Clustering of Text Document using Kea-means with F-measure

Ms. M.J.Yeola
Computer Department,
MAEER's MAE, Alandi (D),
Pune, India.

Abstract– **Document clustering is an area that deals with the unsupervised grouping of text documents into meaningful groups, usually representing topics in the document collection. It is one way to organize information without requiring prior knowledge about the classification of documents. The well-known K-means clustering algorithm allows users to specify the number of clusters. However, if the pre-specified number of clusters is modified, the precision of each result also changes. To solve this problem, this paper proposes a new clustering algorithm based on the Kea keyphrase extraction algorithm. In this paper, documents are grouped into several clusters like K-means, but the number of clusters is automatically determined by finding out the similarities between documents and the extracted keyphrases. It also calculates F-measure value using precision and recall which gives the better clusters.**

Keywords– Keyphrases, cluster, F-measure, precision, recall.

INTRODUCTION

The simplest form of clustering is partitional clustering which aims at partitioning a given data set into disjoint subsets (clusters) so that specific clustering criteria are optimized. The most widely used criterion is the clustering error criterion which for each point computes its squared distance from the corresponding cluster center and then takes the sum of these distances for all points in the data set.

A popular clustering method that minimizes the clustering error is the k-means algorithm. However, the k-means algorithm is a local search procedure and it is well known that it suffers from the serious drawback that its performance heavily depends on the initial starting conditions [1].

To solve this problem, this paper proposes a new clustering algorithm based on document's keyphrases which improves the traditional K-means algorithm. The Kea means clustering algorithm applies the Kea keyphrase extraction algorithm which returns several keyphrases from the source documents by using some machine learning techniques [3, 4].

In this work, documents are grouped into several clusters like K-means, but the number of clusters is automatically determined by the algorithm with some heuristics using the extracted keyphrases. After this phase, the similarity between the keyphrases of each cluster chosen by the Kea keyphrase extraction algorithm and the test document is computed. When the similarity exceeds a specific threshold, the cluster containing the keyphrases is likely to have the document which users are looking for. The Kea-means clustering algorithm provides better clusters by calculating the F-measure value using precision and recall. It provides easy and efficient way to extract test documents from massive quantities of resources.

I. RELATED WORK

A. Kea- Automatic Keyphrase Extraction

Keyphrases are useful for a variety of purposes, including summarizing, indexing, labeling, categorizing, clustering, highlighting, browsing,

Proc. of the Intl. Conf. on Advances in Computer Science and Electronics Engineering
Editor In Chief Sahil Seth.
ISBN: 978-981-07-1403-1
doi:10.3850/978-981-07-1403-1_185

and searching. The task of automatic keyphrase extraction is to select keyphrases from within the text of a given document. Automatic keyphrase extraction makes it feasible to generate keyphrases for the huge number of documents that do not have manually assigned keyphrases.

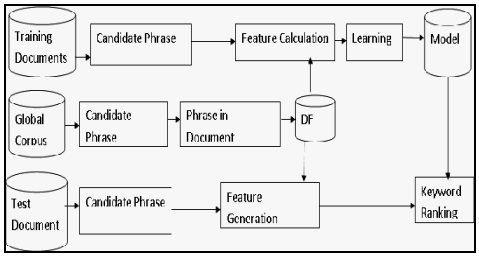

Fig 1 The training and extraction processes

Kea is a Java-based keyphrase extraction algorithm that generates candidate phrases from a document and selects keyphrases from them by using TF-IDF and naive Bayes classifier [8]. Kea's extraction algorithm has two stages, training and extraction. The training stage uses a set of training documents for which known keyphrases are provided. For each training document, candidate phrases are identified and their feature values are calculated. Each phrase is then marked as a keyphrase or a non-keyphrase, using the actual keyphrases for that document. In the extraction phase, the algorithm chooses keyphrases from a new document using the above model. To select keyphrases from a new document, Kea determines candidate phrases and TF-IDF value for every keyphrase, and then applies the model built during training. The model determines the overall probability that each candidate is a keyphrase, and then a post-processing operation selects the best set of keyphrases. This process of Kea is shown in Fig. 1[3]. Both stages choose a set of candidate phrases from their input documents, and calculate the values of certain attributes for each candidate.

B. K-Means Clustering

The K-means clustering algorithm is one of the simplest clustering algorithms in which the number of clusters to be grouped is fixed by the user. The algorithm proceeds by randomly defining *k* centroids and assigning a document to the cluster that has the nearest centroid to the document. In general, obtaining the nearest centroid for a given document and re-calculating new centroids use the cosine measure or the Euclidean distance measure. In K-Means clustering algorithm values of *k is given and k-means* algorithm is implemented in 4 steps:

1. Partition objects into *k* nonempty subsets.
2. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
3. Assign each object to the cluster with the nearest seed point. Go back to Step 2, stop when no more new assignment.

II. KEA-MEANS WITH F-MEASURE

The Improved Kea-means clustering algorithm is our proposed new clustering method that improves the K-means algorithm by combining it with the Kea keyphrase extraction algorithm. The Improved Kea-means clustering attempts to solve the main drawback of K-means i.e. the number of total clusters is to be pre-specified in advance. In Improved Kea-means algorithm, documents are clustered into several groups like K-means, but the number of clusters is determined automatically by the algorithm using the extracted keyphrases [9]. The system architecture of the Improved Kea-means clustering is shown in Fig. 2.

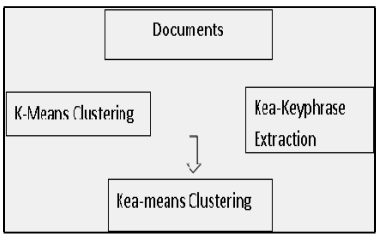

Fig 2 Kea-means clustering system architecture

The main idea of Improved Kea-means is the following: Initially, the number of clusters *k* is set to 2. As in K-means, *k* clusters are formed by generating *k* centroids, and the similarities between centroids and documents are measured, and a document is assigned to the cluster that has the nearest centroid. Then, the Kea algorithm is applied to each document that is nearest to the corresponding centroid to extract keyphrases. These keyphrases are used to assign weights to other phrases. Now, the distance between the weighted documents and centroids are measured, and if the measured values do not reach to the threshold value, the value of *k* is increased by 1. This process is repeated until the measured distance exceeds the threshold value. This process is repeated for no of runs specified.

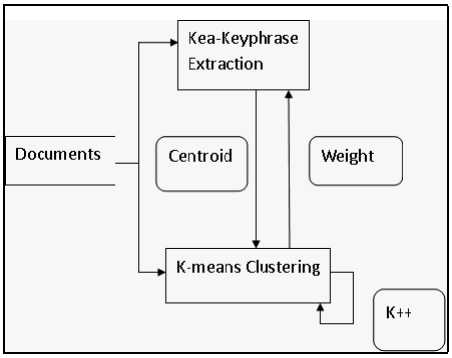

Fig 3 The process of kea-means clustering

At this point, highest F-measure value determines the number of clusters *k* which gives the better clusters and the K-means algorithm is now used for actual clustering. This process of the Improved Kea-means clustering is shown in Fig. 3.

The main characteristics of the Improved Kea-means clustering algorithm can be summarized in two ways. First, compared with previous clustering algorithms in which the task of clustering is done without knowing the semantic nature of each cluster, the Kea-means clustering recognizes this semantic nature of clusters by using the keyphrases extracted from the documents in the cluster. Second, the traditional K-means must specify the number of clusters *k* in advance by the user, which results in the change of clustering results as the value of *k* changes. Improved Kea-means solves this problem by automatically determining this number. Improved Kea-means clustering algorithm uses Cosine measure and the Euclidean distance measure to calculate feature values, simultaneously. Hence, the similarity of two documents is computed by the following expression:

$$Sim(d1, d2) = \frac{cosine(d1,d2)}{euclidean(d1,d2)} \quad \text{.....Eq. 1}$$

Also it calculates F-measure using precision and recall which gives better clusters. Precision and recall is calculated using following expression:

$$Recall(i, j) = C_{ij}/C_i$$

$$Precision(i, j) = C_{ij}/C_j \quad \text{.....Eq. 2}$$

And F-measure is calculated using following expression:

$$F(i, j) = \frac{2*Recall(i,j)*Precision(i,j)}{Precision(i,j)+Recall(i,j)} \quad \text{..Eq.3}$$

Where C_{ij} is the number of members of topic i in cluster j, C*j is* the number of members of cluster j and C*i is* the number of members of topic i.

III. EXPERIMENTS

Experiments are performed to compare the Improved Kea-means clustering algorithm with the K-means clustering algorithm. Documents are collected from standard datasets viz. Reuters and Newsgroup20, and the clustering algorithms are executed for this collection 20 times. In each test, *k* centroids are randomly selected from the document space, and the classification accuracy is measured by assigning documents to the correct cluster. The results of Reuter dataset (Fig. 4) and Newsgroup20 (Fig.5) are as shown below.

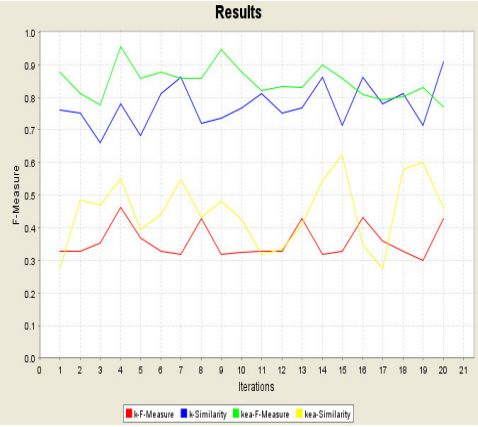

Fig 4 Result of Reuters dataset

Overall, the Improved Kea-means algorithm shows better performance in terms of the F-measure Value.

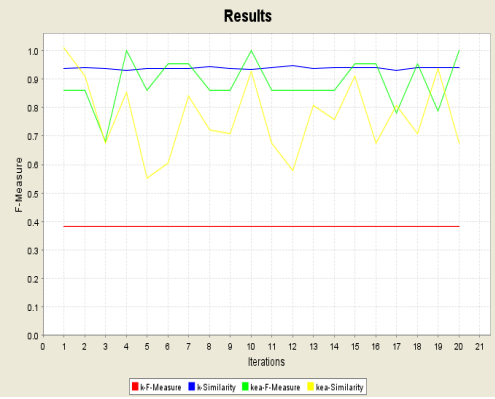

Fig 5 Result of newsgroup20

IV. CONCLUSION

Kea-means clustering algorithm calculates F-measure value with accuracy and similarity value.Here cosine and Euclidean distance measures are used simultaneously to calculate similarity. Cosine value should be high for better clustering while Euclidean distance value should be low for better results. Precision and Recall values are taken into consideration to calculate F-measure value which gives advantage for finding out the better clusters Keyphrase extraction is used for text summarization and similarity analysis. This algorithm outperforms k-means algorithm in terms of F-measure value.

This algorithm is only useful for text documents which are based on the topics. In future it can be enhanced for other type of documents. Also it is not efficient as it requires predefined keyphrases and its process of building clusters by repeatedly incrementing the number of clusters k.

REFERENCES

[1] Likas, N. Vlassis, J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451-461, February 2003.

[2] C. Clifton, R. Cooley, J. Rennie. TopCat: Data mining for topic identification in a text corpus. *IEEE Trans. Knowledge and Data Engineering*, 16(8):949-964, August 2004.

[3] I. Witten, G. Paynter, E. Frank, C. Gutwin, C. Nevill-Manning. Kea: Practical automatic keyphrase extraction. *Proc. 4th ACM Conference on Digital Libraries*,254-255, August 1999.

[4] J. Wu, A. Agogino. Automating keyphrase extraction with multi-objective genetic algorithms. *Proc. 37th Annual Hawaii International Conference on System Sciences (HICSS)*, 104-111, 2004.

[5] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques. *Proc. KDD Workshop on Text Mining*, 1-20, 2000.

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

[7] P. Turney. Learning to extract keyphrases from text. *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 434-439, 2003.

[8] E. Frank, G. Paynter, I. Witten, C. Gutwin, C. Nevill-Manning. Domain-specific keyphrase extraction. *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI)*, 668-673, 1999.

[9] Juhyun Han, Taehwan Kim, Joongmin Choi, Web Document Clustering By Using Automatic Keyphrase Extraction *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Workshops* 2007.

[10] J.A. Lozano, J.M. Pena, P. Larranaga, An empirical comparison of four initialization methods for the k-means algorithm, Pattern Recognition Lett. 20 (1999) 1027–1040.