AN EFFICIENT APPROACH TO IMPROVE ARABIC DOCUMENTS CLUSTERING BASED ON A NEW KEYPHRASES EXTRACTION ALGORITHM

Hanane FROUD , Issam SAHMOUDI and Abdelmonaime LACHKAR

L.S.I.S, E.N.S.A Sidi Mohamed Ben Abdellah University (USMBA) Fez, Morocco
hanane_froud@yahoo.fr
Issam.sah@gmail.com
abdelmonaime_lachkar@yahoo.fr

ABSTRACT

Document Clustering algorithms goal is to create clusters that are coherent internally, but clearly different from each other. The useful expressions in the documents is often accompanied by a large amount of noise that is caused by the use of unnecessary words, so it is indispensable to eliminate it and keeping just the useful information.

Keyphrases extraction systems in Arabic are new phenomena. A number of Text Mining applications can use it to improve her results. The Keyphrases are defined as phrases that capture the main topics discussed in document; they offer a brief and precise summary of document content. Therefore, it can be a good solution to get rid of the existent noise from documents.

In this paper, we propose a new method to solve the problem cited above especially for Arabic language documents, which is one of the most complex languages, by using a new Keyphrases extraction algorithm based on the Suffix Tree data structure (KpST). To evaluate our approach, we conduct an experimental study on Arabic Documents Clustering using the most popular approach of Hierarchical algorithms: Agglomerative Hierarchical algorithm with seven linkage techniques and a variety of distance functions and similarity measures to perform Arabic Document Clustering task. The obtained results show that our approach for extracting Keyphrases improves the clustering results.

KEYWORDS

Arabic Language, Arabic Text Clustering, Hierarchical Clustering, Suffix Tree Algorithm, Keyphrases Extraction, Similarity Measures.

1. INTRODUCTION

As we know text clustering is one of the important text mining tasks and now it becomes a natural activity in every organization. Documents clustering refer to the process of grouping documents with similar contents or topics into clusters using different clustering methods and algorithms.

Sundarapandian et al. (Eds) : ICAITA, SAI, SEAS, CDKP, CMCA-2013
pp. 243–256, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3820

Traditional documents clustering algorithms use the full-text in the documents to generate feature vectors. Such methods often produce unsatisfactory results because there is much noisy information in documents. There is always a need to summarize information into compact form that could be easily absorbed. The challenge is to extract the essence of text documents collections and present it in a compact form that identifies their topic(s).

In this paper, we propose to use Keyphrases extraction to tackle these issues when clustering documents. The Keyphrases extraction from free text documents is becoming increasingly important as the uses for such technology expands. Keyphrases extraction plays a vital role in the task of indexing, summarization, clustering [1], categorization and more recently in improving search results and in ontology learning. Keyphrases extraction is a process by which the set of words or phrases that best describe a document is specified. In our work, we presented our novel Keyphrases extraction algorithm based on the Suffix Tree data structure (KpST) to extract the important Keyphrases from Arabic documents.

The outputs of the proposed approach will be the inputs of the most popular approach of Hierarchical algorithms used in our experimental study on Arabic documents clustering: Agglomerative Hierarchical algorithms with seven linkage techniques for hierarchical clustering algorithms using a wide variety of distance functions and similarity measures, such as the Euclidean Distance, Cosine Similarity, Jaccard Coefficient, and the Pearson Correlation Coefficient [2][3], in order to test their effectiveness on Arabic documents clustering.

The remainder of this paper is organized as follows. The next section discusses our novel approach to extract the Keyphrases using suffix tree algorithm for Arabic documents. Section 3 presents the similarity measures and their semantics. Section 4 explains experiment settings, dataset, evaluation approach, results and analysis. Section 5 concludes and discusses our future works.

2. ARABIC KEYPHRASE EXTRACTION USING SUFFIX TREE ALGORITHM

2.1. Keyphrase Extraction

Keyphrases are widely used in large document collections. They describe the content of single documents and provide a kind of semantic metadata that is useful for a variety of purposes. Many Keyphrases extractors view the problem as a classification problem and therefore they need training documents (i.e. documents which their Keyphrases are known in advance). Other systems view Keyphrases extraction as a ranking problem. In the latter approach, the words or phrases of a document are ranked based on their importance and phrases with high importance (usually located at the beginning of the list) are recommended as possible Keyphrases for a document.

From the observation of human-assigned Keyphrases, we conclude that good Keyphrases of a document should satisfy the following properties:

- **Understandable**. The Keyphrases are understandable to people. This indicates the extracted Keyphrases should be grammatical.
- **Relevant**. The Keyphrases are semantically relevant with the document theme.
- **Good coverage**. The Keyphrases should cover the whole document well.

Keyphrases extraction algorithms fall into two categories: Keyphrases extraction from single documents, which is often posed as a supervised learning task [19], and Keyphrases extraction from a set of documents, which is an unsupervised learning task that tries to discover the topics rather than learn from examples. As an example of an unsupervised Keyphrases extraction approach, the graph-based ranking [20] regards Keyphrases extraction as a ranking task, where a document is represented by a term graph based on term relatedness, and then a graph-based ranking algorithm is used to assign importance scores to each term. Existing methods usually use term co occurrences within a specified window size in the given document as an approximation of term relatedness [20].

In this paper, a novel unsupervised Keyphrases extraction approach based on generalized Suffix Tree construction for Arabic documents is presented and described.

2.2. Suffix Tree

A **Suffix Tree** is a data structure that allows efficient string matching and querying. It have been studied and used extensively, and have been applied to fundamental string problems such as finding the longest repeated substring, strings comparisons , and text compression[24]. The Suffix Tree commonly deals with strings as sequences of characters, or with documents as sequences of words. A **Suffix Tree** of a string is simply a compact tree of all the suffixes of that string. The **Suffix Tree** has been used firstly by Zamir et al. [25] as clustering algorithm named **Suffix Tree Clustering** (STC). It's linear time clustering algorithm that is based on identifying the shared phrases that are common to some **Document's Snippets** in order to group them in one cluster. A **phrase** in our context is an ordered sequence of one or more words. ST has two logical steps: Document's "Cleaning", and Identifying **Suffix Tree Document Model.**

2.2.1. Document's "Cleaning"

In this step, each snippet is processed for Arabic stop-words removal such as (e.g., فانه وان والذي وهذا فكان ستكون...): Stop-word means high frequency and low discrimination and should be filtered out in the IR system. They are functional, general, and common words of the language that usually do not contribute to the semantics of the documents and have no read added value. Many Information retrieval systems (IRS) attempt to exclude stop-words from the list of features to reduce feature space, and to increase their performance. In addition, in our case, to deal especially with Arabic snippets, we propose also in this step to remove Latin words and specials characters such as (e.g. $, #,...).

2.2.2. Suffix Tree Document Model

The Suffix Tree treats documents as a set of phrases (sentences) not just as a set of words. The sentence has a specific, semantic meaning (words in the sentence are ordered). Suffix Tree Document Model (STDM) considers a document d = $w_1w_2…w_m$ as a string consisting of words w_i, not characters (i = 1; 2;…; m). A revised definition of suffix tree can be presented as follow: A **Generalized Suffix Tree** for a set S of n strings, each of length m_n, is a rooted directed tree with exactly $\sum m_n$ leaves marked by a two number index (k,*l*) where k ranges from 1 to n and *l* ranges from 1 to m_k. Each internal node, other than the root, has at least two children and each edge is labeled with a nonempty substring of words of a string in S. No two edges out of a node can have edge labels beginning with the same word. For any leaf (i,j), the concatenation of the edge labels on the path from the root to leaf(i, j) exactly spells out the suffix of S_i that starts at position j , that's it spells out S_i[j…m_i][12].The Figure.1 shows an example of the generated Suffix Tree of a set of three Arabic strings or three Documents–Document1: "**القط يأكل الجبن**" , Document2: "**الفارياكل الجبن ايضا**", Document3: "**القط ياكل الفارايضا** ", respectively the Figure.2 shows the same example in English Language (Document1: "**cat ate cheese**", Document2: "**mouse ate cheese too** “ and Document3: "**cat ate mouse too**") .

Arabic Documents: A

Document (1,"القط يأكل الجبن ")

Document (2, "الفار ياكل الجبن ايضا")

Document (3,"القط ياكل الفار ايضا ")

B

```
STDM {
        1 -> [label="القط ياكل" doc:(1,3))];
                2 -> [label="الجبن $" doc:(1))];
                2 -> [label="الفار ايضا $" doc:(3))];
        1 -> [label="ياكل" doc:(1,2,3))];
                2 -> [label="الجبن" doc:(1,2))];
                        3 -> [label="$" doc:(1))];
                        3 -> [label="ايضا $" doc:(2))];
                2 -> [label="الفار ايضا $" doc:(3))];
        1 -> [label="الجبن" doc:(1,2))];
                2 -> [label="$" doc:(1))];
                2 -> [label="ايضا $" doc:(2))];
        1 -> [label="$" doc:(1,2,3))];
        1 -> [label="الفار" doc:(2,3))];
                2 -> [label="ياكل الجبن ايضا $" doc:(2))];
                2 -> [label="ايضا $" doc:(3))];
        1 -> [label="ايضا $" doc:(2,3))];
}
```

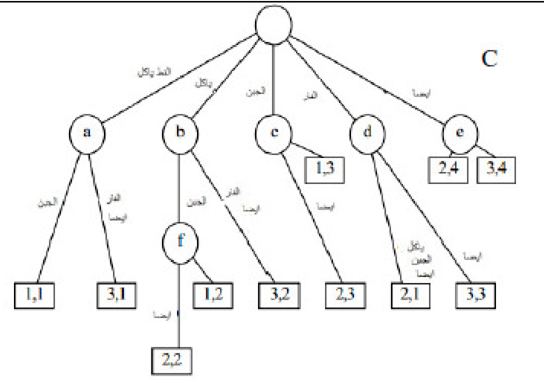


Figure.1 Suffix Tree for Arabic example (A,B,C)

English Documents: A

Document 1 : "cat ate cheese",

Document 2: "mouse ate cheese too",

Document 3: "cat ate mouse too",

B

```
STDM {
        1 -> [label="cat ate" doc:(1,3))];
                2 -> [label="cheese $" doc:(1))];
                2 -> [label="mouse too $" doc:(3))];
        1 -> [label="ate" doc:(1,2,3))];
                2 -> [label="cheese" doc:(1,2))];
                        3 -> [label="$" doc:(1))];
                        3 -> [label="too $" doc:(2))];
                2 -> [label="mouse too $" doc:(3))];
        1 -> [label="cheese" doc:(1,2))];
                2 -> [label="$" doc:(1))];
                2 -> [label="too $" doc:(2))];
        1 -> [label="$" doc:(1,2,3))];
        1 -> [label="mouse" doc:(2,3))];
                2 -> [label="ate cheese too $" doc:(2))];
                2 -> [label="too $" doc:(3))];
        1 -> [label="too $" doc:(2,3))];
}
```

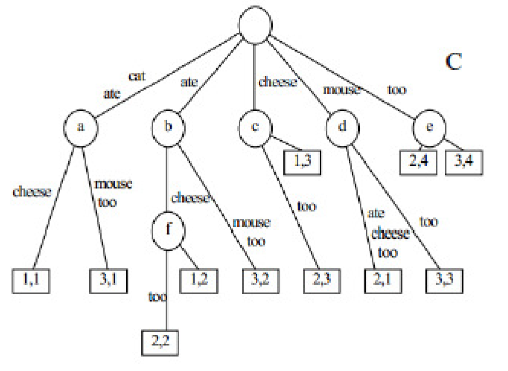


Figure.2. Suffix Tree using English example (A,B,C)

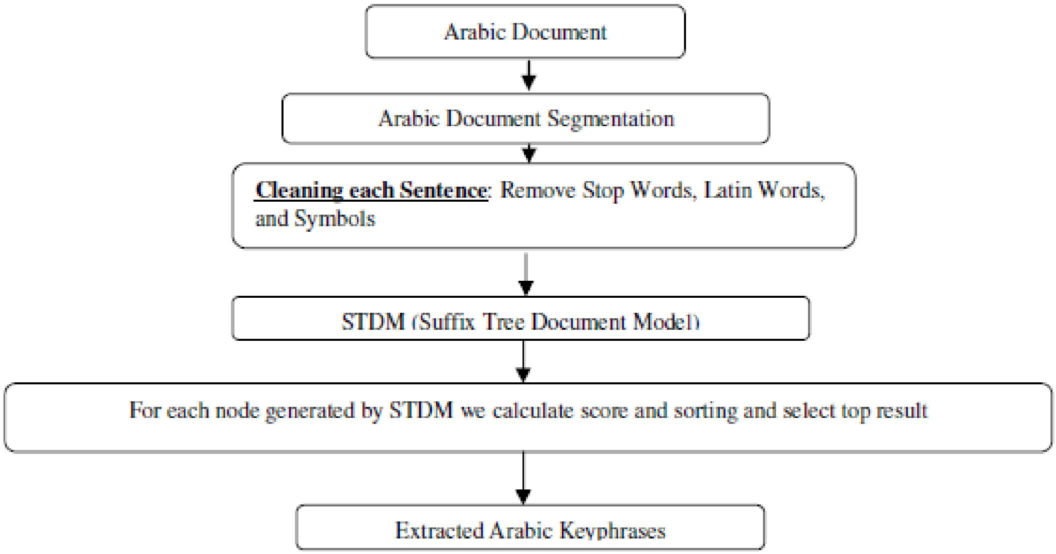

Arabic Document
Arabic Document Segmentation
Cleaning each Sentence: Remove Stop Words, Latin Words, and Symbols
STDM (Suffix Tree Document Model)
For each node generated by STDM we calculate score and sorting and select top result
Extracted Arabic Keyphrases

Figure 3. Description of Our Novel Approach for Arabic Keyphrases Extraction

A **Suffix Tree** of document d is a compact tree containing all suffixes of document d, this tree is presented by a set of nodes and leaves and labels. The label of a node in the tree is defined as the concatenation, in order, of the sub-strings labeling the edges of the path from the root to that node. Each node should have a score. The node can be ranked according to its score.

The score is relevant to:

a. The length of the label node words.
b. The number of the occurrence of word in document Term Frequency

Each node of the suffix tree is scored as:

$$S(B) = |B|*F(|P|) * \sum TFIDF\ (w_i) \qquad (1)$$

$$F(P) = \begin{cases} |P|, \text{ if } 3 \geq |P| \geq 1 \\ \\ 0, \text{ Otherwise} \end{cases} \qquad (2)$$

Where $|B|$ is the number of documents in B, and $|P|$ is the number of words making up the phrase P, w_i represents the words in P and TFIDF represent the Term Frequency Inverse Document Frequency for each word w_i in P.

Figure 3 summarize different step of our novel technique for extraction Keyphrases from each Arabic document.

3. SIMILARITY MEASURES

In this section we present the five similarity measures that were tested in [2] and our works [3][6], and we include these five measures in our work to effect the Arabic text document clustering.

3.1. Euclidean Distance

Euclidean distance is widely used in clustering problems, including clustering text. It satisfies all the above four conditions and therefore is a true metric. It is also the default distance measure used with the K-means algorithm.

Measuring distance between text documents, given two documents d_a and d_b represented by their term vectors $\vec{t_a}$ and $\vec{t_b}$ respectively, the Euclidean distance of the two documents is defined as:

$$D_E(\vec{t_a}, \vec{t_b}) = \left(\sum_{t=1}^{m} \left|w_{t,a} - w_{t,b}\right|^2\right)^{1/2}, \qquad (3)$$

where the term set is $T = \{t_1, ..., t_m\}$. As mentioned previously, we use the $tfidf$ value as term weights, that is $w_{t,a} = tfidf(d_a, t)$.

3.2. Cosine Similarity

Cosine similarity is one of the most popular similarity measure applied to text documents, such as in numerous information retrieval applications [7] and clustering too [8].

Given two documents $\vec{t_a}$ and $\vec{t_b}$, their cosine similarity is:

$$SIM_C(\vec{t_a},\vec{t_b}) = \frac{\vec{t_a}\cdot\vec{t_b}}{|\vec{t_a}|\times|\vec{t_b}|}, \quad (4)$$

where $\vec{t_a}$ and $\vec{t_b}$ are m-dimensional vectors over the term set $T = \{t_1,...,t_m\}$. Each dimension represents a term with its weight in the document, which is non-negative. As a result, the cosine similarity is non-negative and bounded between $[0,1]$. An important property of the cosine similarity is its independence of document length. For example, combining two identical copies of a document d to get a new pseudo document d_0, the cosine similarity between d and d_0 is 1, which means that these two documents are regarded to be identical.

3.3. Jaccard Coefficient

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two documents but are not the shared terms. The formal definition is:

$$SIM_J(\vec{t_a},\vec{t_b}) = \frac{\vec{t_a}\cdot\vec{t_b}}{|\vec{t_a}|^2 + |\vec{t_b}|^2 - \vec{t_a}\cdot\vec{t_b}} \quad (5)$$

The Jaccard coefficient is a similarity measure and ranges between 0 and 1. It is 1 when the $\vec{t_a} = \vec{t_b}$ and 0 when $\vec{t_a}$ and $\vec{t_b}$ are disjoint. The corresponding distance measure is $D_J = 1 - SIM_J$ and we will use D_J instead in subsequent experiments.

3.4. Pearson Correlation Coefficient

Pearson's correlation coefficient is another measure of the extent to which two vectors are related. There are different forms of the Pearson correlation coefficient formula. Given the term set $T = \{t_1,...,t_m\}$, a commonly used form is:

$$SIM_P(\vec{t_a},\vec{t_b}) = \frac{m\Sigma_{t=1}^{m} w_{t,a}\times w_{t,b} - TF_a\times TF_b}{\sqrt{\left[m\Sigma_{t=1}^{m} w_{t,a}^2 - TF_a^2\right]\left[m\Sigma_{t=1}^{m} w_{t,b}^2 - TF_b^2\right]}} \quad (6)$$

where $TF_a = \Sigma_{t=1}^{m} w_{t,a}$ and $TF_b = \Sigma_{t=1}^{m} w_{t,b}$

This is also a similarity measure. However, unlike the other measures, it ranges from -1 to +1 and it is 1 when $\vec{t_a} = \vec{t_b}$. In subsequent experiments we use the corresponding distance measure, which is $D_P = 1 - SIM_P$ when $SIM_P \geq 0$ and $D_P = |SIM_P|$ when $SIM_P \prec 0$.

3.5. Averaged Kullback-Leibler Divergence

In information theory based clustering, a document is considered as a probability distribution of terms. The similarity of two documents is measured as the distance between the two corresponding probability distributions. The Kullback-Leibler divergence (KL divergence), also called the relative entropy, is a widely applied measure for evaluating the differences between two probability distributions. Given two distributions P and Q, the KL divergence from distribution P to distribution Q is defined as:

$$D_{KL}(P \parallel Q) = P \log(\frac{P}{Q}) \tag{7}$$

In the document scenario, the divergence between two distributions of words is:

$$D_{KL}(\vec{t_a} \parallel \vec{t_b}) = \sum_{t=1}^{m} w_{t,a} \times \log(\frac{w_{t,a}}{w_{t,b}}). \tag{8}$$

However, unlike the previous measures, the KL divergence is not symmetric, i.e. $D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$. Therefore it is not a true metric. As a result, we use the averaged KL divergence instead, which is defined as:

$$D_{AvgKL}(P \parallel Q) = \pi_1 D_{KL}(P \parallel M) + \pi_2 D_{KL}(Q \parallel M), \tag{9}$$

where $\pi_1 = \frac{P}{P+Q}, \pi_2 = \frac{Q}{P+Q}$ and $M = \pi_1 P + \pi_2 Q$ For documents, the averaged KL divergence can be computed with the following formula:

$$D_{AvgKL}(\vec{t_a} \parallel \vec{t_b}) = \sum_{t=1}^{m} (\pi_1 \times D(w_{t,a} \parallel w_t) + \pi_2 \times D(w_{t,b} \parallel w_t)), \tag{10}$$

where $\pi_1 = \frac{w_{t,a}}{w_{t,a} + w_{t,b}}, \pi_2 = \frac{w_{t,b}}{w_{t,a} + w_{t,b}},$ and $w_t = \pi_1 \times w_{t,a} + \pi_2 \times w_{t,b}$.

The average weighting between two vectors ensures symmetry, that is, the divergence from document i to document j is the same as the divergence from document j to document i. The averaged KL divergence has recently been applied to clustering text documents, such as in the family of the Information Bottleneck clustering algorithms [9], to good effect.

4. EXPERIMENTS AND RESULTS

In our experiments (Figure 4), we used the Agglomerative Hierarchical algorithms as documents clustering methods for Arabic documents. The similarity measures do not directly fit into the algorithms, because smaller values indicate dissimilarity [3]. The Euclidean distance and the Averaged KL Divergence are distance measures, while the Cosine Similarity, Jaccard coefficient and Pearson coefficient are similarity measures. We apply a simple transformation to convert the similarity measure to distance values. Because both Cosine Similarity and Jaccard coefficient are bounded in $[0,1]$ and monotonic, we take $D = 1 - SIM$ as the corresponding distance value. For Pearson coefficient, which ranges from −1 to +1, we take $D = 1 - SIM$ when $SIM \geq 0$ and $D = |SIM|$ when $SIM \prec 0$.

For the testing dataset, we experimented with different similarity measures in two cases: in the first one, we apply the proposed method above to extract Keyphrases for the all documents in dataset and then cluster them. In the second case, we cluster the original documents. Moreover, each experiment was run for many times and the results are the averaged value over many runs. Each run has different initial seed sets; in the total we had 35 experiments for Agglomerative Hierarchical algorithm using 7 techniques for merging the clusters described below in the next section.

4.1. Agglomerative Hierarchical Techniques

Agglomerative algorithms are usually classified according to the inter-cluster similarity measure they use. The most popular of these are [11][12]:

- Linkage: minimum distance criterion

$$d_{A \to B} = \min_{\substack{\forall i \varepsilon A \\ \forall j \varepsilon B}} (d_{ij}) \qquad (11)$$

- Complete Linkage : maximum distance criterion

$$d_{A \to B} = \max_{\substack{\forall i \varepsilon A \\ \forall j \varepsilon B}} (d_{ij}) \qquad (12)$$

- Average Group : average distance criterion

$$d_{A \to B} = \underset{\substack{\forall i \varepsilon A \\ \forall j \varepsilon B}}{average} (d_{ij}) \qquad (13)$$

- Centroid Distance Criterion :

$$d_{A \to B} = \|C_A - C_B\| = \frac{1}{n_i n_j} \sum_{\substack{\forall i \varepsilon A \\ \forall j \varepsilon B}} (d_{ij}) \qquad (14)$$

- Ward : minimize variance of the merge cluster.

Jain and Dubes (1988) showed general formula that first proposed by Lance and William (1967) to include most of the most commonly referenced hierarchical clustering called SAHN (Sequential, Agglomerative, Hierarchical and nonoverlapping) clustering method. Distance between existing cluster k with nk objects and newly formed cluster (r,s) with nr and ns objects is given as:

$$d_{k \to (r,s)} = \alpha_r d_{k \to r} + \alpha_s d_{k \to s} + \beta d_{r \to s} + \gamma \left| d_{k \to r} - d_{k \to s} \right| \qquad (15)$$

The values of the parameters are given in the in Table 1.

Clustering Method	α_r	α_s	β	γ
Single Link	1/2	1/2	0	-1/2
Complete Link	1/2	1/2	0	1/2
Unweighted Pair Group Method Average (UPGMA)	$\frac{n_r}{n_r + n_s}$	$\frac{n_s}{n_r + n_s}$	0	0
Weighted Pair Group Method Average (WPGMA)	1/2	1/2	0	0
Unweighted Pair Group Method Centroid (UPGMC)	$\frac{n_r}{n_r + n_s}$	$\frac{n_s}{n_r + n_s}$	$\frac{-n_r n_s}{(n_r + n_s)^2}$	0
Weighted Pair Group Method Centroid (WPGMC)	1/2	1/2	-1/4	0
Ward's Method	$\frac{n_r + n_k}{n_r + n_s + n_k}$	$\frac{n_s + n_k}{n_r + n_s + n_k}$	$\frac{-n_k}{n_r + n_s + n_k}$	0

Table 1. The values of the parameters of the general formula of hierarchical clustering SAHN

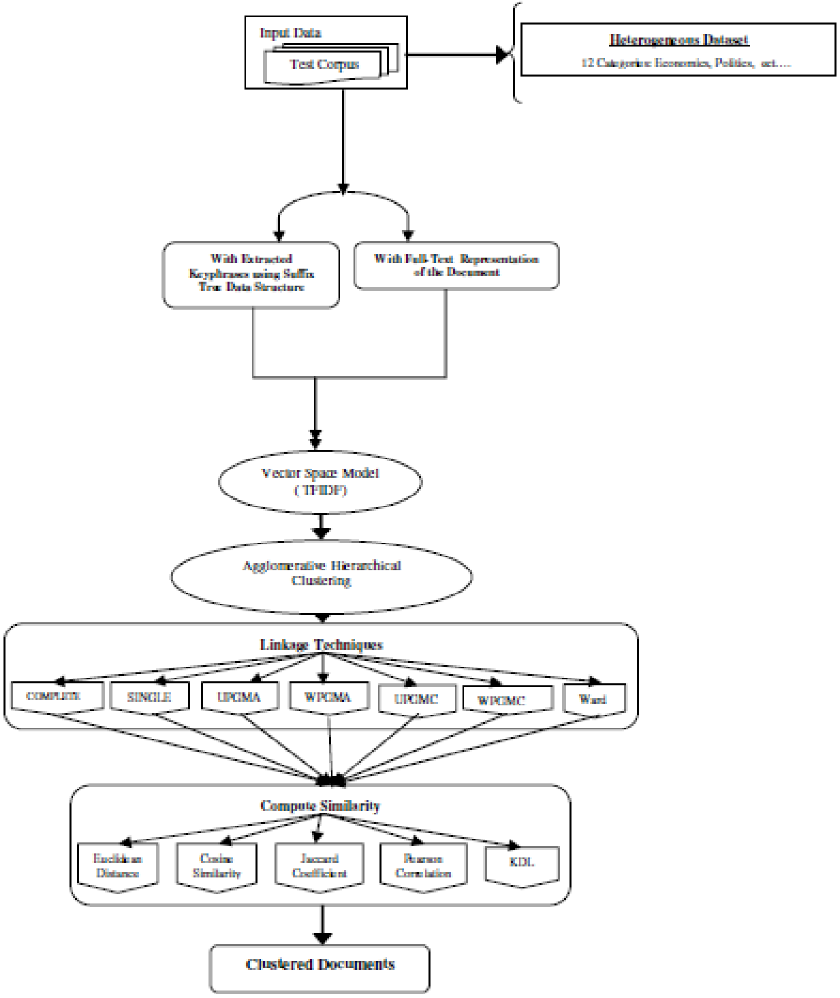


Figure 4. Description of Our Experiments

4.2. Dataset

The testing dataset [13] (Corpus of Contemporary Arabic (CCA)) is composed of 12 several categories, each latter contains documents from websites and from radio Qatar. A summary of the testing dataset is shown in Table 2. To illustrate the benefits of our proposed approach, we extracted the appropriate Keyphrases from the Arabic documents in our testing dataset using this approach, and we ranked terms by their weighting schemes $Tfidf$ and use them in our experiments.

Table 2. Number of texts and number of Terms in each category of the testing dataset

Text Categories	Number of Texts	Number of Terms
Economics	29	67 478
Education	10	25 574
Health and Medicine	32	40 480
Interviews	24	58 408
Politics	9	46 291
Recipes	9	4 973
Religion	19	111 199
Science	45	104 795
Sociology	30	85 688
Spoken	7	5 605
Sports	3	8 290
Tourist and Travel	61	46 093

4.3. Results

The quality of the clustering result was evaluated using two evaluation measures: purity and entropy, which are widely used to evaluate the performance of unsupervised learning algorithms [14], [15]. On the one hand, the higher the purity value (P (cluster) =1), the better the quality of the cluster is. On the other hand, the smaller the entropy value (E (cluster) =0), the better the quality of the cluster is.

The goal of these experiments is to evaluate our proposed approach to extract Keyphrases from Arabic documents then, the obtained results will be compared with our previous work [16].

Tables 3-5 show the average purity and entropy results for each similarity/distance measure with document clustering algorithms cited above.

The overall entropy and purity values, for our experiments using the Agglomerative Hierarchical algorithm with 7 schemes for merging the clusters, are shown in the tables 3, 4, 5 and 6.

Tables 3 and 5 summarize the obtained entropy scores in the all experiments, we remarked that the scores shown in the first one are generally worst than those in the second gotten using the extracted Keyphrases, but for those two tables the Agglomerative Hierarchical algorithm performs good using the COMPLETE, UPGMA, WPGMA schemes, and Ward function with the Cosine Similarity, the Jaccard measures and Pearson Correlation. The same behavior can be concluded from purity scores tables.

The above obtained results (shown in the different Tables) lead us to conclude that:

- For the Agglomerative Hierarchical algorithm, the use of the COMPLETE, UPGMA [16], WPGMA schemes, and Ward function as linkage techniques yield good results.

- Cosine Similarity, Jaccard and Pearson Correlation measures perform better relatively to the other measures.

- The obtained overall entropy values shown in the different tables prove that the extracted Keyphrases can make their topics salient and improve the clustering performance [16].

4.4. Discussion

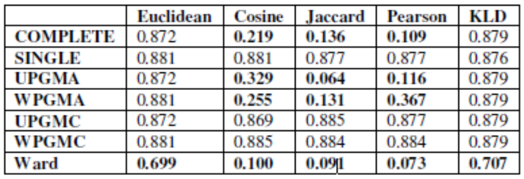
The above conclusions shows that, the use of the extracted Keyphrases instead of the full-text representation of documents is the best performing when clustered our Arabic documents that we investigated. There is also another issue that must be mentioned, our experiments show the improvements of the clustering quality and time. In the following, we make a few brief comments on the behavior of the all tested linkage techniques:

The COMPLETE linkage technique is non-local, the entire structure of the clustering can influence merge decisions. This results in a preference for compact clusters with small diameters, but also causes sensitivity to outliers.

The Ward function allows us to minimize variance of the merge cluster; the variance is a measure of how far a set of data is spread out. So the Ward function is a non-local linkage technique.

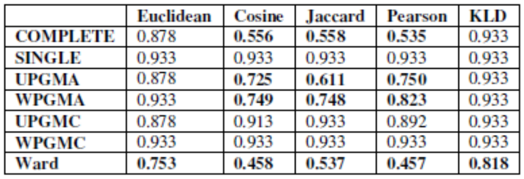
With the two techniques described above, a single document far from the center can increase diameters of candidate merge clusters dramatically and completely change the final clustering. That why these techniques produce good results than UPGMA [18], WPGMA schemes and better than the other all tested linkage techniques; because this merge criterion give us local information. We pay attention solely to the area where the two clusters come closest to each other. Other, more distant parts of the cluster and the clusters overall structure are not taken into account.

Table 3. Entropy Results using Agglomerative Hierarchical Algorithm with Full-Text Representation



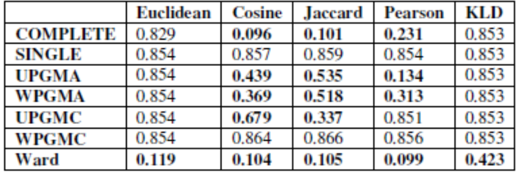
	Euclidean	Cosine	Jaccard	Pearson	KLD
COMPLETE	0.872	**0.219**	**0.136**	**0.109**	0.879
SINGLE	0.881	0.881	0.877	0.877	0.876
UPGMA	0.872	**0.329**	**0.064**	**0.116**	0.879
WPGMA	0.881	**0.255**	**0.131**	**0.367**	0.879
UPGMC	0.872	0.869	0.885	0.877	0.879
WPGMC	0.881	0.885	0.884	0.884	0.879
Ward	**0.699**	**0.100**	**0.091**	**0.073**	**0.707**

Table 4. Purity results using Agglomerative Hierarchical Algorithm with Full-Text Representation



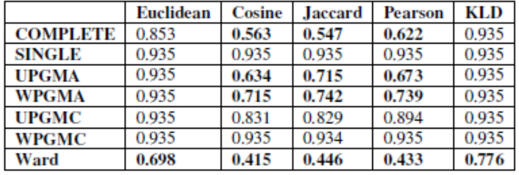
	Euclidean	Cosine	Jaccard	Pearson	KLD
COMPLETE	0.878	**0.556**	**0.558**	**0.535**	0.933
SINGLE	0.933	0.933	0.933	0.933	0.933
UPGMA	0.878	**0.725**	**0.611**	**0.750**	0.933
WPGMA	0.933	**0.749**	**0.748**	**0.823**	0.933
UPGMC	0.878	0.913	0.933	0.892	0.933
WPGMC	0.933	0.933	0.933	0.933	0.933
Ward	**0.753**	**0.458**	**0.537**	**0.457**	**0.818**

Table 5. Entropy Results using Agglomerative Hierarchical Algorithm with extracted Keyphrases



	Euclidean	Cosine	Jaccard	Pearson	KLD
COMPLETE	0.829	**0.096**	**0.101**	**0.231**	0.853
SINGLE	0.854	0.857	0.859	0.854	0.853
UPGMA	0.854	**0.439**	**0.535**	**0.134**	0.853
WPGMA	0.854	**0.369**	**0.518**	**0.313**	0.853
UPGMC	0.854	**0.679**	**0.337**	0.851	0.853
WPGMC	0.854	0.864	0.866	0.856	0.853
Ward	**0.119**	**0.104**	**0.105**	**0.099**	**0.423**

Table 6. Purity results using Agglomerative Hierarchical Algorithm with extracted Keyphrases



	Euclidean	Cosine	Jaccard	Pearson	KLD
COMPLETE	0.853	**0.563**	**0.547**	**0.622**	0.935
SINGLE	0.935	0.935	0.935	0.935	0.935
UPGMA	0.935	**0.634**	**0.715**	**0.673**	0.935
WPGMA	0.935	**0.715**	**0.742**	**0.739**	0.935
UPGMC	0.935	0.831	0.829	0.894	0.935
WPGMC	0.935	0.935	0.934	0.935	0.935
Ward	**0.698**	**0.415**	**0.446**	**0.433**	**0.776**

5. CONCLUSIONS

To conclude, this investigation found that using the full-text representation of Arabic documents; the Cosine Similarity, the Jaccard and the Pearson Correlation measures have comparable effectiveness and performs better relatively to the other measures for all techniques cited above to find more coherent clusters.

Furthermore, our experiments with different linkage techniques yield us to conclude that COMPLETE, UPGMA, WPGMA and Ward produce efficient results than other linkage techniques. A closer look to those results, show that the Ward technique is the best in all cases, although the two other techniques are often not much worse.

Instead of using full-text as the representation for Arabic documents, we use our novel approach based on Suffix Tree algorithm as Keyphrases extraction technique to eliminate the noise in the documents and select the most salient sentences. Furthermore, Keyphrases extraction can help us to overcome the varying length problem of the diverse documents.

In our experiments using extracted Keyphrases, we remark that again the Cosine Similarity, the Jaccard and the Pearson Correlation measures have comparable effectiveness to produce more coherent clusters than the Euclidean Distance and averaged KL Divergence; on the other hand, the good results are detected when using COMPLETE, UPGMA, WPGMA and Ward as linkage techniques.

Finally, we believe that the novel Keyphrases extraction approach and the comparative study presented in this paper should be very useful to support the research in the field any Arabic Text Mining applications; the main contribution of this paper is three manifolds:

1. We must mention that our experiments show the improvements of the clustering quality and time when we use the extracted Keyphrases with our novel approach,

2. Cosine Similarity, Jaccard and Pearson Correlation measures are quite similar to find more coherent clusters,

3. Ward technique is effective than other linkage techniques to produce more coherent clusters using the Agglomerative Hierarchical algorithm.

REFERENCES

[1] Turney, P. D. (2000) Learning Algorithms for Keyphrase Extraction, Information Retrieval, 2, 303-336.

[2] A.Huang, “Similarity Measures for Text Document Clustering”, NZCSRSC 2008, April 2008, Christchurch, New Zealand.

[3] H.Froud, A.Lachkar, S. Ouatik, and R.Benslimane (2010). Stemming and Similarity Measures for Arabic Documents Clustering. 5th International Symposium on I/V Communications and Mobile Networks ISIVC, IEEE Xplore.

[4] P.Berkhin, “Survey of clustering data mining techniques”, Technical report, Accrue Software, San Jose, California, 2002. http://citeseer.nj.nec.com/berkhin02survey.html.

[5] K.Sathiyakumari, G.Manimekalai, V.Preamsudha, M.Phil Scholar, “A Survey on Various Approaches in Document Clustering”, Int. J. Comp. Tech. Appl., Vol 2 (5), 1534-1539, IJCTA | SEPT-OCT 2011.

[6] H. Froud, A. Lachkar, S. Alaoui Ouatik, “A Comparative Study Of Root-Based And Stem-Based Approaches For Measuring The Similarity Between Arabic Words For Arabic Text Mining Applications”, Advanced Computing: An International Journal (ACIJ), Vol.3, No.6, November 2012.

[7] R. B. Yates and B. R. Neto.”Modern Information Retrieval”. ADDISON-WESLEY, New York, 1999.

[8] B. Larsen and C. Aone." Fast and Effective Text Mining using Linear-time Document Clustering". In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999.

[9] N. Z. Tishby, F. Pereira, and W. Bialek. "The Information Bottleneck Method". In Proceedings of the 37th Allerton Conference on Communication, Control and Computing, 1999.

[10] Khoja, S. and Garside, R. "Stemming Arabic Text". Computing Department, Lancaster University, Lancaster, 1999.

[11] Daniel Müllner, "Modern hierarchical, agglomerative clustering algorithms", Lecture Notes in Computer Science, Springer (2011).

[12] Teknomo, Kardi. (2009) Hierarchical Clustering Tutorial.
http://people.revoledu.com/kardi/tutorial/clustering/

[13] L. Al-Sulaiti , E.Atwell, "The Design of a Corpus of Contemporary Arabic", University of Leeds.

[14] Y. Zhao and G. Karypis."Evaluation of Hierarchical Clustering Algorithms for Document Datasets". In Proceedings of the International Conference on Information and Knowledge Management, 2002.

[15] Y. Zhao and G. Karypis."Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering". Machine Learning, 55(3), 2004.

[16] H. Froud, A. Lachkar, S. Alaoui Ouatik, "Arabic Text Summarization Based on Latent Semantic Analysis to Enhance Arabic Documents Clustering", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.1, January 2013.

[17] H. Froud, A. Lachkar, "Agglomerative Hierarchical Clustering Techniques for Arabic Documents", submitted to the Second International Workshop On Natural Language Processing (NLP 2013), KTO Karatay University, June 07 ~ 09, 2013, Konya, Turkey.

[18] Michael Steinbach , George Karypis, andVipin Kumar, "A comparison of document clustering techniques,"In KDD Workshop on Text Mining, 2002.

[19] P. Turney. Coherent keyphrase extraction via web mining. Technical Report ERB-1057, Institute for Information Technology, National Research Council of Canada, 1999.

[20] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing.

[21] Weiner, P. Linear pattern matching algorithms. In Proceedings of the 14th Annual Symposium on Foundations of Computer Science (FOCS'73), 1-11, 1973.

[22] Landau, G. M. and Vishkin, U. Fast Parallel and serial approximate string matching. Journal of Algorithms, 10, 157-169, 1989.

[23] Ehrenfeucht, A. and Haussler, D. A new distance metric on strings computable in linear time. Discrete Applied Math, 40, 1988.

[24] Rodeh, M., Pratt, V. R. and Even, S. Linear algorithm for data compression via string matching. Journal of the ACM, 28(1):16-24, 1981.

[25] O. Zamir and O. Etzioni, "Grouper: a dynamic clustering interface to Web search results", Computer Networks, 31(11-16), pp. 1361-1374, 1999.

Authors

Miss. Hanane Froud Phd Student in Laboratory of Information Science and Systems, ECOLE NATIONALE DES SCIENCES APPLIQUÉES (ENSA-FEZ),University Sidi Mohamed Ben Abdellah (USMBA),Fez, Morocco. She has also presented different papers at different National and International conferences.

Issam SAHMOUDI received his Computer Engineering degree from ECOLE NATIONALE DES SCIENCES APPLIQUÉES, FEZ (ENSA-FEZ). Now he is PhD Student in Laboratory of Information Science and Systems LSIS, at (E.N.S.A), Sidi Mohamed Ben Abdellah University (USMBA), Fez, Morocco. His current research interests include Arabic Text Mining Applications: Arabic Web Document Clustering and Browsing, Arabic Information and Retrieval Systems, and Arabic web Mining.

Pr. Abdelmonaime LACHKAR received his PhD degree from the USMBA, Morocco in 2004, He is Professor and Computer Engineering Program Coordinator at (E.N.S.A, FES), and the Head of the Systems Architecture and Multimedia Team (LSIS Laboratory) at Sidi Mohamed Ben Abdellah University, Fez, Morocco. His current research interests include Arabic Natural Language Processing ANLP, Arabic Web Document Clustering and Categorization, Arabic Information Retrieval Systems, Arabic Text Summarization, Arabic Ontologies development and usage, Arabic Semantic Search Engines (SSEs).