Volume 4, Issue 1, January 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Text Clustring with Fuzzy Measure of Descriptors Weight

Ibtissam El Hassani

Moulay Ismail University, Doctoral Studies Center,
ENSAM Meknes, Morocco

Abstract— Our work consists in implementing a new two-dimensional descriptor in Text Mining. After the morphosyntaxic analysis of the words using the techniques of automatic treatment of the natural language, there is lost additional information which we will not neglect but rather put in a new dimension. This involves a rewriting of weight descriptors in documents by a new "fuzzy" measure. The application of this approach on an Arabic corpus involved a transformation of text words in a set of pairs (root, pattern) to be descriptors of our corpus. The morphosyntactic analysis gives all possibilities and not a single solution. We apply, then the Hidden Markov model morphosyntaxic post-analysis to detect the most likely based on the context of the word analysis. We show that we are able to achieve higher precision when compared to conventional Vector Space Model representation and Latent Semantic analysis in the context of Arabic Text Clustering.

Keywords— Text Mining, Text Processing , Hidden Markov models , Natural language Processing, Fuzzy logic

I. INTRODUCTION

In Text Mining, the pre-treatment of documents gives formal representation as a matrix. This representation assumes in advance that the descriptors are completely independent, actually we need this property in order to measure the distance or similarity between two vectors. This implies, therefore, that the vectors are written in a linearly dependent base. However, descriptors are usually not semantically independent, sometimes this dependence is both syntactic and semantic. There are a few treatments that can partially solve this problem as stemming which can be used to achieve a rapid rapprochement of the word root. The Arabic words, for example, "research" (بحث) and "we are searching" (نبحث) are considered equal semantically taking root as descriptors "research" (بحث) . But dependence is not always binary and we cannot always considered words completely equal if they are derived from the same root, as they cannot be considered completely independent. We want, then, consider this "fuzzy" dependence between descriptors and thus represent the corpus in a new linearly independent base. We thereafter, measure a score that indicates the semantic similarity staking into account dependencies between descriptors. In Arabic context, this will take us to define a new form of bidimentionnels descriptors: the root, and the degree of deviation from this root (pattern).

We will present in the following paragraph related work namely Latent Semantic Analysis. The third paragraph presents the general model without determining a particular language. In fact, the model can be applied to any language and even any data vectorially represented being textual or not, as long as we are able to measure a fuzzy dependency between descriptors. In the fourth one, we model the structure of the Arabic language to automatically measure a fuzzy dependency based on the Techniques of Automatic Natural Language Processing. Finally, we give an example of implementation in the context of Arabic language and experiences conducted on a corpus extracted from Wikipedia.

II. RELATED WORKS

In Text mining, we are necessarily led to find similarities between text segments. The typical approach is to use a simple method of lexical conciliation, and produce a similarity score based on the number of tokens that occur in both input segments. Effective to some extent, these methods of lexical similarity cannot always take into consideration the semantic similarity of texts. For example, there is an obvious similarity between the segments of text that shares the same meaning without one word in common, or that contain words syntactically and, on the result, semantically dependent. Most existing similarity measures documents fail to consider any type of connection between these texts because these words will be considered as totally independent, which means that the two descriptors in the vector representation of the corpus are considered in the direction orthogonal mathematics. Latent Semantic Analysis (LSA) is one of the best known methods that offer a solution to this problem. The LSA method is based on the fact that words that appear in the same context are semantically close. It consist to a matrix dimension reduction managed by a singular value decomposition (SVD). The principle of the approach is therefore to reduce the dimension by projecting the components of the original matrix into a reduced vector space. It concise to reduce the dimensionality by Singular Value Decomposition (SVD) : the Salton matrix $A = [aij]$ where aij is the frequency of occurrence of the word i in the context j, is decomposed into a product of three matrices $U\Sigma V'$ where A and V are orthogonal matrices and Σ is a diagonal matrix.

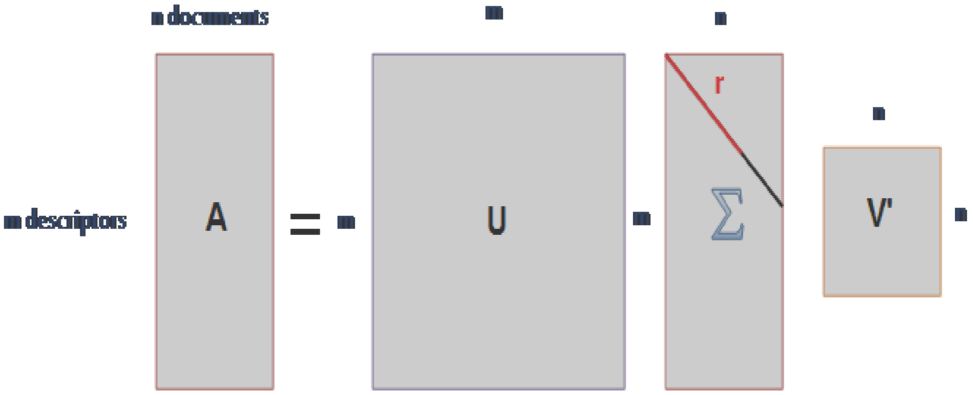

Fig. 1 Singular Value Decomposition (SVD). r is the rank of the matrix A

The LSA approach is due to laboratories BellCore. Originally in 1989, this analysis was use for Information Retrieval [1]. Over time, its use has spread to more diverse areas such as information filtering [2], the automatic evaluation [3], [4], [5]. In the psycholinguistic field through modeling acquisition [6], learning [7]. The principle of the approach is to reduce the projecting dimension of the original matrix components in a vector space reduced. In previous work, we have presented each descriptor by a discrete signal giving different degrees of dependence between descriptors, taking advantage of the Multi Resolution Analysis properties of the Wavelet Transform [8]. Our approach to solve the problem is different, it is not based on an algebraic transformation of the matrix of Salton, but rather on similarity measure between descriptors independent of the corpus, then we group descriptors into classes and reduce the size of the matrix by introducing a fuzzy measure presence. We have validate the relevance of the approach in Information Retrieval in the context of Arabic texts [9], and this work show its validity in text Clustering.

III. Measure of Fuzzy Presence of Descriptors

A. The measurement of the Dependence Between Descriptors

A given descriptor or a "unit of sense" may appear in different syntactic or semantic forms. These forms are not exactly similar but with some dependence. Let $\mathfrak{R}$ be a function that associates to each pair of descriptors a value in [0,1], which measures the degree of dependence or similarity. Once $\mathfrak{R}$ defined, we collect descriptors having a dependency nonzero in classes: a class is defined as a set of descriptors that have the same representative and have a non-zero $\mathfrak{R}$. For each class we choose a representative as shown in the following table:

TABLE I

Example of Documents with Appearance of Dependent Descriptors

Desc	Rep	$\mathfrak{R}$	Doc1	Doc2	Doc3
desc 1	rep1	$\mathfrak{R}(\text{desc1}, \text{rep1})$	1	0	0
desc 2		$\mathfrak{R}(\text{desc2}, \text{rep1})$	0	2	0
desc 3		$\mathfrak{R}(\text{desc3}, \text{rep1})$	0	1	0
desc 4		$\mathfrak{R}(\text{desc4}, \text{rep1})$	0	0	1
desc 5	rep2	$\mathfrak{R}(\text{desc5}, \text{rep2})$	1	0	0
desc 6		$\mathfrak{R}(\text{desc6}, \text{rep2})$	0	1	0
desc 7		$\mathfrak{R}(\text{desc7}, \text{rep2})$	0	0	3
desc 8		$\mathfrak{R}(\text{desc8}, \text{rep2})$	0	0	1

Words $(desc\ i)_{1 \leq i \leq 4}$ are dependents and $rep1$ is the representative of their class. Words $(desci)_{5 \leq i \leq 8}$ are dependent and $rep2$ is the representative of their class. Note that $\mathfrak{R}$ may be imposed or obtained by learning.

B. The measurement of the Dependence Between Document and Descriptor

From the vector representation of a corpus we can measure the distance between two documents by conventional distances on vector spaces. In this representation, noted Salton Matrix [1] [2], the lines are the descriptors and columns are the documents. The elements of the matrix ω_{ij} are a measure of the dependence between descriptors and documents. ω_{ij} are based on the occurrence tf_{ij} (*term-frequency*) of a descriptor i in document j, which is simply the number of occurrences. In our model , we gather descriptors belonging to the same class and we keep only one representative . Each descriptor has a dependency with the representative of its class denoted $\mathfrak{R}$ taking values between 0 and 1. Thereafter , the number of occurrences that measure the presence in the corpus will be replaced by a "fuzzy occurrence" . Whenever a descriptor appears the number of occurrence of the representative document is incremented by its measure of dependence with the representative. Example: Let a class C which consists of three elements $\{\text{desc1}, \text{desc2}, \text{desc3}\}$ whose representative is "rep". The descriptor "desc1" appears twice in a corpus, "desc2" three times and "desc3" once. Let $\mathfrak{R}_1 = \mathfrak{R}(\text{desc1}, \text{rep})$, $\mathfrak{R}_2 = \mathfrak{R}(\text{desc1}, \text{rep})$ et $\mathfrak{R}_3 = \mathfrak{R}(\text{desc1}, \text{rep})$. To estimate the "occurrence" of "rep" we can say that if "desc1" appears 1 times then "rep" appears $\mathfrak{R}_1$ times, $\mathfrak{R}_1$ is a real number measuring a fuzzy appearance. If

" desc1" appears n times then "rep" appears $n\Re_1$ times. Then the frequency of the representative in the corpus would be $2\Re_1 + 3\Re_1 + \Re_1$wich we name ftf_{ij} (*fuzzy-term-frequency*).

C. Fuzzy Term Frequency of a class representative ftf

Frequency $tf_{desc,doc}$ of descriptor $desc$ measures the number of occurrences of descriptor in the document. It is replaced, in our model, by a new measure that takes into account the presence of fuzzy descriptor, which we call $ftf_{rep,doc}$ (fuzzy term frequency). We formalize what we explained in the previous paragraph by the following equation:

$$ftf_{rep,doc} = \sum_{x \epsilon \, classe \, i} \mathrm{tf}_{\mathrm{rep},doc} \; \Re \; (x, rep_i) \qquad (1)$$

With ftf (fuzzy-term frequency) is the frequency of fuzzy descriptors explained above

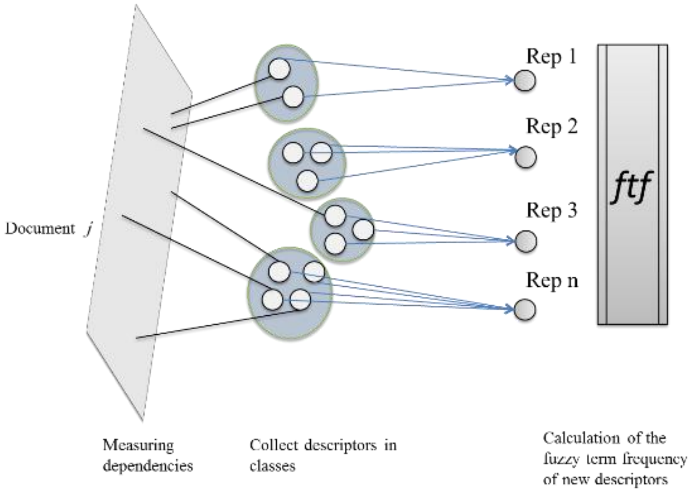
Rep 1
Rep 2
Rep 3
Rep n
ftf
Document j
Measuring dependencies
Collect descriptors in classes
Calculation of the fuzzy term frequency of new descriptors

Fig. 2 Diagram explaining the calculation of the ftf

This will allow us a considerable matrix size reduction. However, this reduction will not be "blind" because the words do not have the same weight, but a fuzzy measure. The calculation of the new weights representatives of classes of descriptors is then performed via the formulas that will be exhibited in the following. Example: Taking the example in Table 1, we have two classes. We keep only the representatives of the two classes in the Salton matrix and calculate its *ftf* by formula (1). Suppose now that the dependence measures are: $\Re(\mathrm{mot1}, \mathrm{rep1}) = 0.5$ *;* $\Re(\mathrm{mot2}, \mathrm{rep1}) = 0.9$ *;* $\Re(\mathrm{mot3}, \mathrm{rep1}) = 0.4$ *;* $\Re(\mathrm{mot4}, \mathrm{rep1}) = 0.7$ *;* $\Re(\mathrm{mot5}, \mathrm{rep2}) = 1$ *;* $\Re(\mathrm{mot6}, \mathrm{rep2}) = 0.7$ *;* $\Re(\mathrm{mot7}, \mathrm{rep2}) = 0.6$ *;* $\Re(\mathrm{mot8}, \mathrm{rep2}) =$ *1.*

TABLE II
EXAMPLE OF MEASURING DEPENDENCY

Rep	Doc1	Doc2	Doc3
rep1	$\Re(\mathrm{desc1}, \mathrm{rep1})=$ 0.5	$2 \times \Re(\mathrm{desc2}, \mathrm{rep1})$ $+ \Re(\mathrm{desc3}, \mathrm{rep1}) =$ $2 \times 0.9 + 0.4 = 2.2$	$\Re(\mathrm{desc4}, \mathrm{rep1}) = 0.7$
rep2	$\Re(\mathrm{desc5}, \mathrm{rep2}) = 1$	$\Re(\mathrm{desc6}, \mathrm{rep2}) = 0.7$	$3 \times \Re(\mathrm{desc7}, \mathrm{rep2})$ $+ \Re(\mathrm{desc8}, \mathrm{rep2}) =$ $3 \times 0.6 + 1 = 2.8$

We note that there's not only a reduction of the size of matrix but also a similarity that has just appeared, since documents share now the same descriptors. However, in Table I there was no similarity between the documents as they had completely different descriptors. We also note that this reduction is not "blind", but it considers the fuzzy degree of occurrence of representative of each class in the corpus.

D. Fuzzy Weight of Descriptor

There are several ways to calculate the weight of the descriptors in the corpus. We reformulate in the following these measures taking into account the new representation explained.

- Fuzzy Frequency : It does not only consider the frequency of occurrence of a descriptor in a document but takes into account the different existing dependencies between descriptors . The conventional formula is: $w_{desc,doc} = \frac{tf_{desc,doc}}{\sqrt{\sum_{k=1}^{Nbmotsdoc} tf_{desc,k}}}$ with $Nbmotsdoc$ the number of distinct descriptors in the document doc. It becomes, then, in this new representation $w_{rep,doc} = \frac{ftf_{rep,doc}}{\sqrt{\sum_{k=1}^{Nbrepdoc} ftf_{desc,k}}}$. We replace $Nbmotsdoc$ by $epdoc$, which is equal to the number of representatives of different classes. According to the given definition of ftf the equation becomes:

$$w_{doc,rep} = \frac{\sum_{x \, \epsilon \, classe \; i} tf_{x,doc} \; \Re\,(x, rep_i)}{\sqrt{\sum_{k=1}^{Nbrepdoc} \; \sum_{x \, \epsilon \, classe \; i} tf_{x,doc} \; \Re\,(x, rep_i)}} \qquad (2)$$

- $fTfidf$: The most common descriptors are not necessarily the most informative . The Tfidf is used, then , to distinguish between different documents and when a descriptor is present throughout this measure is 0. We rewrote the formula by introducing the new concept of fuzzy descriptor presence and we called $fTfidf$ (fuzzy term frequency and inversed paper frequency) . The conventional Tfidf is defined by $\mathrm{w}_{\mathrm{doc,desc}} = \mathrm{Tfidf(doc, desc)} = \mathrm{tf(doc, desc)} \times \log\left(\frac{|\mathrm{Doc}|}{\mathrm{df(desc)}}\right)$, it becomes in this new representation :

$$\mathrm{w}_{\mathrm{desc,doc}} = fTfidf(\mathrm{doc, desc}) = ftf(\mathrm{doc, desc}) \times \log\left(\frac{|\mathrm{Doc}|}{\mathrm{df(desc)}}\right)$$
$$= \sum_{x \,\epsilon\, classe\ \ i} \mathrm{tf}_{\mathrm{doc},x}\, \mathfrak{R}\ (x, rep_i) \times \log(\frac{|\mathrm{Doc}|}{\mathrm{df(desc)}} \qquad (3)$$

- *The* "fuzzy" Entropy :The entropy measure the dispersion of a descriptor in a corpus. The entropy S of a descriptor is given by the formula $\mathrm{E(desc)} = \sum_{doc} \frac{tf_{desc,doc}\, log_2(\frac{tf_{desc,doc}}{gf_{desc}})}{gf_{desc}\, log_2(N)}$, it is in our model:

$$S(rep) = \sum_{doc} \frac{ftf_{rep,doc}\, log_2(\frac{ftf_{rep,doc}}{gf_{rep}})}{gf_{rep}\, log_2(N)} \qquad (4)$$

Where $\mathrm{gf}_{\mathrm{rep}}$ represents the total number of times the descriptor rep appears in the corpus of N documents. A calculation of weight is then given by the formula :

$$w_{\mathrm{rep,doc}} = (1 + E(rep))log(ftf_{\mathrm{rep,doc}} + 1) \qquad (5)$$

E. Measures of similarity

Measures of similarity or distance involved in all processing tasks in Text Mining. In this new representation of the corpus, the fundamental formulas distances do not change course. As against the dimension and weight vectors change, and then measuring of distance or similarity which directly affects the final result.

- Euclidean distance : The distance between the documents doc_a and doc_b represented by their vectors is given by: $D_E(\overrightarrow{doc_a}, \overrightarrow{doc_b}) = \left(\sum_{desc=1}^{m} \left|w_{desc,a} - w_{desc,b}\right|^2\right)^{\frac{1}{2}}$. Where the set of descriptors is $Desc = \{desc_1, \ldots, desc_m\}\}$ and $w_{desc,\mathrm{doc}}$ is the weight of the descriptor $desc$ in document doc.
- The cosine similarity measure : The cosine of the angle between the vectors quantifies the similarity of the two documents. The cosine between documents doc_a and doc_b represented by their vectors is defined by: $\mathrm{SIM_c}(\overrightarrow{\mathrm{doc_a}}, \overrightarrow{\mathrm{doc_b}}) = \frac{\overrightarrow{\mathrm{doc_a}} . \overrightarrow{\mathrm{doc_b}}}{|\overrightarrow{\mathrm{doc_a}}| \times |\overrightarrow{\mathrm{doc_b}}|}$.
- Jaccard coefficient : It is a measure of similarity and varies between 0 and 1. It is 1 when $\overrightarrow{doc_a} = \overrightarrow{doc_b}$ and 0 when disconnected. The value of 1 means that the two objects are the same and 0 means that they are completely different.

F. Independence to language

The model we have presented independent to language. It can be used once we come to measure a dependence between descriptors of a corpus. Arabic as the language of our application provides an opportunity to automatically calculated syntactic dependency, because its structure itself is based on a root with a unit of meaning and a pattern having a degree of differentiation relative to this root. In what follows we present the structure of the Arabic language for such a measure of dependence, we need also tools for Natural Language Processing to extract roots and possible patterns. We implemented then a Hidden Markov Model to identify roots and patterns most probable, to define an automatic measurement of dependence between descriptors $\mathfrak{R}$.

IV. IMPLEMENTATION IN ARABIC CONTEXT

A. Implementation in the Arab context

The Arabic is an inflectional language. The derivation in Arabic is based on morphological patterns and the verb plays a greater inflectional role than in other languages. Furthermore, Arabic words are built-up from "roots" representing lexical and semantic connecting elements.

Let a, b be two words. The morphosyntactic analysis of a, b gives respectively the couples $\{x_1, y_1\}, \{x_2, y_2\} \in R_{corpus} \times S$ where R_{corpus} is the set of roots (جذور) in the corpus and S is the set of patterns (اوزان). We define a function that measures the dependence between these two words $f(\{x_1|y_1\}, \{x_2|y_2\}) = \mathbb{I}_{x_1=x_2} \times \psi(y_1|y_2)$ with $\mathbb{I}_{\mathrm{x_1=x_2}}$ is the function that gives 1 if $x_1 = x_2$ and 0 otherwise. And $\psi(y_1, y_2)$ measures the semantic relation between the patterns $\mathrm{y_1, y_2}$. Every two patterns $\mathrm{y_i, y_j}$ have indeed a certain dependency $\psi_{ij} = \psi(\mathrm{y_i, y_j}) \in [0,1]$. We can propose an automatic estimation using the difference between n_i, n_j the number of letters of prefixes and infix (without suffixes) between the two patterns [11] :

$$\mathfrak{R}(\mathrm{a, b}) = \psi_{\mathrm{ij}} = \frac{1}{1 + |\mathrm{n_i} - \mathrm{n_j}|} \qquad (1)$$

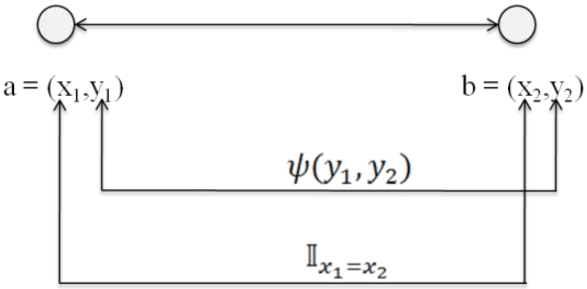

Fig. 3 Diagram Showing the Measurement of the Dependency between two Arabic Words

This function satisfied the properties we need, namely $\psi_{ij} \in [0,1]$. If $\{x_1|y_1\} = \{x_2|y_2\} \Rightarrow f(a,b) = \psi_{ij} = 1$. Several other functions can also be proposed.

Example : Let us consider three documents which appear Arabic words with a syntactic and semantic dependency:

- الاستماع الى الدرس يساعد على فهمه.
- سمعت تقريرا عن دراسة تقنيات سماع نبضات الجنين.
- استمعت ودرست بتركيز ملخص دراسته.

Note that, in the vector representation, the documents were considered different because they did not share any common descriptor. And if we keep only the roots, they will then be considered completely equal and dependent and this is not the case. So we do a morphosyntactic analysis that will allow us to define classes of dependence with a single representative which is the root. It will also allow defining a measure of dependence between words.

TABLE III
AN EXAMPLE OF DEPENDENCY MEASURE OF DESCRIPTORS IN THREE ARABIC DOCUMENTS

Desc	Root	Pattern S	Doc1	Doc2	Doc3
الاستماع	سمع	افْتِعَال	1	0	0
سمعت	سمع	فَعِلَت	0	1	0
سماع	سمع	فَعَال	0	1	0
استمعت	سمع	افْتَعَلْت	0	0	1
الدرس	درس	فَعْل	1	0	0
دراسة	درس	فَعَالَة	0	1	0
ودرست	درس	فَعَلْتُ	0	0	1
دراسته	درس	فَعَلْتُ	0	0	1

With the current representation of descriptors no similarity between each couple of documents will be detected, because they have no term in common. We will regroup the descriptors that have a dependency $\Re \in]0,1]$ and for each class we selected a representative which is the root of words of each cluster.

TABLE IV: GROUPING DESCRIPTORS THAT HAVE A DEPENDENCY $\Re \in]0,1]$

representative	Document 1	Document 2	Document 3

rep1 = سمع	ℜ(سمع, الاستماع	ℜ(سمع, سماع)
		ℜ(سمع, اسمعت)
			ℜ(سمع, استمعت

rep1 = در	ℜ(درس, الدرس)	ℜ(درس, دراسة)	ℜ(درس, دراسته ℜ(درست, درس)

We then reduced the descriptor space since each cluster is represented by the "root", and calculate ftf_{ij} (fuzzy term frequency of descriptor j in document i) :

TABLE V: CALCULATION OF FUZZY TERM FREQUENCY

Rep	Doc1	Doc2	Doc3
سمع	0.5	$0.9 - 0.4 = 1.3$	0.7
درس	1	0.7	$0.6 - 1 = 1.6$

B. Corpus Representation

We build an Arabic corpus from Wikipedia we named WikipediaArabia2012. The size of the extracted corpus is 4856 documents, distributed over six topics, in this case:

- Class A : Engineering هندسة تطبيقية ;
- Class B : Philosophy of science فلسفة العلوم ;
- Class C : Sociology علم الاجتماع ;
- Class D : Mathematics رياضيات ;
- Class E : Artificial intelligence ذكاء اصطناعي ;
- Class F : Economy اقتصاد .

The corpus was divided into two subsets of documents. Where 90% of the corpus was dedicated to training and 10% of the overall documents formed the evaluation corpus.

Text pre–processing is the first step in a Text Classification. It aims to reduce the noise in documents by removing all the unnecessary terms and mistyped words along with transforming documents content from a plain text to a suitable form that can be easily handled by automated programs. The most important text pre-processing operations are:

1) Documents encoding unification: the encoding unification process ensures the same encoding for all the documents in the document collection. In our work we adopted the UTF–8 character set, which supports the characters of the Arabic language.

2) Documents normalisation: suppression of symbols, numbers, markers, special characters, etc.

3) Normalization of certain Arabic characters: a/ Removal of diacritics : We have removed the following vowels: *Fatha, Damma, Kasra, Sukun, Shadda, double Fatha, double Damma,* and double *Kasra*. b/ Removal of *Tatweel* (Elongation of letters). c/ Normalization of *Hamza*: The following letters are converted to *Alef* by systematically removing the *Hamza* (*Alef Madda, Alef Hamza Aabove, Below Alef Hamza, Hamza Aabove, and Below Hamza*).

4) Stems extraction: In our work we used the Alkhalil [12] morphological analyser to generate a list of stems for each document. Alkhalil analysis each word in the documents and returns among other morphological information the word's possibly related stems, roots and patterns. We have also implemented a Viterbi algorithm [13] to select, exclusively the stems that are relevant to the context [9].

5) Stop words elimination: elimination of noisy words by comparing each word with the elements of a handmade list of noisy words including: prepositions, demonstrative pronouns, identifiers, logical connectors, etc. Stop words do not carry any useful information and therefore their removal will not affect our process.

In order to evaluate our model based on measure of fuzzy term frequency, we compare our model to the conventional Vector Space Model and Latent Semantic Analysis in the task of text clasturing in Arabic context. the results are shown in Table VI and figure 4.

C. Results

We use three standard indicators: precision, recall and F–score.

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$F - score = \frac{(1+\beta^2)Precisio\ n \times Recall}{\beta^2 Precision\ + Recall} \tag{4}$$

The parameter β is set to 1 to provide the same importance to the recall and precision. The following table illustrates the four categories TP, TN, FP and FN.

TABLE VI

EVALUATION OF TEXT CLASSIFICATION

	Real class	Other classes
Predicted class	TP	FN
Other classes predicted	FP	TN

TABLE VII

PERFORMANCE COMPARISON OF FUZZY MEASURE MODEL (FTFIDF) ON WIKIPEDIAARABIA2012 DATA

% of corpus	**Classe**	**Recall**			**Precision**			**F-measure**		
		tfidf	***LSA***	***ftfidf***	***tfidf***	***LSA***	***ftfidf***	***tfidf***	***LSA***	***ftfidf***
30%	A	89,00%	93,00%	89,00%	70,08%	93,00%	92,00%	78,41%	93,00%	90,48%
	B	45,00%	61,00%	60,00%	68,18%	61,00%	87,84%	54,22%	61,00%	71,30%
	C	81,19%	90,10%	85,13%	57,34%	90,10%	85,60%	67,21%	90,10%	85,36%
	D	77,00%	89,00%	85,00%	85,56%	91,75%	80,91%	81,05%	90,35%	82,90%
	E	78,00%	87,00%	97,00%	87,64%	90,62%	86,59%	82,54%	88,77%	91,50%
	F	73,53%	81,37%	74,51%	85,23%	81,37%	96,08%	78,95%	81,37%	83,93%

50%	A	85,00%	85,00%	87,00%	87,63%	85,00%	87,00%	86,29%	85,00%	87,00%
	B	66,00%	64,00%	65,00%	82,50%	64,00%	84,19%	73,33%	64,00%	73,36%
	C	66,34%	65,35%	75,26%	87,01%	65,35%	80,36%	75,28%	65,35%	77,73%
	D	90,00%	93,00%	89,00%	74,38%	75,61%	80,91%	81,45%	83,41%	84,76%
	E	78,00%	94,00%	96,00%	80,34%	80,34%	80,00%	79,15%	86,63%	87,27%
	F	91,18%	90,20%	93,14%	83,78%	90,20%	93,00%	87,32%	90,20%	93,07%
70%	A	82,00%	85,00%	89,00%	86,32%	85,00%	89,00%	84,10%	85,00%	89,00%
	B	65,00%	65,00%	65,00%	85,53%	65,00%	83,84%	73,87%	65,00%	73,23%
	C	73,27%	72,28%	78,28%	86,05%	72,28%	85,76%	79,15%	72,28%	81,85%
	D	88,00%	93,00%	87,00%	75,86%	77,50%	82,41%	81,48%	84,55%	84,64%
	E	95,00%	93,00%	97,00%	77,24%	82,30%	82,91%	85,20%	87,32%	89,40%
	F	90,20%	91,18%	93,14%	85,98%	91,18%	93,14%	88,04%	91,18%	93,14%
100%	A	85,00%	82,00%	83,00%	88,54%	82,00%	91,00%	86,73%	82,00%	86,82%
	B	65,00%	65,00%	65,00%	84,42%	65,00%	80,94%	73,45%	65,00%	72,10%
	C	72,28%	73,27%	78,60%	84,88%	73,27%	88,76%	78,07%	73,27%	83,37%
	D	93,00%	88,00%	82,00%	76,23%	75,86%	90,11%	83,78%	81,48%	85,86%
	E	93,00%	95,00%	71,00%	82,30%	77,24%	82,91%	87,32%	85,20%	76,49%
	F	91,18%	90,20%	92,00%	85,32%	90,20%	84,51%	88,15%	90,20%	88,10%

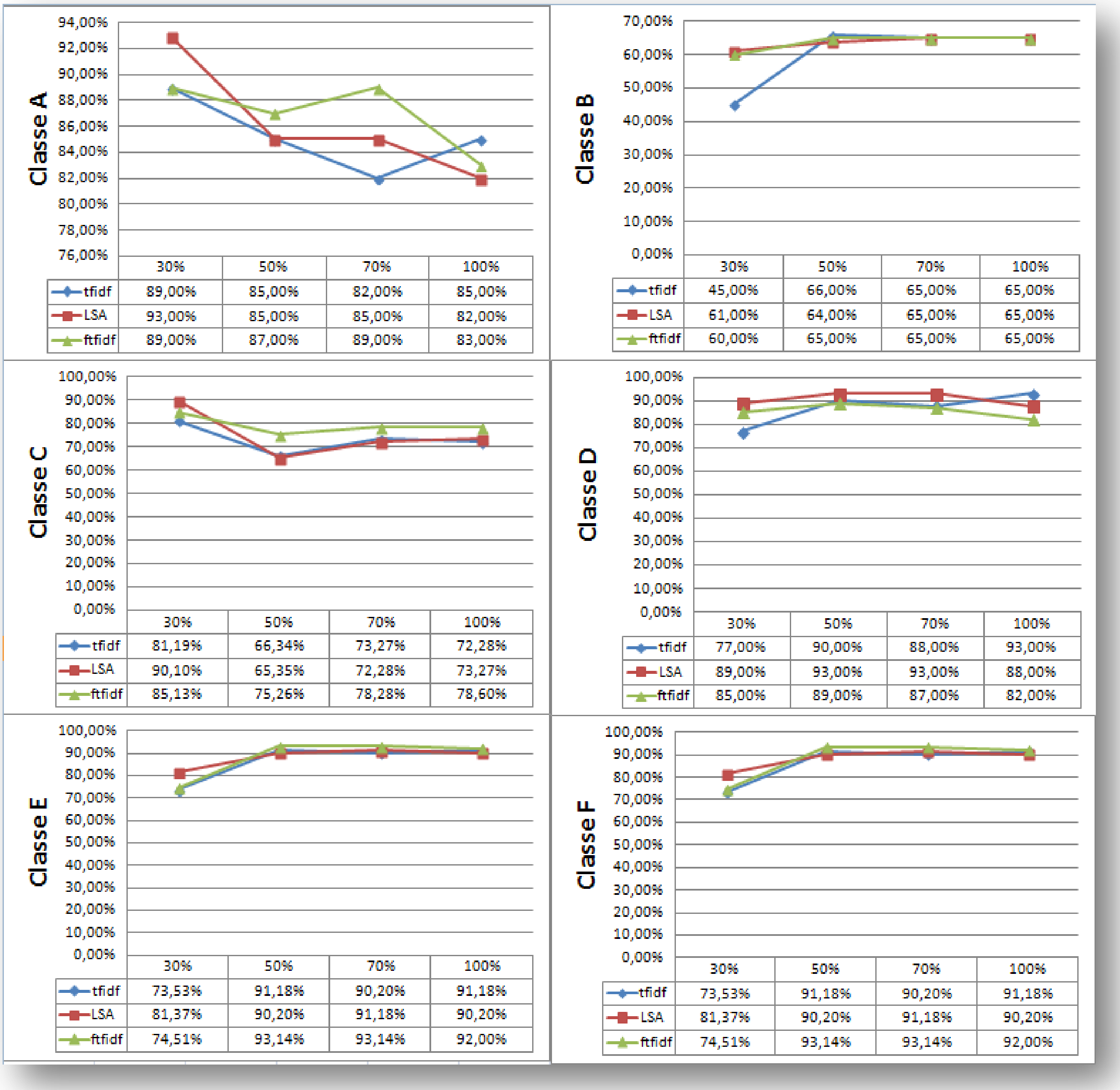

Fig. 4 F-measure Comparison of Fuzzy Measure Model ($ftfidf$) on WikipediaArabia2012 data

V. CONCLUSION

We have combine a few approaches : Hidden Markov Model, adding the Arabic "patterns"' dimensions to the "roots" dimensions to come up with a fuzzy measure of descriptor's presence. We presented a new method to measure the relationship between descriptors in Text Mining based on a notion of "fuzzy measure of the presence" and we adapted the traditional statistical measures to this context. Experiments conducted on Arabic scientific corpus shows that average F-measure score higher in the task of Arabic document clustering, namely 83.86 %, when compared to Latent Semantic Analysis (82.46%) and Vector Space Model (79.77%). However, our method could be computationally expensive. Indeed we use, in Arabic context, a measure based on the results of the automatic processing of natural language yet Natural Language Process operates in the order of a few words per second. It remains a challenge to see how the spectral semantic representation can be made much more efficient for very large text corpora. For future work, our proposed technique could possibly be applied to other languages by defining a quantitative measure of similarity between two descriptors. It could also be applied to other types of Text Mining tasks such as selection of concepts taking into account the descriptors dependency.

REFERENCES

[1] S. Deerwester, G. Dumais, T. Launder and R. Harshmann, "Indexing by Latent Semantic Analysis," in *Dans les actes de Journal of the American Society for Information Science*, 1990.

[2] P. W. Foltz and S. T. Dumais, "Personalized information delivery : an analysis of information filtering methods," in *Communications of the ACM*, 1992.

[3] P. W. Foltz., "Latent Semantic Analysis for Text- Based Research," in *Behavior research methods instruments and computers*, 1996.

[4] M. Schreiner, B. Rehder, D. Laham, P. W. Foltz, W. Kintsch, T. K. L, T. K. Landauer, M. B. W. Wolfe and M. B. W. Wolfe., "Learning from text : Matching readers and texts by Latent Semantic Analysis," 1998.

[5] P. Wiemer-Hastings and al., "Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis," in *Dans les actes de Artificial Intelligence in Education, pp. IOS Press*, 1999.

[6] T. Landauer and S. Dumais, "A Solution to Plato's Problem : The Latent Semantic Analysis Theory of Acquisition Induction and Representation of Knowledge," *Psychological Review* , p. 211–240, 1997.

[7] V. Zampa and B. Lemaire, "Latent Semantic Analysis for User Modeling," in *J. Intell. Inf. Syst*, 2002.

[8] I. El Hassani and T. Masrour, "Arabic Semantic Text Classification Based on Wavelet Spectral Analysis," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 2, no. 6, pp. 2489-2496, 2013.

[9] I. El Hassani, A. Kriouile and Y. BenGhabrit, "Measure of fuzzy presence of descriptors on Arabic Text Mining," *IEEE In Information Science and Technology (CIST),* pp. 58-63, 2012, October.

[10] G. Salton and M. E. Lesk, "The SMART automatic document retrieval systems an illustration," in *Communications of the ACM*, 1965.

[11] G. Salton and C. Yang, "On the Specification of Term Values in Automatic Indexing," *Cornell University*, 1973.

[12] A. Boudlal, A. Lakhouaja, A. Mazroui, A. Meziane, M. O. a. o. bebah and a. M. Shoul., "Alkhalil morphosys: Morphosyntactic analysis system for non ocalized arabic," in *Seventh International Computing Conference in Arabic*, 2011.

[13] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* no. 77(2), pp. 257-286, 1989.