body schematic algorithm for the decomposition). Our initial source in the diet of the lexicon is an XML file containing a dictionary.

— Special cases: We could include this class in the lexicon, but it is better to separate them for easier processing. It includes proper nouns, prepositions, articles ... We determined all these elements with the help of an expert in Arabic grammar therefore we were able to synthesize them.

— The cases dealt with: represent the already treated words whether by decomposition or by adding the user. They are stored with their decomposition to be used by the recognition module forms.

— Prefixes and suffixes: are essential to stemming algorithm. We were able, with the help of an expert, to determine a set of 77 prefixes and 165 suffixes in the Arabic language [11].

4.2. Latent Dirichlet Allocation Model in Classification

4.2.1. Why Do We Use LDA Model?

A challenging aspect in the document classification problem is the choice of features. Treating individual words as features yields a rich but very large feature set [22]. One way to reduce

this feature set is to use an LDA model for dimensionality reduction.

In particular, LDA reduces any document to a fixed set of real-valued features — the posterior Dirichlet parameters $\gamma^*(w)$ associated with the document. It is of interest to see how much discriminatory information we lose in reducing the document description to these parameters.

Another character of LDA is that once the distribution has been established for each document in the collection, the ones that share one common topic can be placed in the same group. As each document is a mixture of different topics, in the classifying process, the same document can be placed in more than one group.

4.2.2. Construct LDA Model

To construct LDA model for a document $d = (w_1, w_2, \ldots, w_N)$ the following steps are necessary.

Model a k-dimensional random vector θ as a Dirichlet distribution $Dir(\alpha)$. This step generates θ as parameter for the probability distribution of latent variable z_n as described in Formula (2).

For n-th term, model n-th latent variable z_n as a multinomial distribution $Multi(\theta)$. z_n is k-dimensional vector with k-of-1 schema.

$$p(z_n/\theta) = \prod_{i=1}^{k} \theta_i^{z_{n,i}} \qquad (13)$$

Model w_n as a multinomial distribution $Multi(z_n, \beta)$, where β is a $k \times V$ parameter matrix. Indeed, this step is using Z_n to choose one row of β as the parameter for the probability distribution of Z_n. Together with the whole space of Z_n, it indeed makes a mixture multinomial distribution for w_n.

$$p(w_n/z_n, \beta) = \prod_{j=1}^{V} (\sum_{i=1}^{k} z_{n,i}, \beta_{i,j})^{w_{n,j}} \qquad (14)$$

Once given the parameters α and β, the joint distribution of θ, $Z_n = \{Z_1, Z_2, \ldots, Z_N\}$ and $d = \{w_1, w_2, \ldots, w_N\}$ is given by formula (2).

The marginal distribution of a document d is given by:

$$p(d/\alpha, \beta) = \int_{\theta} p(\theta, z, w/\alpha, \beta) \qquad (15)$$

For a corpus D including M documents, the data log-likelihood is given by:

$$\begin{aligned} L(D; \alpha, \beta) &= \log p(D/\alpha, \beta) \\ &= \sum_{d \in D} \log p(d/\alpha, \beta) \end{aligned} \qquad (16)$$

The task of model documents is to learn the various distributions (the set of topics, their associated word probabilities, and the topic of each word and the particular topic mixture of each document) which are a problem of Bayesian inference. Fundamentally, it is to find optimal α and β that maximize the data log-likelihood $L(D; \alpha, \beta)$.

The original paper [17] used a variational Bayes approximation of the posterior distribution; we describe a simple convexity-based variational algorithm for inference in LDA. The basic idea is use of Jensen's inequality to obtain an adjustable lower bound on the log likelihood. One considers a family of lower bounds, indexed by a set of variation parameters. The variation parameters are chosen by an optimization procedure that attempts to find the tightest possible lower bound. This family is characterized by the following variation distribution.

$$q(\theta, z/\gamma, \phi) = q(\theta/\gamma) \prod_{n=1}^{N} q(z_n/\phi_n) \qquad (17)$$

Where, the Dirichlet parameter γ and the multinomial parameters $(\phi_1; \ldots; \phi_N)$ are the free variation parameters.

Having specified a simplified family of probability distributions, the next step is to set up an optimization problem that determines the values of the variation parameters γ and ϕ. The desideratum of finding a tight lower bound on the log likelihood translates directly into the following optimization problem:

$$\begin{aligned} (\gamma^*, \phi^*) = \arg \min_{(\gamma, \phi)} & D(q(\theta, z/\gamma, \phi) \\ & //p(\theta, z/w, \alpha, \beta)) \end{aligned} \qquad (18)$$

Thus the optimizing values of the variation parameters are found by minimizing the Kullback-Leibler (KL) divergence between the variation

distribution and the true posterior $p(\theta, z/w, \alpha, \beta)$. This minimization can be achieved via an iterative fixed-point method. In particular, by computing the derivatives of the KL divergence and setting them equal to zero, we obtain the following pair of update equations:

$$\phi_{ni} \propto \beta_{iw_n} \exp\left\{\psi(\gamma_i) - \psi(\sum_{j=1}^{k} \gamma_i)\right\} \quad (19)$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \quad (20)$$

Where, ψ is the first derivative of the log Γ function which is computable via Taylor approximations [21].

It is important to note that the variation distribution is actually a conditional distribution, varying as a function of $\boldsymbol{w}$. This occurs because the optimization problem in Formula (10) is conducted for fixed $\boldsymbol{w}$, and thus yields optimizing parameters (γ^*, ϕ^*) that are a function of $\boldsymbol{w}$. We can write the resulting variation distribution as $q(\theta; z/\gamma^*(w), \phi^*(w))$, where we have made the dependence on w explicit. Thus the variation distribution can be viewed as an approximation to the posterior distribution $p(\theta, z/w, \alpha, \beta)$.

In the language of text, the optimizing parameters $(\gamma^*(w), \phi^*(w))$ are document-specific. In particular, we view the Dirichlet parameters $(\gamma^*(w)$ as providing a representation of a document in the topic simplex).

As described above, variation inference provides us with a tractable lower bound on the log likelihood, a bound which we can maximize with respect to α and β. We can thus find approximate empirical Bayes estimates for the LDA model via an alternating variation EM procedure that maximizes a lower bound with respect to the variation parameters γ and ϕ, and then, for fixed values of the variational parameters, maximizes the lower bound with respect to the model parameters α and β.

The derivation yields the following iterative algorithm:

— ***(E-step)*** For each document, find the optimizing values of the variation parameters $\{\gamma_d^*, \phi_d^* : d \in D\}$. This is done as described in the previous section.

— ***(M-step)*** Maximize the resulting lower bound on the log likelihood with respect to the model parameters α and β. This corresponds to finding maximum likelihood estimates with expected sufficient statistics for each document under the approximate posterior which is computed in the E-step.

LDA model starts with a set of topics. Each of these topics has probabilities of generating various words. Words without special relevance, like articles and prepositions, will have roughly even probability between classes (or can be placed into a separate category). A document is generated by picking a Dirichlet distribution over topics and, given this distribution, picking the topic of each specific word. Then words are generated given their topics. The parameters of Dirichlet distribution are estimated by variation EM algorithm.

4.3. Text Classification Algorithm Based on Latent Dirichlet Allocation

The LDA model is used to classify document in a discriminative framework. For each document d, we are using LDA to learn γ_d for that document treat γ as the features, and we use SVM classifier to attach class label. Normally the number of topics is rather smaller than the number of terms, thus LDA can effectively reduce the feature dimension. The whole process is described in this algorithm:

Algorithm LDA-SVM:

Let D be a corpus, supposed that it has been split into training set and testing set.

Input: Training set and testing set

Output: Class of documents in testing set

1. *Preprocess documents in training*
2. *Learn parameters of LDA and get θ (matrix of "document*topic") and ϕ (matrix of "topic*word") in the case of appointing the value of "K".*
3. *Preprocess documents in testing set*
4. *Model documents in testing set according to the parameter got from step 2, that is, transform documents in testing set into the form of matrix "document*topic".*
5. *Perform classification on corpus using SVM classifier, that is, input the matrix "document*topic" of training set and testing set into SVM classifier.*
6. *Evaluate classification results by using various metrics.*

5. Experimental Validation and Discussions

5.1. Arabic Data Collection

To test the effectiveness of our Arabic classification system and to evaluate the effectiveness of the proposed LDA-SVM algorithm, we have used an in house corpus collected from online Arabic magazines and newspapers, including al-Jazeera, al-Nahar, al-Ahram and other specialized websites. The corpus contains 1500 documents that vary in length and writing styles [33].

These documents fall into nine classification categories that vary in the number of documents. In this Arabic dataset, each document was saved in a separate file within the directory for the corresponding category, i.e., the documents in this dataset are single-labeled. Table 3 shows the number of documents in each category.

Category	Training Texts	Testing Texts	Total Number
Society	54	31	85
Economics	147	73	220
International	51	27	78
Arts	77	38	115
Culture	75	32	107
Medicine	155	77	232
Politics	123	61	184
Religion	152	85	237
Sports	155	87	242
Dataset total number	989	511	1500

Table 3. TC Arabic dataset.

Figure 7 gives the distribution of document amount in each class in our corpus. The largest class includes nearly 250 documents while the smallest class includes only 85 documents.

We adopt the open source of LDA [23] to model our corpus and we set topic number as $K = 50$ in LDA model.

TC effectiveness [24] is measured in terms of Precision, Recall, and the F_1 measure. Denote

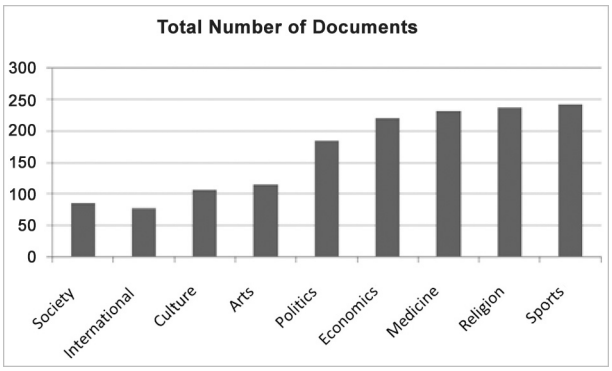

Figure 7. Document distrubution of data set.

the precision, recall and F_1 measures for a class C_i by P_i, R_i and F_i, respectively. We have:

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{21}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{22}$$

$$F_i = \frac{2P_iR_i}{R_i + P_i} = \frac{2TP}{FP_i + FN_i + 2TP_i} \tag{23}$$

Where: TP_i: (true-positive): number of documents correctly assigned.
FP_i: (false positives): number of documents falsely accepted.
FN_i: (false-negative): number of documents falsely rejected.

To evaluate the classification performance for each category, precision, recall, and the F_1 measure are used. To evaluate the average performance over many categories, the macro-averaging $F_1(F_1^M)$, micro-averaging $F_1(F_1^u)$, are used and defined as follows:

$$F_1^M = \frac{2}{N} \cdot \frac{\sum_{i=1}^{|C|} R_i \sum_{i=1}^{|C|} P_i}{\sum_{i=1}^{|C|} R_i + \sum_{i=1}^{|C|} P_i} \tag{24}$$

$$F_1^u = \frac{2\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} FP_i + \sum_{i=1}^{|C|} FN_i + \sum_{i=1}^{|C|} TP_i} \tag{25}$$

Table 4 shows the results of the LDA-SVM classifier. Results are shown in terms of Precision, Recall and the F-measure. The Macro-averaging F_1 score is 0.881 and the Micro-averaging F_1 score is 0.914.

Category	Precision	Recall	F_1-Measure
Society	0.785	0.68	0.728
Economics	0.93	0.714	0.807
International	0.857	0.857	0.857
Arts	0.969	0.969	0.969
Culture	0.928	0.812	0.866
Medicine	0.95	0.987	0.968
Politics	0.90	0.762	0.825
Religion	0.961	0.986	0.973
Sports	0.991	0.857	0.919
Macro-averaging F_1 measure $= 0.881$			
Micro-averaging F_1 measure $= 0.914$			

Table 4. LDA-SVM results.

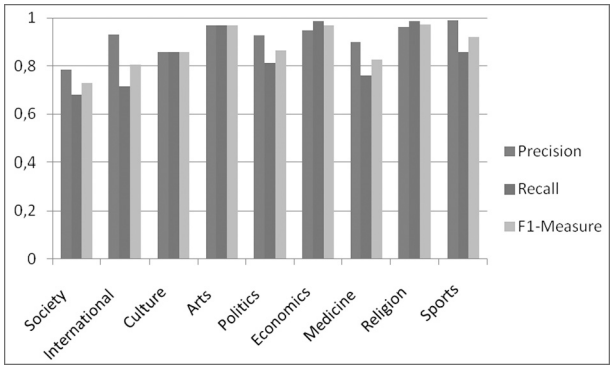

Figure 8. Histogram of Precision/Recall/F-measure for each class.

5.2. Comparison

For comparison purposes, we have used the same pre-processing steps to implement the Naïve Bayes and k Nearest Neighbour (kNN) classifiers. As shown in table 5, it is obvious that the LDA-SVM classifier outperforms the SVM (without LDA), Naïve Bayes and kNN classifiers.

Figure 9 and Figure 10 show separately the precision and recall metric on each class. We could observe that the tendency of precision is the same for small classes (International, Society, Culture, Art) for LDA-SVM, Naïve Bayes and KNN. But for big classes (Economics, Medcine, Religion, Sport), the value of precision on LDA-SVM is higher than those of Naïve-Bayes and KNN.

Classifier Type	Macro -averaging F_1 measure	Micro -averaging F_1 measure
LDA-SVM	0.881	0.914
SVM	0.834	0.874
Naïve Bayes	0.82	0.845
kNN	0.756	0.727

Table 5. Different classifiers performance in Arabic TC tasks.

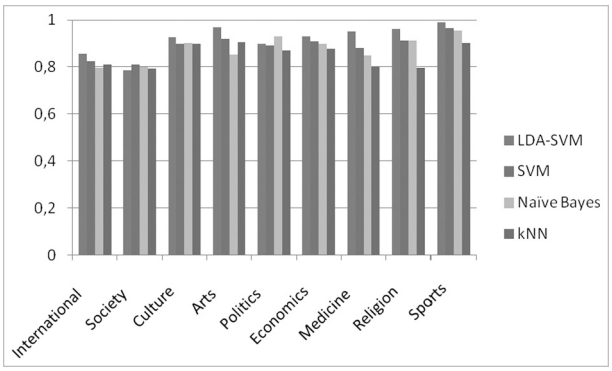

Figure 9. Precision values on each class.

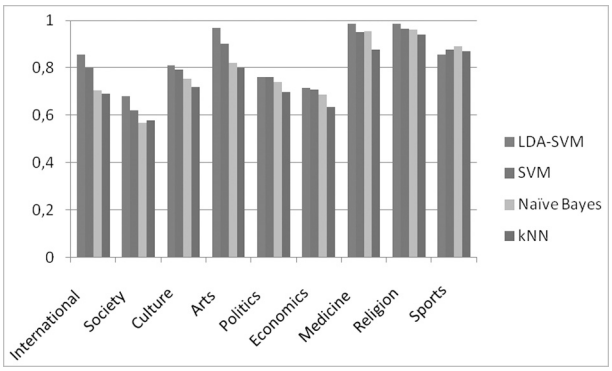

Figure 10. Recall values on each class.

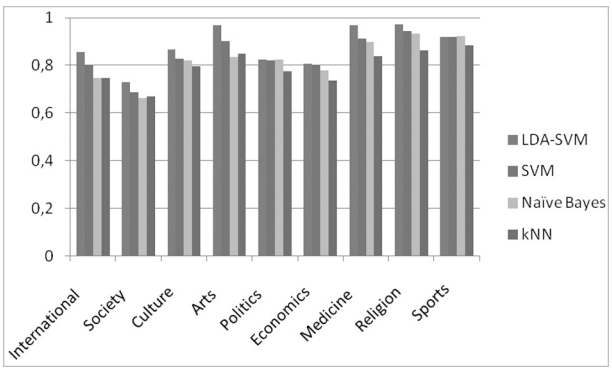

Figure 11. F_1 value on each class.

On the contrary, the value of recall on LDA-SVM is higher for small classes, but for big classes LDA-SVM is near to Naïve Bayes and KNN [35].

Figure 11 shows the F_1 performance on each class; we could find that F1 values of LDA-SVM algorithm are always higher than those of Naïve Bayes and KNN.

We summarize the performance of three metrics and conclude: LDA-SVM enhances classification performance mainly by improving classification effectiveness on big classes, because for small classes we detected a problem at the level of the smoothing model LDA (LDA allocation of zero probability for the terms that do not appear in learning documents) where the need to develop a new smoothing algorithm for the effectiveness of the classification system is independent of the size of the classes.

6. Conclusion

Our work focuses on the Arabic text classification. Our main contribution is the proposal of a text representation model and a hybrid algorithms classification based on Latent Dirichlet Allocation and Support Vector Machine Classifiers. This was achieved through creating an Arabic text classification framework to evaluate our approach[38].

Current researches usually adopt VSM to represent text in Text Classification applications. In this way, the document is coded as a vector of words; this results in huge feature space and semantic loss. The proposed model in this paper adopts "topic" sampled by LDA model as text features. The proposed algorithm LDA-SVM extracts topics statistically from texts and then texts are represented by using the topic vector. It effectively avoids the above problems.

An experimental stage was implemented to evaluate the effectiveness of our Arabic classification system and the proposed LDA-SVM algorithm. The results show the effectiveness of our system which exceeds that of Naive Bayes and KNN in term of precision and recall and F-measure for the bigger classes[37].

To make our system independent of the different sizes of the classes, we propose, as a future project, developing a new smoother LDA-SVM algorithm.

References

[1] F. CIRAVEGNA, et al., Flexible Text Classification for Financial Applications: the FACILE System, *Proceedings of PAIS-2000, Prestigious Applications of Intelligent Systems Subconference of ECAI2000*, 2000, pp. 696–700.

[2] F. PENG, X. HUANG, D. SCHUURMANS AND S. WANG, Text Classification in Asian Languages without Word Segmentation, *Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages*, Association for Computational Linguistics, July 7, Sapporo, Japan, 2003, pp. 41–48.

[3] J. HE, A-H. TAN AND C-L. TAN, On Machine Learning Methods for Chinese Document Categorization, *Applied Intelligence*, 2003, pp. 311–322.

[4] A. M. SAMIR, W. ATA AND N. DARWISH, A New Technique for Automatic Text Categorization for Arabic Documents, *Proceedings of the 5th Conference of the Internet and Information Technology in Modern Organizations*, December, Cairo, Egypt, 2005, pp. 13–15.

[5] Sakhr Company: `http://www.sakhr.com`.

[6] A. BEN HAMADOU, *Vérification et Correction Orthographique des Textes Arabes à Partir d'une Analyse Affixale des Textes Ecrits en Langage Naturel : le cas de l'Arabe non Voyellé*, Thèse de Doctorat ES-Sciences, Univeristé des Sciences, des Techniques et de Medecine de Tunis, Mars 1993.

[7] A. M. MESLEH, CHI Square Feature Extraction Based SVMs Arabic Language Text Categorization System, *Proceedings of the 2nd International Conference on Software and Data Technologies, (Knowledge Engineering)*, Vol. 1, Barcelona, Spain, July, 22–25, 2007, pp. 235–240.

[8] M. ELKOURDI, A. BENSAID AND T. RACHIDI, Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm, *Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Scriptbased Languages*, Geneva, August 23rd–27th , 2004, pp. 51–58.

[9] R. AL-SHALABI, G. KANAAN, M. GHARAIBEH, Arabic Text Categorization Using kNN Algorithm, *Proceedings of The 4th International Multiconference on Computer Science and Information Technology*, Vol. 4, Amman, Jordan, April 5–7, 2006.

[10] H. SAWAF, J. ZAPLO AND H. NEY, Statistical Classification Methods for Arabic News Articles, Paper presented at the *Arabic Natural Language Processing Workshop (ACL2001)*, Toulouse, France.

[11] RAMI AYADI, MOHSEN MARAOUI, MOUNIR ZRIGUI, Intertextual distance for Arabic texts classification, *ICITST* 2009: 1–6.

[12] R. M. DUWAIRI, Machine Learning for Arabic Text Categorization, *Journal of American society for Information Science and Technology*, Vol. 57, No. 8, 2006, pp. 1005–1010.

[13] M. SYIAM, Z. FAYED AND M. HABIB, An Intelligent System for Arabic Text Categorization, *International Journal of Intelligent Computing and Information Sciences*, Vol. 6, No. 1, 2006, pp. 1–19.

[14] M. BENKHALIFA, A. MOURADI AND H. BOUYAKHF, Integrating WordNet knowledge to supplement training data in semi-supervised agglomerative hierarchical clustering for text categorization, *International Journal of Intelligent Systems*, Vol. 16, No. 8, 2001, pp. 929–947.

[15] G. SALTON, A. WONG AND C. S. YANG, A Vector Space Model for Automatic Indexing, *Communications of the ACM*, 18(11), pp. 613–620, 1975.

[16] D. BLEI AND J. LAFFERTY, Correlated topic models, in Y. Weiss, B. Schölkopf, and J. Platt editors, *Advances in Neural Information Processing Systems 18*, MIT Press, Cambridge, MA. 2006.

[17] D. BLEI, A. NG AND M. JORDAN, Latent Dirichlet Allocation, *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003.

[18] LAROUSSI MERHBENE, ANIS ZOUAGHI, MOUNIR ZRIGUI, Arabic Word Sense Disambiguation. *ICAART* (1) 2010, pp. 652–655.

[19] T. JOACHIMS, Text categorization with Support Vector Machines: learning with man y relevant features, in *Proceedings of the European Conference on Machine Learning (ECML'98)*, Berlin, pp. 137–142, Springer (1998).

[20] V. N. VAPNIK, *Statistical Learning Theory*, John Wiley, New York (1998).

[21] MOHSEN MARAOUI, GEORGES ANTONIADIS, MOUNIR ZRIGUI, SALA: Call System for Arabic Based on NLP Tools, *IC-AI* 2009, pp. 168–172.

[22] JOACHIMS, Making large-scale SVM learning practical, in *Advances in Kernel Methods – Support Vector Learning*, M.I.T. Press, 1999.

[23] `http://gibbslda.sourceforge.net/.`

[24] R. BAEZA-YATES, B. RIBEIRO-NETO, *Modern Information Retrieval*, Addison Wesley & ACM Press, New York (1999).

[25] M. ZRIGUI, *Contribution au traitement automatique de l'arabe*, HDR en Informatique, Grenoble 3, France, 2008.

[26] `http://www.qamus.org\morphology.html`

[27] SALEH M. AL-SALEEM, Associative Classification to Categorize Arabic Data Sets, *The International Journal of ACM Jordan* (ISSN 2078-7952), Vol. 1, No. 3, September 2010.

[28] F. HARRAG, E. EL-QAWASMEH, P. PICHAPPAN, Improving Arabic text categorization using decision trees, *The First International Conference on Networked Digital Technologies NDT '09*, pp. 110–115, 2009.

[29] MOHSEN MARAOUI, GEORGES ANTONIADIS, MOUNIR ZRIGUI, CALL System for Arabic Based on Natural Language Processing Tools, *IICAI* 2009, pp. 2249–2258.

[30] FADI THABTAH, WA'EL MUSA HADI, GAITH ALSHAMMARE, VSMs with K-Nearest Neighbour to Categorise Arabic Text Data, *Proceedings of the World Congress on Engineering and Computer Science 2008 WCECS 2008*, October 22–24, 2008, San Francisco, USA.

[31] G. KRISHNALAL, S. BABU RENGARAJAN, K. G. SRINIVASAGAN, A New Text Mining Approach Based on HMM-SVM for Web News Classification, *International Journal of Computer Applications* (0975 – 8887), Vol. 1, No. 19, 2010.

[32] M. ALJLAYL AND O. FRIEDER, On Arabic search: improving the retrieval effectiveness via a light temming approach, in *ACM CIKM 2002 International Conference on Information and Knowledge Management*, McLean, VA, USA, pp. 340–347, 2002.

[33] ANIS ZOUAGHI, MOUNIR ZRIGUI, GEORGES ANTONIADIS, Automatic Understanding of Spontaneous Arabic Speech – A Numerical Model, *TAL* 49(1), pp. 141–166, (2008).

[34] MOUNIR ZRIGUI, MBARKI CHAHAD, ANIS ZOUAGHI, MOHSEN MARAOUI, A Framework of Indexation and Document Video Retrieval based on the Conceptual Graphs, *CIT* 18(2), (2010).

[35] RAMI AYADI, MOHSEN MARAOUI, MOUNIR ZRIGUI A Multilingual System for the Blind, *ICTA* 2009, pp. 111–116.

[36] MOURAD MARS, GEORGES ANTONIADIS, MOUNIR ZRIGUI, Statistical Part of Speech Tagger for Arabic Language, *IC-AI* 2010, pp. 894–899.

[37] RAMI AYADI, MOHSEN MARAOUI, MOUNIR ZRIGUI, SCAT: A system of classification for Arabic texts, *IJITST*, Vol. 3/1, 2011.

[38] LAROUSSI MERHBENE, ANIS ZOUAGHI, MOUNIR ZRIGUI, Ambiguous Arabic Words Disambiguation, *SNPD* 2010, pp. 157–164.

Received: January, 2010
Revised: July, 2012
Accepted: July, 2012

Contact addresses:
Mounir Zrigui
LaTICE Laboratory (Research Unit of Monastir)
University of Monastir, Tunisia
e-mail: mounir.zrigui@fsm.rnu.tn

Rami Ayadi
Faculty of Economics and Management
University of Sfax, Tunisia
e-mail: ayadi.rami@planet.tn

Mourad Mars
Stendhal University
Grenoble, France
e-mail: mourad.mars@e.u-grenoble3.fr

Mohsen Maraoui
University of Monastir, Tunisia
e-mail: maraoui.mohsen@gmail.com

MOUNIR ZRIGUI received his PhD from the Paul Sabatier University, Toulouse, France in 1987 and his HDR from the Stendhal University, Grenoble, France in 2008. Since 1986, he was a Computer Sciences Professor at Brest University, France, and after at the Faculty of Science of Monastir, Tunisia. He started his research, focused on all aspects of automatic processing of natural language (written and oral), in RIADI laboratory and continued it in LaTICE Laboratory. He has run many research projects and published many research papers in reputed international journals/conferences. Currently, he is the director of virtual teaching at the University of Monastir, Tunisia.

RAMI AYADI graduated in Computer Sciences and Multimedia from the University of Sfax 'ISIMSF', Tunisia, in 2009. In 2010, he worked in the research unit of Technologies of Information and Communication, Higher School of Sciences and Technologies of Tunis, Tunisia, as a permanent researcher. Currently, he is pursuing his PhD in Computer Sciences at the Faculty of Economics and Management of Sfax, under the guidance of Dr. Mounir Zrigui. His main research interests are in classification and indexing of textual documents in Arabic language on the web, and developing systems to facilitate the processing of digital data on internet.

MOURAD MARS is a PhD student at the Stendhal University, Grenoble, since 2007. His main research interests are in all aspects of e-learning tools for Arabic language. He has published many research papers in reputed international journals/conferences.

MOHSEN MARAOUI received his PhD degree in Computer Science from the Stendhal University, Grenoble, France in 2010. Currently, he is an assistant professor at the University of Monastir, Tunisia. His research is focused on NLP. He has published many research papers in reputed international journals/conferences.