Applying Clustering of Hierarchical K-means-like Algorithm on Arabic Language

Sameh H. Ghwanmeh

***Abstract*—**In this study a clustering technique has been implemented which is K-Means like with hierarchical initial set (HKM). The goal of this study is to prove that clustering document sets do enhancement precision on information retrieval systems, since it was proved by Bellot & El-Beze on French language. A comparison is made between the traditional information retrieval system and the clustered one. Also the effect of increasing number of clusters on precision is studied. The indexing technique is Term Frequency * Inverse Document Frequency (TF * IDF). It has been found that the effect of Hierarchical K-Means Like clustering (HKM) with 3 clusters over 242 Arabic abstract documents from the Saudi Arabian National Computer Conference has significant results compared with traditional information retrieval system without clustering. Additionally it has been found that it is not necessary to increase the number of clusters to improve precision more.

Keywords—Hierarchical K-mean like clustering (HKM), K-means, cluster centroids, initial partition, and document distances

I. INTRODUCTION

INFORMATION is of strategic importance for many businesses and governmental agencies as well as for every citizen [7]. The amount of online text data has grown tremendously due to the popularity of the Internet and the World Wide Web. As a result, there is an overriding need to provide effective content-based text retrieval, search and querying capabilities [11]. A classical information retrieval system returns a list of documents to a user query. The answer list is often so long that users cannot explore all the documents retrieved. A classification of the retrieved documents allows to thematically organize them and to improve precision [6].

Text Classification is the problem of grouping text documents into classes or categories [8]. Consider the problem of automatically classifying text documents. This problem is of great practical importance given the massive volume of online text available through the World Wide Web, Internet news feeds, electronic mail, corporate databases, medical patient records and digital libraries. Existing statistical text learning algorithms can be trained to approximately classify documents, given a scent set of labeled training examples [3].

Classification of documents is an increasingly important tool for handling the exponential growth in available online texts. Many algorithms have been suggested for this task in the past few years. The most common approaches start by evaluating the co-occurrence matrix of words versus documents, given document training data [5].

Clustering of pages is performed in two phases. In the first phase, high-probability clusters are identified using a conservative threshold value. In the second phase, any remaining single pages are combined into nearby clusters, using a more forgiving threshold value. This approach is intended to handle the common situation where a document contains short 1- or 2-page sequences which differ significantly from nearby pages in the same document, but are less similar to other documents. Examples of this include first or last pages, and pages consisting mostly of diagrams or tables [4].

In recent years we have seen a tremendous growth in the number of text document collections available on the Internet. Automatic text categorization, the process of assigning unseen documents to user-defined categories, is an important task that can help in the organization and querying of such collections [11].

The enormous increase in the amount of available textual data has resulted in various new algorithms for automatic text classification [8]. In this paper we will apply an algorithm for automatic text classification, which is K-means-like Algorithm on Arabic language.

Category is a powerful tool to manage a large number of text documents. By grouping text documents into a set of categories, it is possible for us to efficiently keep or search for information we need. At this point, the structure of categories, called *category model*, becomes one of the most important factors that determine the efficiency of organizing text documents. In the past, two traditional category models, called flat and hierarchical category models, were applied in organizing text documents [10]. as we will see later we will use the second category model (hierarchical category models) in the first phase of clustering which is the initial partitions.

The quality of the document list produced after classification depends on the number of clusters. Indeed, K-Means like methods require some *a-priori* decisions about the number of clusters. It is critical but not so easy to determine the number of clusters even if we have shown that it could be computed effectively according to query size [6].

There are three types of features; *Irrelevant features*, which can be ignored without degradation in the classifier performance, *strongly relevant features* that contain useful

Received Dec 2004; Accepted Nov 2005.

Sameh. H. Ghwanmeh is the Director, Computer and Information Centre, and faculty member at Department of Computer Engineering, Yarmouk University 211-63, Irbid, Jordan, (e-mail: sameh@yu.edu.jo)

information such that if removed the classification accuracy will degrade and *weakly relevant features* that contain information useful for the classification, but are unnecessary given that some other words are present in an instance [2].

The most straightforward basic term to be used to represent a text document is a word. For text classification, in many cases a word is a meaningful unit of little ambiguity even without considering context and it is has been successfully used many times. In this case the bag-of-terms representation is in fact a bag-of-words [7].

Most of previous works on text classification focus on classifying text documents into a set of flat categories. The task is to classify documents into a predefined set of categories (or classes) where there are no structural relationships among these categories [10].

Recent approaches to text classification have used two different first-order probabilistic models for classification, both of which make the naive Bayes assumption. Some use a multi-variate Bernoulli model, that is, a Bayesian Network with no dependencies between words and binary word features. Others use a multinomial model, that is, a uni-gram language model with integer word counts [1]. Thanaruk Theeramunkong and Verayuth Lertnattee found that the centriods based classifier has better results than K-NN and NB, and CB1 is best work on the WebKB.

Peng Dai, Uri Iurgel, and Gerhard Rigoll found that appropriate combination of different type of features can work better than a single type of feature. The idea of combining different form of features can not only be used in text classification tasks but also has been used in speech recognition [7].

In most cases, the hierarchical-based classification performs better than the flat-based classification. Moreover, an interesting observation is that classifying on the worse dimension before the better one yields a better result [10].

Patrice Bellot & Marc El-Bèze found that best results are not always obtained with a large number of clusters (at least when the number of clusters is not too large). During Amaryllis'99, the number of retrieved documents for each query was limited to 250. This number is too small (smaller than for TREC) and the evaluation method they use (ranking clusters according to their precision) favors a great number of clusters [6].

Figure 1 shows results obtained by Patrice Bellot & Marc El-Bèze after classification according to the number of clusters or without classification. The quality of the results obtained is similar to those reported in French corpora [6].

The pioneering work on the application of supervised decision trees to natural language concerned probabilistic language modeling. Decision trees were also employed to syntactically tag a word according to the surrounding text. They were applied to the classification of newspaper articles in some predefined classes [6].

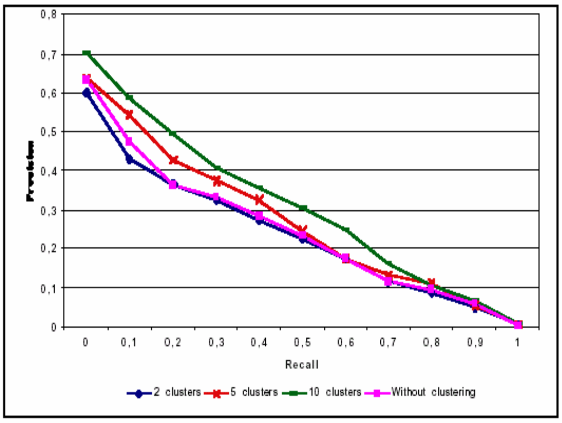

Fig. 1 Precision at several recall levels (HKM) obtained by Patrice Bellot & Marc El-Bèze

II. RESEARCH METHODOLOGY

A. Outline

In this study an algorithm for text clustering has been applied, which is a combination of Hierarchical and K-means-like Algorithm (HMK), after that we will compare it with a classical information retrieval system, we will also see the effect of number of clusters on precision on our corpora. However it is worth to understand some important issues before that. What is the difference between classification and clustering?

Classification is a supervised criteria were clustering is unsupervised one, that is the classification techniques uses some training data to classify others were clustering techniques the training data is unknown.

The training data (observations, measurements, etc.) in classification are accompanied by labels indicating the class of the observations and new data is classified based on the training set [11].

There are two types of clustering [9]:

Hard Clustering: where each object is in one and only one cluster

Soft Clustering: Each object has a probability of being in each cluster.

B. Clustering

Grouping of similar observations into separate clusters is one of the fundamental tasks in exploratory data analysis. Depending on the form of the data, the clustering can be done by using central or pair wise clustering techniques [11]. In this study this technique will be implemented as the second part.

- Central clustering techniques minimize the average distance between an observation and its cluster center. Thus, the clustering solution can be described by means of cluster centroids [11].
- The other possibility to cluster observations is pair-wise clustering, where the dissimilarities between the

observations are exploited. In pair-wise clustering, the clusters are formed by minimizing the average dissimilarity between the observations within the same cluster.

C. K-Means

K-means algorithm is one of the most widely used central clustering techniques. In the algorithm, the data set is divided iteratively into k clusters by minimizing the average squared Euclidean distance between the observation and its cluster center. The algorithm starts with assigning k observations as initial cluster centroids and assigning all the observations to the nearest cluster. After this new clustering, the centroids are calculated as means of the observations belonging to that cluster. The observations are assigned again to the new clusters, and new cluster centroids are once again calculated. This iteration procedure is continued until the centroids stabilize [11].

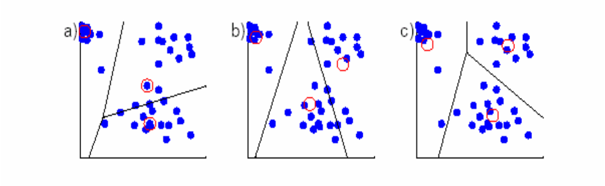

Fig. 2 illustrates the K-means algorithm with 40 observations and K=3: a) observations (blue dots), the initial cluster centroids (red circles) and cluster boundaries, b) cluster centroids and boundaries after the first iteration, c) final cluster centroids and boundaries

K- Means have the following properties [9]:

- Each data point xi is encoded by its nearest cluster center centerj.
- When the algorithm stops, the partial derivative of the Distortion with respect to each center attribute is zero.
- Each center is the centroids of its cluster.

But we note that it is very difficult to choose k (the number of centroids for each cluster), this is the most important drawback of this clustering technique.

Hierarchical and K-Means-Like Algorithm

This method is a combination between hierarchical and cluster-based method [6]. The advantage of using the cluster-based (K-Mean-Like) is that it allows texts to be classified quickly, so if our purpose is speed we have to choose this method.

The algorithm starts with an initial partition and then we apply the other documents and relocate them iterately until we have the final partition which have a property that it is not allowed to transfer any document from one cluster to another. The initial partition is built using hierarchical classification by using some of the documents where retrieved in a previous step. We use subset of the document in the hierarchical classification to reduce computational cost because we have to compute the distance between all documents, so we apply hierarchical classification on a subset of documents then we apply the cluster-based algorithm as done on French language by Bellot and El-Beze.

However the initial partition has large computational time to build but this is not a serious problem because we need to build this initial partition only once to classify our corpora, after that we compute the class centroids which really represent every class. For the search task we only compute the distance between the query and these class centroids to determine which class to return and this is a fast process, and even if we want to increase the size of our corpora; we just need to compute the distance between every class centroids and the new documents to determine where to locate these new documents and not with the whole class.

In a second step, the application of the "Nuées dynamiques", a K-Means-like method, allows to classify the documents ignored during initialization [6].

D. The Algorithm

The main classification step is performed as follows:

- Find an initial partition
- *Do:*
- Compute centroids of each cluster
- Allocate each document to the nearest cluster (that has the lowest distance).
  while there is little or no change in cluster membership.

We note that in this clustering criterion if a document is assigned to one cluster it is not allowed to assign it to another cluster because we are using hard clustering technique. To assign a document to a cluster we have to compute the distance between this document and the cluster were the distance is calculated with respect to the cluster centroids, if this distance does not exceed a specific threshold it is assigned otherwise not [6].

The number of clusters is specified before the procedure starts, in our system we clustered our corpora into two, three, and five clusters each cluster has three centroids, then the documents retrieved are ranked in descending order using the similarity measure (The cosine similarity measure is used) as they were before classification. At the end of this process there will be some documents that are not assigned to any cluster; these documents are not our interest because most of them are usually irrelevant to the user query.

E. Calculating The Distance Between Documents

The first step of the HKM algorithm is to find the initial partition, however to find this initial partition we have to compute the distance between every pair of documents. In this subsection we will explain how to calculate this distance.

Let R and D be two documents, u a lemma and its syntactical tag, N(u) the number of documents containing u in the corpus as a whole.

Given S, the number of documents in the corpus, the information quantity of a term in a document is based on its occurrences in the corpus —IDF(u)— (and not in the set of documents to cluster) and on its frequency in the document —TF(u)—.

We need to compute the information quantity first because we will need it in calculating the distance.

The information quantity of a document is the sum of the weights of its terms [6]:

$$I(D) = \sum TF(u).IDF(u) = \sum - TF(u).Log_2\left[\frac{N(u)}{S}+1\right] \qquad (1)$$

We assume that the greater the information quantity of the intersection of the lemma sets from two documents, the closer they are [6].

In order to allow convergence of classification process, we must have a true distance (verifying the triangular inequality). That is the case of the so-called MinMax distance between two documents D and D' [6]:

$$D(D, D') = 1 - \frac{I(D \cap D')}{Max(I(D), I(D'))} \qquad (2)$$

F. Cluster Centroids

The cluster X is represented by k documents that are the nearest to X geometric center. For each document, we compute the sum of distances in same cluster and choose the k documents corresponding to the k smallest distances as centroids or ‘representatives’ This avoids computing a "mean vector" and allows using the same similarity values during K-Means iterations (similarities between documents are computed only once) [6].

However determining the constant number k (number of centroids in each cluster) is not a simple choice, because we have to take large number enough of documents that really represent the whole cluster and small enough to reduce computational cost.

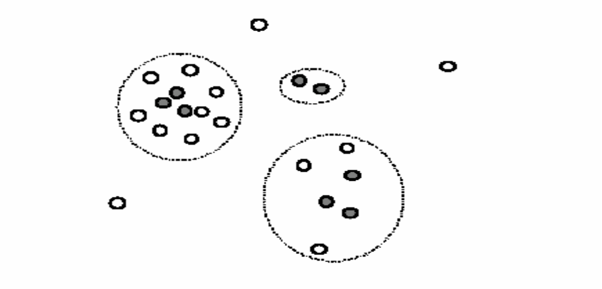

Fig. 3 Cluster centroids (each cluster has 3 centroids)

Let d be the distance between a document and a cluster:

$$D(D,C) = \min(d(D, N_1\ )) \qquad (3)$$
$$1 <= I < k$$

Since centroids in fact are documents, we can use the similarity measure between the document and the query to rank the clusters according to the query [6].

G. Initial Partition

The initial partition affects the results obtained by the cluster-based method, because the initial partition is the first input to it, so we have to choose it carefully.

To obtain the initial partition we do the following two procedures [6]:

- *single-link :*
  For each couple of documents i and j such that d(i,j) < threshold :
- If i and j are not yet in a class, create a new one;
- If i and/or j are already allocated, merge all the documents of the class containing I (resp. j) with those of the class containing j (resp. i);
- *partial hierarchical classification:*

After this step, the number of classes may be greater than the number of clusters wanted.

So, as long as the number of classes is greater than the predefined one, we can:

- Compute class representatives;
- Compute distances between every pair of classes (triangular matrix);
- Merge the two closest classes.

III. EXPERIMENTAL RESULTS

There are several criteria's to evaluate the quality of clustering, as mentioned by Bellot & El- Beze we can consider the best ranked cluster which should contain the most relevant documents, or we can lock at the best ranked documents of each cluster [6]. We choose to take the best ranked cluster.

In our system we applied 59 queries on 242 Arabic abstract documents, which are clustered into several sets of clusters (2, 3 and 5), then we compared the results with the traditional IR system.

As mentioned, when we change the number of corpora clusters we have to compute each cluster centroids, the following table shows cluster centroids for each cluster.

TABLE I
CLUSTER CENTROIDS FOR EACH CLUSTER

Number of Clusters	2		3			5				
Cluster Centroids	C1	C2	C1	C2	C3	C1	C2	C3	C4	C5
	D151	D40	D18	D151	D109	D15	D88	D122	D125	D131
	D88	D55	D15	D88	D110	D40	D151	D137	D130	D132
	D156	D15	D40	D201	D121	D6	D55		D123	

We found that the effect of Hierarchical K-Mean Like clustering (HKM) with two and three clusters over 242 Arabic abstract documents from the Saudi Arabian National Computer Conference has significant results compared with traditional information retrieval system without clustering. Also we found that it is not necessary when increasing number of clusters that will improve precision more, because when we clustered our corpora into five clusters we noted that the results are worse than the traditional information retrieval system. This fact means that our corpora are talking about two or three main topics (Table II, Fig. 4). Table II shows a comparison between average precision of traditional IR system versus HKM system with 2, 3, and 5 clusters (Fig. 4).

TABLE II
COMPARISON OF AVG. PRECISION BETWEEN TIR SYSTEM WITH HKM (2, 3, 5 CLUSTERS)

Average Precision				
59 Query				
Recall	Precision			
		2 Clusters	3 Clusters	5 Clusters
	Traditional IR	HKM	HKM	HKM
0	0.495125571	0.623949	4.96E-01	0.598762
0.1	0.430327917	0.545192	4.68E-01	0.50622
0.2	0.409629791	0.506637	4.35E-01	0.391798
0.3	0.422326785	0.482722	4.54E-01	0.347748
0.4	0.403071831	0.448129	4.54E-01	0.315876
0.5	0.401611859	0.423424	4.54E-01	0.28682
0.6	0.364446288	0.386385	4.48E-01	0.276785
0.7	0.352122464	0.352531	4.48E-01	0.273935
0.8	0.347943723	0.326402	4.48E-01	0.273785
0.9	0.345470868	0.306917	4.48E-01	0.273785
1	0.343353455	0.292732	4.48E-01	0.274365

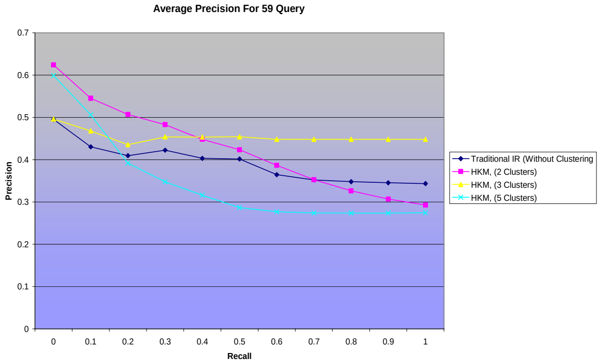

Fig. 4 Comparison of Avg. precision Between TIR system with HKM (2, 3, and 5 clusters)

TABLE III
COMPARISON OF AVG. PRECISION BETWEEN TIR SYSTEM WITH HKM (2, 3, 5 CLUSTERS)

Recall Level	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Without Clustering	0.45	0.43	0.41	0.42	0.40	0.40	0.36	0.35	0.35	0.35	0.34
HKM (3 Clusters)	0.50	0.49	0.46	0.45	0.45	0.45	0.49	0.45	0.45	0.45	0.45
Improvement percentage	1%	6%	5%	3%	5%	5%	13%	12%	10%	10%	11%

IV. CONCLUSIONS

In this study the concept of clustering documents has shown significant results on precision compared with traditional retrieval systems without clustering. However these results assure the results obtained by Bellot & El-Beze during there test on Amaryllis'99 corpora.

It also has been found that it is not necessary to increase number of clusters to gain better results. In this corpora a series of tests have been made at several number of clusters (2, 3, and 5), and was found that the best results is at 3 clusters which means that this corpora talks mainly about three topics.

Table III shows the percentage of improvement on precision between traditional information retrieval system and HKM system with three clusters.

ACKNOWLEDGMENTS

The author gratefully acknowledges and highly appreciates the financial support and the remarkable resources provided by Yarmouk University, Irbid, Jordan.

REFERENCES

[1]. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification”, in Proc. of the AAAI-98/ICML-98,Workshop on Learning for Text Categorization (AAAI), Madison; 1998, pp. 71-74.

[2]. D. Fragoudis, D. Meretakis and S. Likothanassis, *Integrating Feature and Instance Selection for Text Classification*, 2000, pp. 27-37.

[3]. K. Nigam, A. Kachites, S. Thrun and T. Mitchell, *Text Classification from Labeled and Unlabeled Documents using EM*. Kluwer Academic Publishers, Boston. 1999.

[4]. K. Thompson and R. Nickolov, “A Clustering-Based Algorithm for Automatic Document Separation”, in *Proc. of the SIGIR 2002, Workshop on Information Retrieval* , 2002, pp. 38-43.

[5]. N. Slonim and N. Tishby, “The Power of Word Clusters for Text Classification”, *in Proc. of the 23rd European Colloquium on Information Retrieval Research*, 2001,pp. 1-12

[6]. P. Bellot and M. El-Bèze, “Clustering by means of Unsupervised Decision Trees or Hierarchical and K-means-like Algorithm”, in Proc. of RIAO 2000, pp. 344-363.

[7]. P. Dai, U. Iurgel and G. Rigoll, “A Novel Feature Combination Approach for Spoken Document Classification with Support Vector Machines”, in *Proc Multimedia Information Retrieval Workshop in conjunction*, 2003, pp. 1-5.

[8]. R. Ghani, “Using error-correcting codes for text classification”, in *Proc. 17th International Conference on Machine Learning (ICML-00)*, Stanford, CA, 2000, pp. 303–310.

[9]. R. Ramakrishnan and J. Gehrke, *Database Management Systems*. McGraw-Hill, 2002.

[10]. T. Theeramunkong and V. Lertnattee, “Multi-Dimensional Text Classification”, in *Proc. of the 19th International Conference on Computational Linguistics*, Taipei, 2002, pp. 34-38.

[11]. Y. Fang, S. Parthasarathy, and F. Schwartz, “Using Clustering to Boost Text Classification”, in *Proc. of the IEEE International Conference on Data Mining*, California, USA, 2001, pp. 123-127.