Comparing between Arabic Text Clustering using K Means and K Mediods

Mahmud S. Alkoffash
Department of Software Engineering,
Prince Abdullah bin Gazi faculty of Information Technology,
Al-Balqa Applied University
Salt, Jordan

ABSTRACT

In this study we have implemented the Kmeans and Kmediods algorithms in order to make a practical comparison between them. The system was tested using a manual set of clusters that consists from 242 predefined clustering documents. The results showed a good indication about using them especially for Kmediods. The average precision and recall for Kmeans compared with Kmediods are 0.56, 0.52, 0.69 and 0.60 respectively. we have also extract feature set of keywords in order to improve the performance, the result illustrates that two algorithms can be applied to Arabic text, a sufficient number of examples for each category, the selection of the feature space, the training data set used and the value of K can enormously affect the accuracy of clustering.

Key words: Arabic Text Clustering, Data mining, Kmeans, Kmediods

1. INTRODUCTION

Arabic language is one of the six international languages that are used by more than 300 millions humans all over the world. It differs from other languages in which it is written from right to left. The alphabet of Arabic language is composed of 28 letters; also it contains other symbols that reach to 90 or more. [1, 2, 3, 4, 14]

Most of the Arabic words are found from the list of Arabic language roots which could be roots of three, four, five or six letters. Sentences in Arabic language consist of nouns, verbs and letters(Pronouns and Preposition and Conjunction etc.), that the noun and the verb have a root while the letter doesn't have so we cancel it in our program.[1, 6, 7, 14]

Arabic is a challenging language for information retrieval (IR) for a number of reasons. First, orthographic variations are prevalent in Arabic; certain combinations of characters can be written in different ways. Second, Arabic has a very complex morphology. Third, broken plurals are common. Broken plurals are somewhat like irregular English plurals except that they often do not resemble the singular form as closely as irregular plurals resemble the singular in English. Because broken plurals do not obey normal morphological rules, they are not handled by exiting stemmers. Fourth, Arabic words are often ambiguous due to the trilateral root system. In Arabic, a word is usually derived from a root, which usually contains three letters. In some derivations one or more of the root letters may be dropped, rending many Arabic words highly ambiguous with one another. Fifth, short vowels are omitted in written Arabic. Six, synonyms are widespread, perhaps because a variety in expression is appreciated as part of a good writing style by Arabic speakers. [14]

Those previous mentioned problems make exact keyword matching inadequate for Arabic retrieval. In our work we focus on Arabic retrieval especially on Arabic language clustering using Kmeans and Kmediods because there is no enough work proposed in this field.

Kmeans and Kmediods are one of the most common methods used in Automatic Clustering especially in Data Mining System. In order to implement this system several steps are carried out. First of all the distribution process of the corpus is carried out to construct a set of files (training data). Then we store the files in a database file to use them in the future work. The next step is to build the inverted file and compute similarity. Finally, Kmeans and Kmediods are operated based on the previous subsystems.

2. RELATED WORK

The goal of cluster ensemble is to combine the cluster results of multiple cluster algorithms to obtain better quality, which is very difficult because of inconsistency among different cluster algorithms and noise in experimental data. Even though many clustering algorithms have been developed [5, 8], not much work is done in cluster ensemble in data mining and machine learning literature compared with classification ensemble method. K.M.Hammouda ,2002 proposed an adaptive meta-clustering approach for combining different cluster results[4]. P.Berkhin, 2002 proposed a hypergraph-partitioned approach to combine different clustering results [9]. It is very difficult to combine different clusters in an optimal way. To combine clusters we must consider this as a natural phenomenon because each object has various characteristics, for example people in the university can be classified according to gender nationality and faculty. Most clustering knowledge are compensative not competitive this makes a combination of different algorithms very difficult [5]. In the proposed research we must take care of symmetrical and unbiased consensus with regard to every object.

A new mechanism is needed to combine different cluster results. Zamir, 1999 proposed to use a suffix tree to find the maximum word sequences (phases) between two documents [6]. Two documents sharing more common phrases are more similar to each other. Bakus, Hussin, and Kamel used a hierarchical phrase grammar extraction procedure to identify phrases from documents and used these phrases as features for document clustering[10].Mladenic and Grobelnik used a Naïve Bayesian method to classify documents base on word sequences if different length[14]. Many other researcher worked in this field but the results were not good due to high noise and similarity between training data.

3. ARABIC STEMMER

Arabic word formation is based on an abstraction, namely, the root. These roots join with various vowel patterns to form simple nouns and verbs to which affixes can be attached for more complicated derivations.

As we mentioned previously an already implemented stemmer has been used to alter all words to their roots. The proposed algorithms based on suffix and prefix removal of the Arabic word and find the associated patterns (most known) that match the resulting word. Thus, in order for any Arabic information retrieval system to work well, it needs Normalization to handle the different shapes of letters. Without normalization and stemming there is a strong likelihood of mismatch between the form of a word in a document and the forms found in other documents.

We use in our work a light stemmer which was developed by Khoja [1] to improve the retrieval effectiveness for Arabic language. The proposed algorithm works by executing the following steps:

1. Removing Stop words

2. Remove all prefixes indicated that are stored in database table:

a. *That have length=3, (ex: "وال","كال","مست") if the remaining word length >=3.*
b. *That have length=2, (ex:"سي","ست") if the remaining word length >=3.*

3. Remove all suffixes indicated that are stored in database table:

a. *That have length=5, (ex: "كموها") if the remaining word length >=3.*
b. *That have length=3, (ex:"هما") if the remaining word length >=3.*
c. *Repeat step b, (ex:"ناهما").*
d. *That have length=1, (ex:"ة") if the remaining word length >=3.*

4. Repeat step 1 for prefixes of length 1 (ex: "ي","أ").

5. Find the pattern corresponding to the resulting word.

6. If no match found then return the original word. (e.g.: "تكنولوجيا")

4. INDEXING

For small collections of documents it may be possible for an IR system to assess each document in turn, deciding whether or not. However, for larger collections, especially in interactive systems, this becomes impractical. Hence it is usually necessary to prepare an easily accessible representation for the raw document collection; one that makes it easy to target those documents that are most likely to be relevant, for example, those documents that contain at least one word that appears in the documents [8, 9]. This transformation from a document text to a ***representation*** of a text is known as ***indexing*** the documents.

We use indexing approach because it is fast and flexible to further improvement. There are a variety of indexing techniques such as **Signature Files, that uses** the hash techniques to produce an index, this technique was popular in past [4], **Patricia (PAT) Trees** which is a binary digital tree where the individual bits of the keys where the decision on the branching [4] and **Inverted Files** that is described as a word-oriented mechanism for indexing a text collection in order to speed up the searching task [2]. The inverted file contains, for each term in the lexicon, an inverted list that stores a list of pointers to all occurrences of that term in the main text, where each pointer is, in effect, the number of a document in which that term appears.

Structure of Inverted File usually contains ***Vocabulary*** that refers to the set of all distinct words in the text and Occurrences which refers to the lists containing all information necessary for each word of the vocabulary (text position, frequency, documents where the word appears, etc). In our work we use the inverted file to store the result of the indexing process.

5. SIMILARITY MEASURES

A Similarity Measure is a function that computes the degree of similarity between two vectors (documents) [7, 11, 12]. Using a similarity measure, a set of documents can be compared against each other or against queries and the most similar documents are returned. Many different ways to do that some of them are *Inner Product(dot product) that computes* the similarity between vectors for the documents x_i and y_i can be computed as the vector inner product *and Cosine Similarity* which measures cosine of angle between document-document (or document-query) vector . In our research we use another method to compute a similarity between the documents

$$Sim(Di, Dj) = \frac{SimilarTermInDjtoDi}{TermsofDi}$$

- Where Dj and Di are two documents in corpus

Figure [2]: Similarity equation

6. METHODOLOGY

In our work several steps are carried out as we mentioned previously, figure [1] summarizes them. In the first step as you see the corpus is read in which the training data is divided into a set of documents, then we build the inverted file that will contain the files, terms, and their frequencies. After that the similarity between documents is computed in order to use it in the next step of clustering. Kmeans and Kmediods are used to build clusters in the final step.

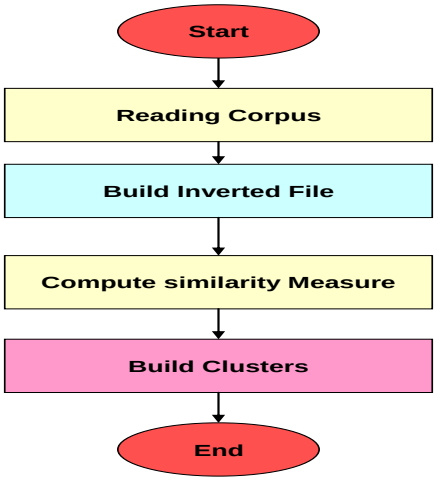

Figure [1]: The document clustering system.

6.1 Reading Corpus

First of all we will decide to determine the number of documents that we shall work on and give every document a name through that the size and the file name is stored. After that the reading document is divided into meaning words.

6.2 Build Inverted File

Inverted file was built to contain file name, terms, and frequency of each one. The stored words were manipulated using the light stemmer algorithm in aim to improve performance of search and store by removing redundant words.

6.3 Compute Similarity between documents

To compute the similarity between the stored files equation in figure [2] was used. The results are stored in special file in aim to utilize them in clustering process.

6.4 Build clusters

In this step we distributed the documents to a group of clusters, so every cluster includes an *exclusive* number of documents in which each documented refers to only one cluster. For building the clusters we use two methods, which are Kmeans and Kmediods that work as ulterior:

K-means steps:

1. Input k, where K is the number of suggested clusters such that 1≤K≤ Number of documents.
2. Select k centers randomly using the common Linear Congruential generators LCG[13].
3. Determine numbers of trials to test Clusters, Such that number of trials is larger than or equal one.
4. Assign each document to the closest cluster based on the distance between centers and the document. Distance is computed based on the similarity between center document and selected document, where the most similar center is less distance or the closest
5. Compute square error between centers and selected documents.
6. Store the results in the database
7. Compute average for each cluster to determine the K centers.
8. Repeat steps 4-7 until the number of trials is reached.

Kmediods steps:

1. Input k, where K is the number of suggested clusters such that 1≤ K ≤ Number of documents.
2. Select k centers randomly using LCG generator which was mentioned previously.
3. Determine numbers of trials to test Clusters, Such that number of trials is larger than or equal one.
4. Assign each document to the closest cluster based on the distance between centers and the document. Distance is computed based on the similarity between center document and selected document, where the most similar center is the least distance or the closest
5. Compute square error between centers and selected documents.
6. Store the results in the database
7. Select k new centers randomly using LCG generator
8. Repeat steps 4-7 until the number of trials is reached

7. RESULTS

Evaluation is a very important step in any retrieval system, to satisfy the success of that retrieval system. In this research we use the recall and precision measurers for evaluation. The system was evaluated using 242 predefined documents that were clustered manually. Table [1] and table [2] show the results of clustering process. Results show Kmediods is better than Kmeans due to the chance that is given for several files in Kmediods to become a center for a given cluster.

Kmeans Results		
Number of Clusters	**Average Recall**	**Average Precision**
2	**0.23**	**0.8**
3	**0.49**	**0.6**
4	**0.63**	**0.5**
5	**0.43**	**0.6**
6	**0.64**	**0.46**
7	**0.73**	**0.43**
Average	**0.525**	**0.565**

Table [1]: Kmeans training results

Kmediods Results		
Number of Clusters	**Average Recall**	**Average Precision**
2	**0.48**	**0.9**
3	**0.57**	**0.86**
4	**0.693**	**0.6**
5	**0.52**	**0.7**
6	**0.69**	**0.46**
7	**0.67**	**0.6**
Average	**0.60**	**0.69**

Table [2]: Kmediods training results

The results in figure [3] and figure [4] show that our works are accepted since the curve is located in the desired **equality error point.**

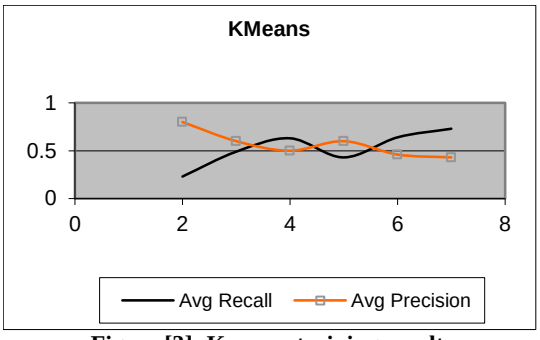

Figure [3]: Kmeans training results

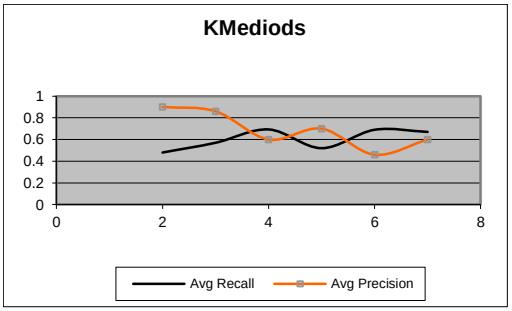

Figure [4]: Kmediods training results

8. CONCLUSION

Manipulating large corpus may give results that are more nearby to the manual one. Clustering environment is more unbiased than manual due to its dependability on the system rather than user opinion. Most of the errors or weakness that appear in Arabic retrieval systems due to the strength of language itself that contains several features not existed in any other one. The problem of Kmeans and Kmediods are represented by Selecting Initial Points, Problems of differing Sizes, Densities, and shapes and Outliers data.

9. REFERNCES

[1] Aljlayl, Mohammed, Frieder, Ophir," 2002. On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach". CILM'02, November 4-9, M clean, Virginia, USA. ACM 1-58113-492-4/02/0011.

[2] A.K.Jain, M.N.M urty,P.J.Flynn, 1999. Data Clustering : a review, ACM computing surveys(CSUR),v.31n.3,P.264-323,sept.

[3] D. Mladenic and M.Grobelnik,"Word Sequenc as Features in Text-Learning," 1999. In Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK-98),Ljubljana,Slovenia.

[4] Ghosh , Joydeep, 2003. "Scalable Clustering",The Handbook of Data Mining, Nong Ye(Ed), Lawrence Erlbam Assoc. chapter 10,pp.247-278.

[5] G.Salton, 1989. "Automatic Text Processing: TheTransaction, Analysis, and Retrieval of Information by computer," Addison-Wesley.

[6] Jiawei Han and Micheline Kamber, 2006, Data Mining: Concept and Techniques chapter 7, Depertment of Computer science ,University of Illinois at Urbana-chapaign: www.cs.uiuc.edu/~hanj

[7] J.Bakus, M.F.Hussin, and M.Kamel,2002."A SOM-Based Document Clustering using phrases,"In proceeding of the 9th International Conference on Naural Information processing (ICONIP'02),vol.5,pp.2212-2216.

[8] K.M.Hammouda, Web Mining : Identifying Document Structure for Web Document Clusering, Master's Thesis, 2002. Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada.

[9] Orengo et al Binoformatics-Genes,2003. Protein & Computers.BIOS,ISBN: 1-85996-054-5.

[10] O.Zamir, 1999. Clustering Web Document: A phrase-Based Method for Group Search Engine Result, ph.D. dissertation, Dept.Computer Science & Engineering, Univ.of Washington.

[11] P.Berkhin,2002. Survey of clustering data mining techniques. Technical report, Accrue soft ware.San Jose,CA.

[12] Ricard Baeza-Yates and Berthier Ribeire-1999.Neto.Modern Onformation Retrieval ,January,.

[13] Stephen K.Park and Keith W.Miller 1988. Random Number Generators:Good ones are hard to find communications of the ACM,31(10):1192-1201.

[14] Xu,Jinxi, Fraser, Alexander, Weischedel, Ralph, 2002. "Empirical Studies in Strategies for Arabic Retrieval".SIGIR'02,August 11-15, , Tampere, Finland.ACM 1-58113-561-0/02/0008.