Partitioned-based Fuzzy Clustering to Learn Documents' Triadic Similarity

http://dx.doi.org/10.3991/ijes.v1i1.2932

S. Alouane Ksouri[1,2], M. Sassi Hidri[1] and K. Barkaoui[2]
[1] Université de Tunis El Manar, Tunis, Tunisia
[2] CEDRIC-CNAM, Paris, France

***Abstract*—With the development of the Web and the high availability of storage spaces, more and more documents become accessible. For that reason, similarity learning suffers from a scalability problem in both memory use and computational time when a data set is large. This paper provides a fuzzy triadic similarity measure to calculate memberships in a context of document co-clustering. It allows computing simultaneously fuzzy co-similarity matrices between documents/sentences and sentences/words. Each one is built on the basis of the others. The proposed model is extended to tackle the problem of large data sets by a splitting architecture which deals with a new fuzzy triadic similarity to parallelize both memory use and computation on distributed computers. This architecture is based on fuzzy clustering for partitioning data sets into similar groups (or clusters) in order to create more coherent sub-sets.**

Index Terms—Fuzzy clustering, Fuzzy sets, Fuzzy Triadic similarity, Parallel computing, Sentence matching,

I. INTRODUCTION

With the increasing number of available documents, the processing efficiency and scalability of the systems and their underlying computations become a major concern. For economic and operational reasons it is often preferable not to execute the computations on a single machine. One of the major problems is the similarity computing due to these huge data. Several methods dealing with this task are referred to as co-clustering approaches and have been extensively studied. In [1], a co-similarity measure has been proposed, called X-Sim [1][2] which builds on the idea of iteratively generating the similarity matrices between documents and words. This measure works well for unsupervised document clustering.

However, in recent research, the sentence has been considered as a more informative feature term to improve the effectiveness of document clustering [3]. While considering three levels Documents×Sentences×Words to represent the data set, we are able to deal with a dependency between them. It is done throught weights computing based on statistical models. But it has spawned the view that classical probability theory is unable to deal with uncertainties in natural language and machine learning.

We proceed to a fuzzification control process which converts crisp similarities to fuzzy ones. The conversion to fuzzy values is represented by the membership functions [4]. These fuzzy similarity matrices are used to calculate fuzzy similarity between documents, sentences and words in a triadic computing called FT-Sim (Fuzzy Triadic Similarity). Several extensions to the co-clustering methods have been proposed to deal with such multi-view data. Some works aims at combining multiple similarity matrices to perform a given learning task [5], [6], [7]. The idea being to build clusters from multiple similarity matrices computed along different views.

To combine multiple occurrences of FT-Sim, we can adopt sequential, merging or splitting-based parallel architectures. In this work, we are interested in the splitting-based mode. In this case, there are no methods to construct smaller data sets, and a random strategy is often used. It is for that reason that we propose a strategy to split a given data set which is considered as huge into smaller ones using the Fuzzy C-Means (FCM) clustering [8].

The rest of this paper is organized as follows: in section 2 we highlight backgrounds related to textual data co-clustering. In section 3 we discuss some previous work and present our motivations. In section 4 we provide a detailed description of our fuzzy triadic similarity. In section 5 we present our splitting-based model based on fuzzy clustering. Corresponding parallel architecture is described in section 6. In section 7 we conclude the paper and give some indications about further research.

II. BACKGROUNDS

The purpose of clustering is generally to organize a set of objects following criteria of similarity, to discover the structure according to which they are organized. The clustering has for objective to group these objects in homogeneous classes. So, the similarity of a couple of objects belonging to the same class must be maximized, whereas that of a pair of objects belonging to two different classes must be minimized.

Before the document clustering, a document *cleaning* procedure is executed for all documents. Several researchers consider that the main unit which characterizes a document is the word. The preprocessing step aims at the decrease of the noise, and the transformation of the data into an appropriate format, while extracting the most representative terms in the analyzed corpuses.

First, all non-word tokens are stripped off. Second, the text is parsed into words. Third, all stop words are identified and removed [9], [10].

The obtained data must be indexed. Several techniques have been proposed to index documents. The most commonly used one is the Vector Space Model (VSM) [11] and its graphical representation as a k-partite graph [12].

A second aspect that we must consider is weighting the terms (or words). Various techniques have been proposed to weight terms such as the binary valued vector weighting scheme indicating the presence or not of the word in the document; or real valued, indicating the importance of the word in this last. Several models have been proposed for computing real valued weights such as tf-idf [13], term distribution [14], or simply the number of occurrences of a word in a document, etc.

Another point of view is to represent a document as a collection of sentences. To achieve a more accurate document clustering, a more informative feature word-sentence has been considered in recent research work. A sentence of a document is an ordered sequence of one or more words [3].

Bigrams and trigrams [15] are commonly used methods to extract and identify meaningful sentences in statistical natural language processing. In [16], a method to compute all sub strings' (sentences) word and document frequencies in large document corpuses by using suffix array is presented. In [17], they propose a sentence-based document index model namely Document Index Graph (DIG) which allows an incremental construction of a sentence-based index for a set of documents. The quality of obtained Web documents clustering based on this model surpassed the traditional VSM-based approach.

Classically, data are described as a set of instances characterized by a set of features. For example, when using the VSM, text corpuses are represented by a matrix whose rows represent document vectors and whose columns represent the word vectors. The similarity between two documents obviously depends on the similarity between the words they contain and vice-versa.

The purpose of co-clustering is to take into account this duality to identify the relevant clusters [1].

Consequently, the concept of higher-order co-occurrences has been investigated and recently a new algorithm called X-Sim [18] was introduced. It exploits the duality between words and documents in a documents corpus as well as their respective higher order co-occurrences. While most researchers have focused to directly co-cluster the data, X-Sim [18] consists of building two similarity matrices, one for the rows and one for the columns, each being built iteratively on the basis of the other.

Moreover, with the development of the Web and the high availability of the storage spaces, more and more documents become accessible. Data can be provided from multiple sites and can be seen as a collection of matrices. By separately processing these matrices, we get a huge loss of information.

Several extensions to the co-clustering methods have been proposed to deal with such multi-view data. Some works aim at combining multiple similarity matrices to perform a given learning task [5], [6], the idea being to build clusters from multiple similarity matrices computed along different views.

Multi-view co-clustering such as MV-Sim [7] architecture, based on X-Sim measure [18] deals with the problem of learning co-similarities from a collection of matrices describing interrelated types of objects. It was proved that this architecture provides some interesting properties both in terms of convergence and scalability and it allows an efficient parallelization of the process.

III. DISCUSSION

In the traditional document models such as the VSM, words or characters are considered to be the basic terms in statistical feature analysis and extraction. The statistical features of all words are taken into account of the term weights (usually tf-idf) and similarity measures, whereas the sequence order of words is rarely considered in the clustering approaches based on the VSD model.

The motivation in this paper is that we believe that document clustering should be based not only on single word analysis, but on analysis of the sentence as well. Sentence-based analysis means that the similarity between documents should be based on matching sentences rather than single words only. Sentences contain more information than single words (information regarding proximity and order of words) and have a higher descriptive power. Thus a document must be broken down into a set of sentences, and a sentence is broken down into a set of words. We focus our work on how to combine the advantages of two representation models in document co-clustering. As a result, each document is represented as a vector of sentences, and each sentence is represented as a vector of words.

Most of the models use statistical approaches or probabilistic methods to model the membership of sentences (or words) in the documents. The question that arises is: are probabilistic methods and statistical techniques the best available tools for solving problems involving uncertainty?

This question is often answered negatively, especially by computer scientists and engineers. These respondents are motivated by the view that probability is inadequate for dealing with certain kinds of uncertainty.

In [19], it has been claimed that probability lacks sufficient expressiveness to deal with uncertainty in natural language. In contrast, fuzzy set theory prescribes a calculus for the treatment of uncertainty associated with classification. We purpose to apply this theory to the memberships computing. To determine the membership of a sentence (resp. word) in a document (resp. in a sentence), it is necessary to take into account the size of the document (number of sentences or words) and the frequency of appearance of the sentence (or of the word) in the document, to assure a coherence between the size and the number of occurrence.

However, with the development of the Web and the high availability of the storage spaces, more and more documents become accessible. Data can be provided from multiple sites and can be seen as a collection of matrices.

By separately processing these matrices, we get a huge loss of information.

We provide a splitting-based model for FT-Sim to tackle the problem of learning similarities from a collection of matrices. For multi-source or large matrices, we propose a parallel architecture in which each FT-Sim is the basic component or node we will use to deal with multiple matrices.

Thus, we consider a model in which data sets are distributed into N sites (or relation matrices). They describe the connections between documents for each local data set.

Our goal is then to compute a fuzzy Documents × Documents matrix $\widetilde{D}_2^{(i)}$ for each site i $(i = 1..N)$ trying to take into account all the representative information expressed in the relations.

IV. FT-SIM: FUZZY TRIADIC SIMILARITY

A. Assumptions and Notations

The following assumptions and notations are used in developing the proposed model:

- $D = \{D_1, D_2, \dots, D_i\}$ is a set of i documents $(i = 1..I)$
- $S = \{S_1, S_2, \dots, S_j\}$ is a set of j sentences $(j = 1..J)$
- $W = \{W_1, W_2, \dots, W_k\}$ is a set of k words $(k = 1..K)$
- $SD = [SD]_{ji}$ $(i = 1..I, j = 1..J)$: is a Sentence × Document similarity matrix of size J (sentences) by I (documents). It represents the number of occurrences of the j^{th} $(j = 1..J)$ sentence to the i^{th} $(i = 1..I)$ document.
- $WS = [WS]_{kj} (k = 1..K, j = 1..J)$: : is a Word × Sentence similarity matrix of size K (words or terms) by J (sentences). It represents the number of occurrences of the k^{th} $(k: 1..K)$: word to the j^{th} $(j = 1..J)$.
- $\widetilde{SD}_{ji} = [\mu]_{ji}$ $(i = 1..I, j = 1..J)$: is a fuzzy Sentence × Document matrix of size J (Sentences) by I (Documents). It represents the memberships degrees associated to the j^{th} sentence according i^{th} document.
- $\widetilde{WS}_{kj} = [\mu]_{kj}$ $(k = 1..K \text{ and } j = 1..J)$: a fuzzy Word × Sentence matrix of size K (Words) by J (Sentences). It represents the memberships degrees associated to the k^{th} word according j^{th} sentence.
- $\widetilde{D}_2^{(t)} = [\mu]_{lm}^{(t)}$ $(l: 1..I, m = 1..I)$: is a fuzzy Document × Document co-similarity matrix from the t^{th} iteration $(t = 0..it)$.
- $\tilde{S}_2^{(t)} = [\mu]_{lm}^{(t)}$ $(l: 1..J, m = 1..J)$: is a fuzzy Sentence × Sentence co-similarity matrix from the t^{th} iteration $(t = 0,1, \dots, it)$.
- $\widetilde{W}_2^{(t)} = [\mu]_{lm}^{(t)}$ $(l: 1..K, m = 1..K)$: is a fuzzy Word × Word co-similarity matrix from the t^{th} iteration $(t = 0,1, \dots, it)$.

To represent our textual data set, two representations have been proposed: the collection of matrices and the k-partite graph [12]. In the first, each matrix describes a view on the data. In the second, a graph is said to be k-partite when the nodes are partitioned into k subsets with the condition that no two nodes of the same subset are adjacent. Thus in the k-partite graph paradigm [12], a given subset of nodes contains the instances of one type of objects, and a link between two nodes of different subsets represents the relation between these two nodes.

From a functional point of view, the proposed FT-Sim model can be represented in the following way as shown in figure 1, where SD and WS are two data matrices representing a corpus and describing the connection between Documents/Sentences and Sentences/Words, brought by the three-partite graph [20].

$\widetilde{D}_2$ matrix provides a fuzzy similarity between the documents of the corpus, $\tilde{S}_2$ provides that between sentences of the corpus and $\widetilde{W}_2$ matrix provides that between words of the corpus. $\widetilde{D}_2$, $\tilde{S}_2$ and $\widetilde{W}_2$ are initialized with the identity matrix I at the first iteration. For each one, the matrix $\widetilde{D}_2$ is updated taking into account the similarity provided by $\tilde{S}_2$. $\tilde{S}_2$ is updated taking into account the similarity provided by $\widetilde{W}_2$ and $\widetilde{D}_2$. $\widetilde{W}_2$ is updated while taking into account the similarity provided by $\tilde{S}_2$.

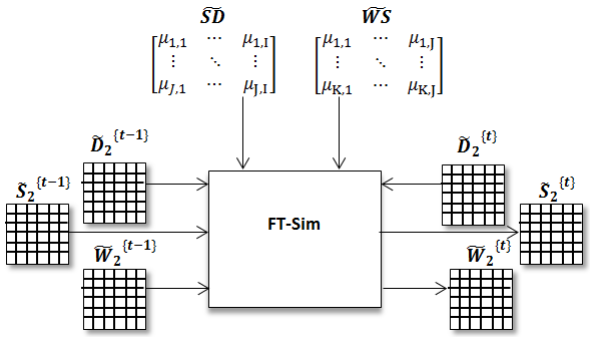

Figure 1. Functional diagram of FT-Sim

B. Fuzzification Controller Process

Let us consider the following similarity matrices between sentences and documents (resp. words and sentences):

$$SD = \begin{array}{c} \\ S_1 \\ S_2 \\ \vdots \\ SJ \end{array} \begin{array}{c} \begin{array}{cccc} D_1(S) & D_2(S) & \dots & D_I(S) \end{array} \\ \begin{bmatrix} SD_{11} & SD_{12} & \dots & SD_{1I} \\ SD_{21} & SD_{22} & \dots & SD_{2I} \\ \vdots & \vdots & \ddots & \vdots \\ SD_{J1} & SD_{J2} & \dots & SD_{JI} \end{bmatrix} \end{array} \quad (1)$$

and

$$WS = \begin{array}{c} \\ W_1 \\ W_2 \\ \vdots \\ W_k \end{array} \begin{array}{c} \begin{array}{cccc} S_1(W) & S_2(W) & \dots & S_J(W) \end{array} \\ \begin{bmatrix} WS_{11} & WS_{12} & \dots & WS_{1J} \\ WS_{21} & WS_{22} & \dots & WS_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ WS_{k1} & WS_{k2} & \dots & WS_{kJ} \end{bmatrix} \end{array} \quad (2)$$

Once these values are determined, we proceed to a fuzzification process. It converts crisp values to fuzzy ones. The conversion to fuzzy values is represented by the membership functions [4]. They allow a graphical representation of a fuzzy set. The x axis represents the universe of discourse (number of occurrences of sentences or words), whereas the y axis represents the membership degrees in the [0,1] interval.

For each document, we define a fuzzy membership function through a linear transformation between the lower bound value Li, a membership of 0, to the upper bound value Ui, which is assigned a membership of 1. This function is used because smaller values linearly increase in membership to the larger values for a positive slope and opposite for a negative slope.

The mathematical formulations of these functions are given in the following equations:

$$\widetilde{SD}_i = [\mu_{ji}] = \begin{cases} 1, & if\ SD_{ji} \geq L_i \\ \dfrac{SD_{ji} - U_i}{U_i - L_i}, & if\ L_i < SD_{ji} < U_i \\ 0 & if\ SD_{ji} \leq L_i \end{cases} \quad (3)$$

and

$$\widetilde{WS}_j = [\mu_{kj}] = \begin{cases} 1, & if\ WS_{kj} \geq L_j \\ \dfrac{WS_{kj} - U_j}{U_j - L_j}, & if\ L_j < WS_{kj} < U_j \\ 0 & if\ WS_{kj} \leq L_j \end{cases} \quad (4)$$

By applying these formulas, the fuzzy matrices are as follows:

$$\widetilde{SD} = [\mu]_{ji} = \begin{array}{c} \\ \tilde{S}_1(D) \\ \tilde{S}_2(D) \\ \vdots \\ \tilde{S}_J(D) \end{array} \begin{array}{c} \begin{array}{ccc} \widetilde{D}_1(S) & \dots & \widetilde{D}_I(S) \end{array} \\ \begin{bmatrix} \mu_{11} & \dots & \mu_{1I} \\ \mu_{21} & \dots & \mu_{2I} \\ \vdots & \ddots & \vdots \\ \mu_{J1} & \dots & \mu_{JI} \end{bmatrix} \end{array} \quad (5)$$

and

$$\widetilde{WS} = [\mu]_{kj} = \begin{array}{c} \\ \widetilde{W}_1(S) \\ \widetilde{W}_2(S) \\ \vdots \\ \widetilde{W}_k(S) \end{array} \begin{array}{c} \begin{array}{ccc} \tilde{S}_1(W) & \dots & \tilde{S}_J(W) \end{array} \\ \begin{bmatrix} \mu_{11} & \dots & \mu_{1J} \\ \mu_{21} & \dots & \mu_{2J} \\ \vdots & \ddots & \vdots \\ \mu_{K1} & \dots & \mu_{KJ} \end{bmatrix} \end{array} \quad (6)$$

Before proceeding to fuzzy triadic computing, we must initialize Documents×Documents, Sentences×Sentences and Words×Words matrices with the identity ones denoted as $\widetilde{D}_2^{(0)}$, $\tilde{S}_2^{(0)}$ and $\widetilde{W}_2^{(0)}$. The similarity between the same documents (resp. sentences and words) has the value equal to 1. All others values are initialized with zero. $\widetilde{D}_2^{(t)}$, $\tilde{S}_2^{(t)}$ and $\widetilde{W}_2^{(t)}$ are as follows:

$$\widetilde{D}_2 = \begin{array}{c} \\ D_1 \\ D_2 \\ \vdots \\ D_l \end{array} \begin{array}{c} \begin{array}{ccc} D_1 & \dots & D_m \end{array} \\ \begin{bmatrix} 1 & \dots & \mu_{1m}^{(t)} \\ \mu_{21}^{(t)} & \dots & \mu_{2m}^{(t)} \\ \vdots & \ddots & \vdots \\ \mu_{l1}^{(t)} & \dots & 1 \end{bmatrix} \end{array} \quad (7)$$

where $\mu_{lm}^{(t)}$ ($l = 1..I\ (resp.\ l = 1..J\ and\ l = 1..K)$, $m = 1..I\ (resp.\ m = 1..J$ and $l = 1..K\ )$ is the membership degree of the l^{th} document according the m^{th} one. Similarly, we represent $\tilde{S}_2^{(t)}$ and $\widetilde{W}_2^{(t)}$.

After initializing $\widetilde{D}_2^{(t)}$ with $\widetilde{D}_2^{(0)}$, we calculate the new matrix $\widetilde{D}_2^{(t)}$ representing fuzzy similarities between documents while using $\tilde{S}_2^{(t-1)}$ and $\widetilde{SD}$.

Usually, the similarity measure between two documents D_l and D_m is defined as a function that is the sum of the similarities between shared sentences.

Our idea is to generalize this function in order to take into account the intersection between all the possible pairs of sentences occurring in documents D_l and D_m. In this way, not only can we capture the fuzzy similarity of their common sentences but also the fuzzy ones coming from sentences that are not directly common in the documents but are shared with some other documents. For each pair of sentences not directly shared by the documents, we need to take into account the fuzzy similarity between them as provided by $\tilde{S}_2^{(t-1)}$.

Since we work with fuzzy matrices formed by membership degrees, we should certainly be applied in accordance with the operators for fuzzy sets, especially the intersection and union. Thus, $\mu_{lm}^{(t)}$, except the case $l = m$, can be formulated as follows:

$$\mu_{lm}^{(t)} = \sum_{i=1}^{J} \sum_{j=1}^{J} Min(\mu_{il}, \mu_{jm}) * \mu_{ij}^{\tilde{S}_2^{(t-1)}} \quad (8)$$

As we have shown for $\widetilde{D}_2^{(t)}$ computing, we generalize fuzzy similarities in order to take into account the intersection between all the possible pairs of words occurring in sentences Sl and Sm. In this way, not only can we capture the fuzzy similarity of their common words but also the fuzzy ones coming from words that are not directly common in the sentences but are shared with some other sentences.

For each pair of words not directly shared by the sentences, we need to take into account the fuzzy similarity between them as provided by $\widetilde{W}_2^{(t-1)}$.

The overall fuzzy similarity between documents S_l and S_m is defined in the following equation:

$$\mu_{lm}^{(t)} = Min[\sum_{i=1}^{I} \sum_{j=1}^{I} Min(\mu_{il}, \mu_{jm}) * \mu_{ij}^{\widetilde{D}_2^{(t-1)}}, \sum_{i=1}^{K} \sum_{j=1}^{K} Min(\mu_{il}, \mu_{jm}) * \mu_{ij}^{\widetilde{W}_2^{(t-1)}}] \quad (9)$$

For each pair of words not directly shared by the sentences, we need to take into account the fuzzy similarity between them as provided by $W_2^{(t-1)}$.

The overall fuzzy similarity between documents W_l and W_m is defined in the following equation:

$$\mu_{lm}^{(t)} = \sum_{i=1}^{K}\sum_{j=1}^{K} Min(\mu_{il}, \mu_{jm}) * \mu_{ij}^{\widetilde{W}_2^{(t-1)}} \quad (10)$$

As shown by algorithm 1, the Fuzzy triadic algorithm proposed is based on an iterative approach, in which each iteration t consists in evaluating the similarities according to the documents/sentences/words three-partite graph.

Algorithm 1 Fuzzy Triadic Algorithm

Inputs: DS, SW, It

Outputs: $\widetilde{D}_2^{(t)}, \tilde{S}_2^{(t)}, \widetilde{W}_2^{(t)}$,

1: $\widetilde{D}_2^{(0)} \leftarrow$ Identity

2: $\tilde{S}_2^{(0)} \leftarrow$ Identity

3: $\widetilde{W}_2^{(0)} \leftarrow$ Identity

4: **For** t = 1..It **do**

5: Computing $\widetilde{D}_2^{(t)}$ with $\tilde{S}_2^{(t-1)}$ eq.(8)

6: Computing $\tilde{S}_2^{(t)}$ with $\widetilde{D}_2^{(t-1)}$ and $\widetilde{W}_2^{(t-1)}$ eq.(9)

7: Computing $\widetilde{W}_2^{(t)}$ with $\tilde{S}_2^{(t-1)}$ eq.(10)

8: **End For**

This model presents a solution which is not based on any strategy. There is no guarantee to obtain an interesting matrix for an optimal processing by varying the number of splits. Thus, we cannot deduct rules to be applied to this kind of problem.

V. SPLITTING-BASED MODEL

In order to reduce the complexity of the problem of treating huge databases, it is possible to split a given data matrix into a collection of smaller ones, each sub-matrix becoming a component of our network and processed as a separate view. The splitting strategy can be random or use the FCM algorithm [8].

A. Random Split

Let us suppose H machines (or nodes) are allocated in a distributed environment for our target similarity learning task. In the random split strategy, we can choose the number of splits H, corresponding with the number of cores we have.

Then we explore the behavior of the proposed architecture while varying the number of H splits, obtaining H sub-matrices with the aim of finding the one most suitable with our solution. The split is performed on sentences, (random split sentence method); for $\widetilde{SD} = [\mu_{ji}]\ (i = 1..I, j = 1..J)$ and $\widetilde{WS} = [\mu_{kj}]\ (k = 1..K, j = 1..J)$ matrices, the sentences are divided into H sub-sets there by forming H sub-matrices $\widetilde{SD}^{(i)}$and $\widetilde{WS}^{(i)}(i = 1..H)$ of size respectively $\frac{J}{H} by\ I$ and k by $\frac{J}{H}$. So, The number of *FT-Sim(i)* instances in the proposed network is equal to the number of splits H. Figure 2 shows the overview of the random splitting process.

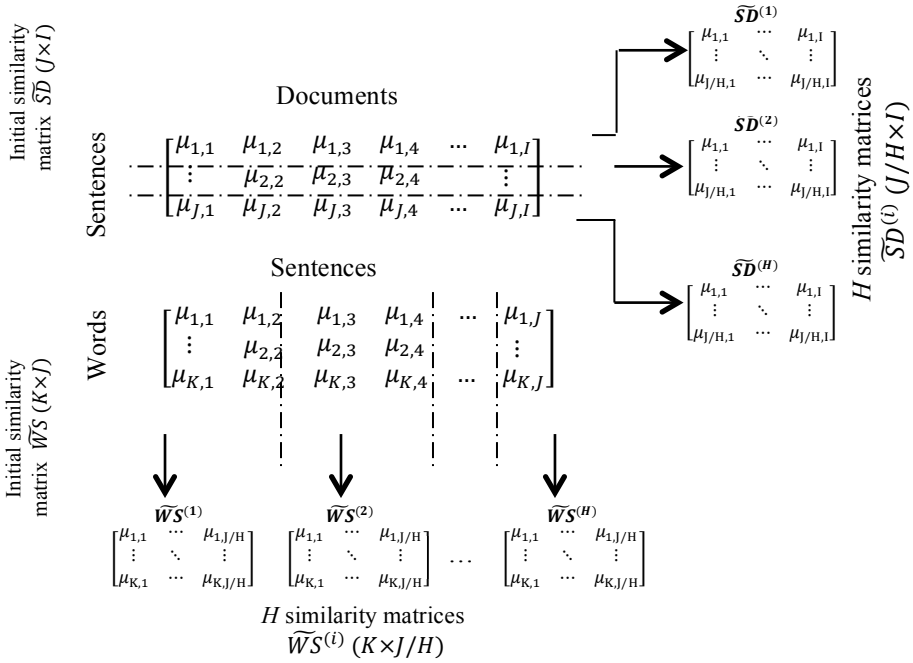

Figure 2. Random splitting

B. FCM-based Split

The second alternative to split a given data set, is to adopt the FCM algorithm [8]. As shown in section 2, the *FT-Sim* is the exploitation of the trial nature of the problem of similarity.

That means the relationship between groups of sentences that occur in a group of documents and the relationship between groups of words that occur in a group of sentences. Thus, documents are considered similar and hence grouped together, if they contain similar sentences, and sentences in turn are considered similar and therefore grouped together, if they occur in similar documents, etc.

The idea behind this method is that by preceding a rapid clustering of sentences before construction of the sub-matrices for each core as it is shown in figure 3, we can already obtain groups of similar sentences. This will facilitate the following task which is the parallel co-similarity learning.

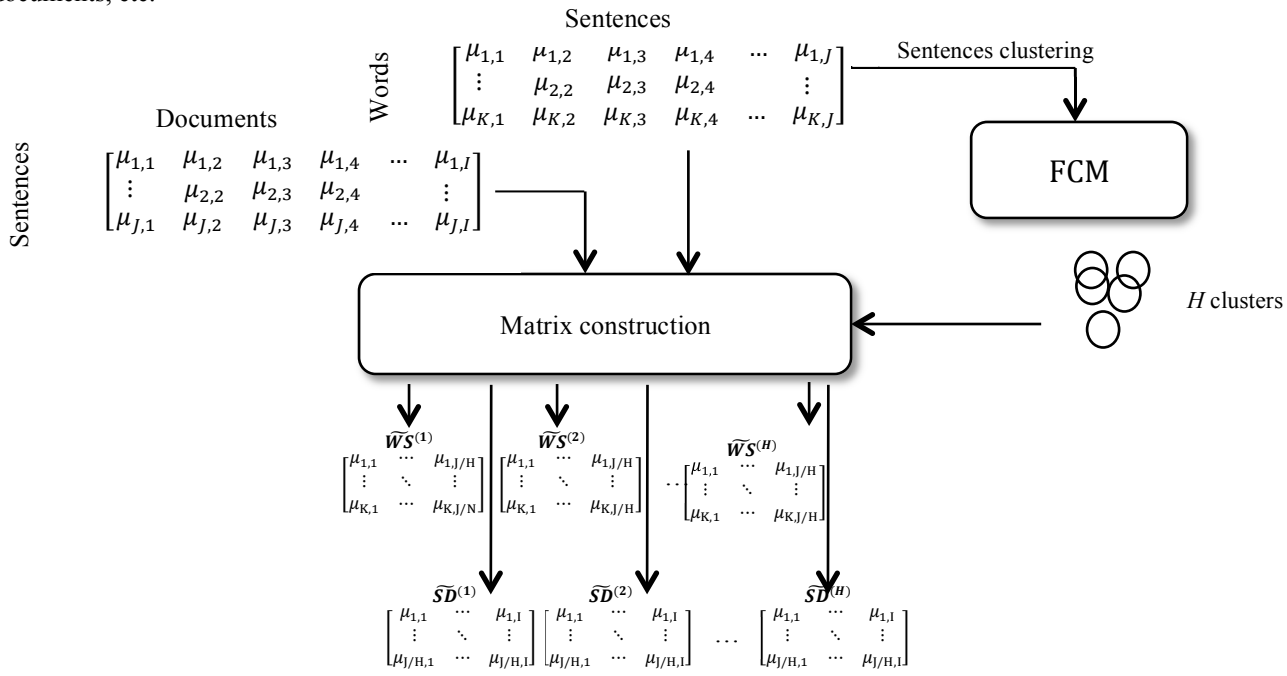

Figure 3. FCM-based splitting

The main idea behind this is the unsupervised data clustering while adopting a fuzzy partitioned-based strategy. Fuzzy clustering methods allow objects to belong to several clusters simultaneously, with different degrees of membership. The data set $X = \{X_1, .., X_N\} \subset R_M$ is thus partitioned into c fuzzy subsets. The result is the partition matrix $U = [\mu_{ji}]$ for $i = 1..N$ and $j = 1..c$. The aim of this algorithm is to minimize an objective function, denoted as j_m, in the following form:

$$Minimise\, J_m(U,V) = \sum_{i=1}^{N}\sum_{j=1}^{c} \mu_{ij}^{m} \left\| (x_i - v_j) \right\|^2 \qquad (11)$$

μ_{ij} must satisfy $\sum_{i=1}^{N} \mu_{ij} = 1, \mu_{ij} \geq 0 \; \forall \; i = 1,2,..,N$ and $j = 1,2,..,c$. The parameter $m > 1 \; in \; J_m(U,V)$) is defined and well-studied in [8].

Thus, the FCM can be used to obtain a set of clusters with similar sentences.

In this way we can exploit these results to construct sub-matrices Documents × Sentences and Sentences × Words with similar sentences, with the aim to have more coherent matrices.

Dividing a huge database into smaller ones can considerably reduce the time and the complexity of the computing, but we must add the complexity of the FCM run; which we could ignore because FCM is considered as rapid compared with the other clustering method.

On the other hand this solution permitted us to gain in time for the following task which is the co-similarity learning.

VI. SPLITTING-BASED PARALLEL ARCHITECTURE

After the splitting step, we compute the similarity matrices from several local sub-matrices and aggregate them before performing the co-clustering algorithm on it.

Figure 4 shows the splitting-based parallel architecture.

In this topology, all local $FT - SIM^{(i)}$ instances $(i = 1..H)$ are run in parallel, then the similarity matrices $\widetilde{D}_2^{(i)}$ are simultaneously updated with an aggregation function. This method offers the benefit that all the instances of $FT - SIM^{(i)}$ have the same influence.

The aggregation function takes H matrices $(\widetilde{D}_2^{(1)})^{\{t\}}$, $\widetilde{D}_2^{(2)})^{\{t\}}, .., \widetilde{D}_2^{(H)})^{\{t\}}$ issue from each data source i for a given iteration t. If a given document does not appear in a single local data source, then we assign its corresponding similarity measures directly in $\widetilde{D}_2$. If a particular document appears in several different local data sources, we assign the minimum of all similarity measures relevant to this document to $\widetilde{D}_2$ without taking into account the value of 0. The different steps of aggregation computing are presented in algorithm 2.

So, for a given iteration t, each instance $FT - SIM^{(i)}$ produces its own similarity matrix $(\widetilde{D}_2^{(i)})^{\{t\}}$. We thus get a set of output similarity matrices

$\left\{(\widetilde{D}_2^{(1)})^{\{t\}}, \widetilde{D}_2^{(2)})^{\{t\}}, .., \widetilde{D}_2^{(H)})^{\{t\}}\right\}$ the cardinal of which being equal to the number of local data sets related to H.

Therefore, we use the aggregation function denoted by $\otimes$ and developed in the aggregation function to compute a consensus similarity matrix merging all of the $(\widetilde{D}_2^{(1)})^{\{t\}}, \widetilde{D}_2^{(2)})^{\{t\}}, .., \widetilde{D}_2^{(H)})^{\{t\}}$ with the current matrix $\widetilde{D}_2^{\{t\}}$.

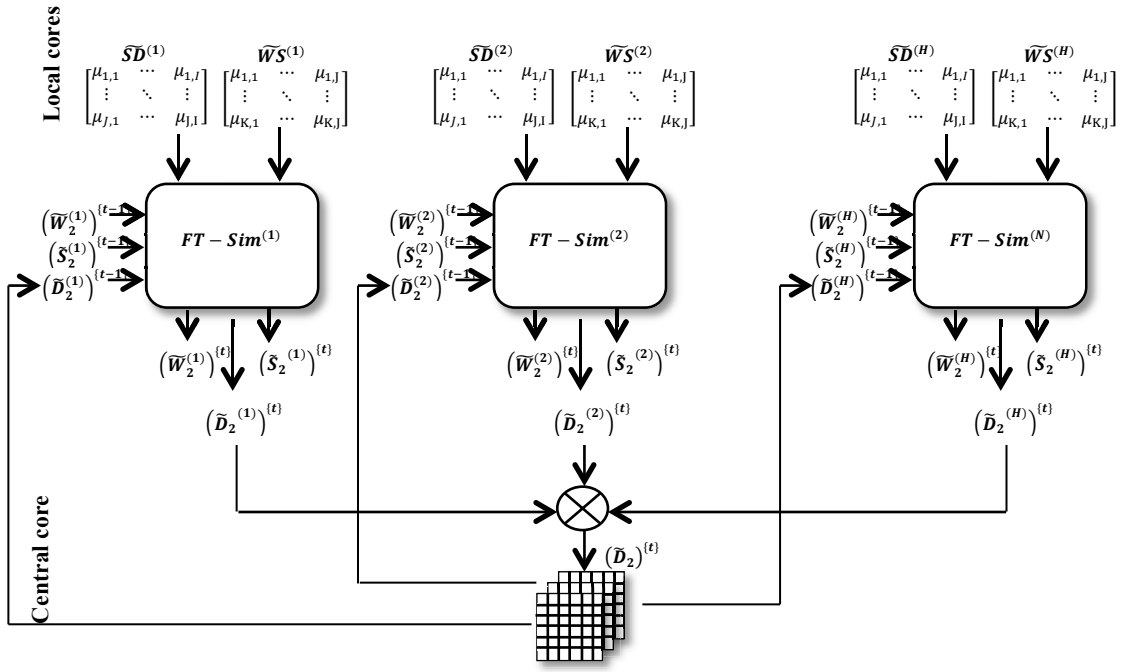

Figure 1. Splitting-based parallel architecture

In turn, this resulting consensus matrix is connected to the inputs of all the $FT - SIM^{(i)}$ instances, to be taken into account in the $t + 1^{th}$ iteration, thus creating feedback loops allowing the system to spread the knowledge provided by each $(\widetilde{D}_2^{(i)})^{\{t\}}$within the network.

Algorithm 2 Aggregation Function
Inputs: Collection of H matrices $\left\{(\widetilde{D}_2^{(1)})^{\{t\}}, \widetilde{D}_2^{(2)})^{\{t\}}, .., \widetilde{D}_2^{(H)})^{\{t\}}\right\}$
Output: $\widetilde{D}_2$
1: I ← Compute the number of documents in $\left\{(\widetilde{D}_2^{(1)})^{\{t\}}, \widetilde{D}_2^{(2)})^{\{t\}}, .., \widetilde{D}_2^{(H)})^{\{t\}}\right\}$
2: Let $\widetilde{D}_2 = [\mu_{lm}]$ (l=1..I and m = 1..I)
3: $\widetilde{D}_2$ ← Identity
4: **For** Each document $D_l of\ \widetilde{D}_2$ **do**
5: **If** D_l Appear in only one data set s **Then**
6: $\mu_{l,*} \leftarrow \mu_{l,*}{}^{(s)}$
7: **Else**
8: $\mu_{l,*}$ ← min(All $\mu_{l,*}{}^{(i)}$) i ∈ { sites where D_l appear} with $\mu_{l,*}{}^{(i)} \neq 0$
9: **End If**
10: **End For**

The complexity of this architecture is obviously related to that of the $FT - SIM^{(i)}$ algorithm. In the parallel splitting-based architecture, as each instance of $FT - SIM^{(i)}$can run on an independent core, the method can easily be parallelized, thus keeping the global complexity unchanged (considering the number of iterations as a constant factor). So, the complexity of the aggregation function can be ignored.

By splitting a matrix, we lost some information. The solution does not compute the co-similarities between all pairs of sentences but only between the words occurring in each $\widetilde{SD}^{(i)}$. Thanks to the feedback loops of this architecture and to the presence of the common similarity matrix $\widetilde{D}_2$, we will be able to spread the information through the network and alleviate the problem of inter-matrix comparisons.

The algorithm 3 presents the different steps of the parallel splitting-based process.

Algorithm 3 Parallel splitting-based algorithm
Inputs: Collection of matrices $\widetilde{SD}^{(i)}$, $\widetilde{WS}^{(i)}$, $(i = 1..H)$, T
Output: $\widetilde{D}_2$
1: **For all** i **do**
2: $(\widetilde{D}_2^{(i)})^{\{0\}}$← Identity
3: $(\tilde{S}_2^{(i)})^{\{t\}}$← Identity
4: $(\widetilde{W}_2^{(i)})^{\{t\}}$← Identity
5: **For** i = 1..T **do**
6: Execute every $FT - SIM^{(i)}$with $\widetilde{SD}^{(i)}$,$\widetilde{WS}^{(i)}$ and t = *1*
7: $(\widetilde{D}_2)^{\{t\}}$ ← Aggregation of all $(\widetilde{D}_2^{(i)})^{\{t\}}$ Alg.(2)
8: Update each $(\widetilde{D}_2^{(i)})^{\{t\}}$
9: **End For**
10: **End For**

Thus, by using a parallel version of $FT - SIM^{(i)}$on H cores, we will gain both in time and space complexity. Indeed, the time complexity decreases, leading to an overall gain of $1/H^2$. In the same way, the memory

needed to store the similarity matrices between words will decrease by a $1/H$ factor.

VII. CONCLUSION

In this paper, a fuzzy triadic similarity model for the co-clustering task has been proposed. It takes, iteratively, into account three abstraction computing levels (documents/sentences/words). The sentences consisting of one or more words are used to designate the fuzzy co-similarity of two documents. We are able to cluster together documents that have similar concepts based on their shared (or similar) sentences and in the same way to cluster together sentences based on words. This also allows us to use any classical clustering algorithm such as FCM or other fuzzy partitioned-based clustering approaches.

We propose that our fuzzy Triadic similarity-based model gives an ownership of words-sentences memberships in accordance with the size of a document. This can deal with uncertainty associated with co-clustering. This ensures a good interpretation of the result of the co-clustering method which does not need to cluster the words-sentences for clustering the documents.

To tackle the problem of big dimensions of matrices, we have proposed an extension of our fuzzy triadic co-similarity model to the multi-view one. There is no single right way to approach analysis of a large data set, and it is often an assortment of complementary approaches, building one upon another. Thus, we have proposed to use our fuzzy triadic similarity method associated with the FCM clustering. Before proceeding to the similarity learning, we have proposed to split the data set, considered as huge, into a set of smaller ones, according to the sentences. There are several possible alternatives of splitting. We opted for the split, based on a preliminary rapid clustering using the FCM method, and that to obtain coherent sub-matrices. Then, a parallel architecture is presented, which combines FT-Sim instances to compute similarities on different cores. The combination of these techniques may be particularly useful to help simplify computing on large amounts of textual data and to focus on the rich, descriptive, and expressive details of qualitative data.

For future work, many directions seem compelling to explore in the splitting approach. More sophisticated models will be investigated such as a two dimensional splitting in order to divide the data set according to sentences and words.

REFERENCES

[1] F. Hussain, X-Sim: *A New Cosimilarity Measure: Application to Text Mining and Bioinformatics*. Phd Thesis, 2010.

[2] G. Bisson and F. Hussain, "Chi-Sim: A new similarity measure for the co-clustering task." *International Conference on Machine Learning and Applications*, vol. 7, pp. 211–217, 2008.

[3] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering." *Knowledge and Data Engineering, IEEE Transactions*, vol. 20, pp. 1217–1229, 2008. http://dx.doi.org/10.1109/TKDE.2008.50

[4] S. Kundu, "Min-transitivity of fuzzy leftness relationship and its application to decision making," *Fuzzy Sets and Systems,* vol. 93, pp. 357–367, 1997. http://dx.doi.org/10.1016/S0165-0114(96)00122-4

[5] Z. W. Tang and I.S. Dhillon, "Clustering with multiple graphs," *IEEE International Conference on Data Mining*, vol. 9, pp. 1016–1021, 2009.

[6] Y. L. F. de Carvalho and F. Melo, "Partitioning hard clustering algorithms based on multiple dissimilarity matrices," *in Pattern Recognition 45*.

[7] G. Bisson and C. Grimal, "Co-clustering of multi-view datasets: a parallelizable approach," *IEEE International Conference on Data Mining*, vol. 12, pp. 828–833, 2012.

[8] J.C.Bezdek, "FCM: The fuzzy c-means clustering algorithm." *Computers et Geosciences*, vol. 10(2-3), pp. 191–203, 1984. http://dx.doi.org/10.1016/0098-3004(84)90020-7

[9] J.M. Torres-Moreno, P. Velzquez-Morales and J. Meunier, "Un algorithme pour la condensation automatique de textes," *in Colloque Interdisciplinaire en Sciences Cognitives*, 2001.

[10] M. Roche, T. Heitz, O. Matte-Tailliez and Y. Kodratoff, "Exit: Un système itératif pour l'extraction de la terminologie du domaine à partir de corpus spécialisés." *in Journée internationales d'Analyse statistique des Données Textuelles JADT*, 2004.

[11] G. Salton, A. Wong and C.S. Yang, "A vector space model for automatic indexing." *Communications of the ACM,* vol. 18, pp. 613–620, 1975. http://dx.doi.org/10.1145/361219.361220

[12] Z.B. Long, X. Wu and Z.Y. Philip, "Unsupervised learning on k-partite graphs," *12th ACM SIGKDD international conference on Knowledge discovery and data mining*, vol. (12), pp. 317–326, 2006.

[13] G. Salton and C. Buckley, "Term-weighing approaches in automatic text retrieval." *Communications of the ACM Information Processing and Management*, vol. 34, pp. 513–523, 1988. http://dx.doi.org/10.1016/0306-4573(88)90021-0

[14] V. Lertnattee and T. Theeramunkong, "Effect of term distributions on centroid-based text categorization." *Information Sciences*, vol. 158, pp. 89–115, 2004. http://dx.doi.org/10.1016/j.ins.2003.07.007

[15] M. Sven, L. Jorg and N. Hermann, "Algorithms for bigram and trigram word clustering." *Speech Communication*, vol. 24, p. 1937, 1998.

[16] M. Yamamoto and W. Kenneth, "Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus." *Computational Linguistics*, vol. 27, pp. 1–30, 2001. http://dx.doi.org/10.1162/089120101300346787

[17] F. Bashirahamad, "Web document clustering using document index graph." *in International Conference on Advanced Computing and Communications*, 2006, pp. 32–37.

[18] G. Bisson, F. Hussain and S. Chi, "A new similarity measure for the coclustering task." *in 7th International Conference on Machine Learning and Applications (ICMLA)*, 2008, pp. 211–217.

[19] L.A. Zadeh, "A simple view of the dempster-shafer theory of evidence and its implications for the rule of combination of evidence." *AI Magazine*, vol. 7, pp. 85–90, 1986.

[20] S. Alouane, M. Hidri and K.Barkaoui, "Fuzzy triadic similarity for text categorization: towards parallel computing." *in International Conference on Web and Information Technologies*, 2013, pp. 265–274.

AUTHORS

S. Alouane Ksouri is a PhD student with the collaboration of LR-SITI from the National Engineering School of Tunis (ENIT), Tunisia, and the CEDRIC-CNAM Paris. (e-mail: alouane.sonia@yahoo.fr).

M. Sassi Hidri is an Assistant Professor of Computer Science at the Information and Communication Technologies (TIC) department from ENIT. She is with LR-SITI laboratory, BP. 37, le Belvédère 1002 Tunis, Tunisia. (e-mail: minyar.sassi@enit.rnu.tn).

K. Barkaoui is a Professor of Computer Science at the Conservatoire National des Arts et Métiers (CNAM - Paris). He is with CEDRIC-CNAM laboratory, Rue Saint-Martin Paris 75003, France. (e-mail: kamel.barkaoui@cnam.fr).

Submitteed 20 June 2013. Published as re-submitted by the authors 23 July 2013.