www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 5 May 2015, Page No. 11956-11961

Search Engine Results Clustering Using TF-IDF Based Apriori Approach

Hetal C. Chaudhari[1], K. P.Wagh[2] and P.N.Chatur[3]

[1]PG Scholar, Government College of Engineering,
Department of Computer Science and Engineering, Amravati. INDIA
hetalchaudhary90@gmail.com

[2]Assistant Professor, Government College of Engineering,
Department of Information Technology, Amravati. INDIA
kishorwagh2000@yahoo.com

[3]Associate professor, Government College of Engineering,
Department of Computer Science and Engineering, Amravati. INDIA
chatur.prashant@gcoea.ac.in

Abstract— ***Internet is being used at a greater extent nowadays. All the types of data are available very easily on the internet. The user submits a query to the search engine and thousands of related documents are retuned as a result to the query. The web documents contain different types of data like text, images, videos, etc. So, the web documents are not structured properly and are unorganized. It becomes much difficult for users to find specific document from thousands of documents. The solution to this problem is clustering the web documents. Clustering congregates the documents showing similar context to the user query. The similar documents are assembled in a cluster. So, clustering reduces user's task to discriminate among the thousands documents returned as a result to a query. Also, ranking can be applied further to view the most relevant documents at the top. Different documents in a cluster are ranked and the documents can be arranged according to their similarity. Different functions can be used to calculate the similarity measure among the documents. We combine these two concepts and propose a tf-idf based apriori scheme for web document clustering and ranking. In this scheme, first clustering is applied on the documents. The modified tf-idf based apriori algorithm is used to serve this purpose. And then, ranking is performed to arrange the most pertinent documents at the top with regard to the user query. We use online web pages returned as results for a query as the dataset for our experimental work. This approach gives a good F-measure value, i.e. 81%. The proposed method is found superior to some traditional clustering approaches.***

Keywords— apriori algorithm, web documents, search results, term frequency, inverse document frequency

1. INTRODUCTION

In response to the user query, search engines return a list of results. The results are arranged in order of their relevance to the user query. The results obtained for a specific word can be of many different contexts according to different meanings of that word. In case of long or ambiguous queries, the resulting list may extend many vast areas on different subtopics depending on different meanings of the query. Also, there is no concrete way to know which results are relevant to the query and which are not. In this case, the results can be subdivided further and those which show similar context can be grouped together to form the clusters. Documents belonging to a group are much different than those belonging to other groups. Clustering makes a shortcut to the items that are similar in the meaning. Also, it favours systematic representation of the results. Neural networks, Association techniques, rule-based techniques, decision trees and many other techniques are used for clustering of the search results. Clustering can be characterized as the process of dividing the results in such a way that the elements belonging to a cluster are similar and those belonging to different clusters are not similar.

Clustering is performed on the search results obtained after processing a query. Here, query specific features are used for the clustering purpose. Whereas, if groups obtained after clustering are decided before getting the search results, the features which are frequent but are not relevant to the query in hand may also be taken into consideration.

Clustering subdivides the input into different regions. The input space S is divided into N no. of regions. The groups into which the input space has to be divided may not be known in advance. So, the process of clustering can also be referred as unsupervised leaning process. Figure 1 shows the architecture of the proposed system.

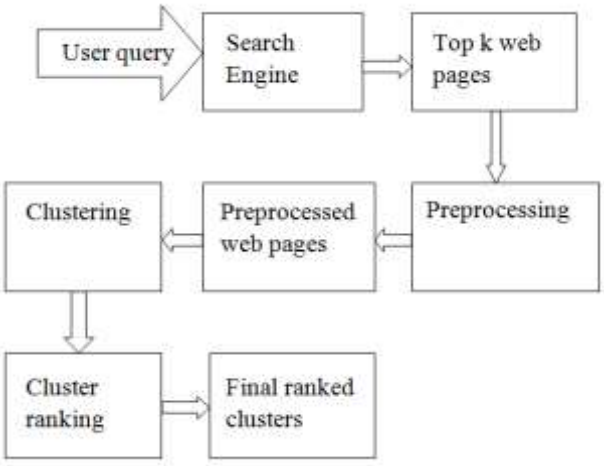

Figure 1. Architecture of the proposed system

In this approach, a technique, i.e. Tf-Idf based Apriori has been proposed. Frequent itemsets are made from the set of documents using this approach. The technique calculates threshold which is used to eliminate the terms which are having their tf-idf values greater than the threshold. The threshold is being calculated as

$$\text{threshold} = \frac{1}{\text{minimum support}} \times \log_{10} \frac{\text{total no of documents}}{\text{minimum support}} \quad (1)$$

This threshold is used to discard rows and columns of tf*idf table. After the elimination, sets of different frequent itemsets are formed. Usually, the apriori algorithm is used for forming frequent itemsets from the candidate itemsets. The process needs to create candidate sets for that. The concept of generating frequent candidate itemsets is similar to the concept of generating frequent itemset and candidate itemsets of traditional apriori algorithm.

In the ranking process of the documents of a cluster, the cosine similarity between every pair of documents in each cluster is calculated. Using these values, the similarity factor of each document is calculated. It shows how far the two documents in a cluster are similar to each other. And finally, ranking is performed based on the user query. In this way, the user can get all the documents in a properly ranked and organised manner.

2. RELATED WORK

The process of clustering congregates the documents that are similar to each other in a single group. Different Similarity measures are used for computing similarity among the documents. There are various techniques available for performing clustering of the documents. But, k-means algorithm is considered as the most popular algorithm of document clustering. But, the number of clusters to be formed has to be decided first in this case and when the choice of no. of clusters is inappropriate, the results are also affected badly.

CBC - A clustering algorithm, invented by Patrick Pantel [1], identifies word senses from the web documents. First, a set of clusters that well occupies the input space is discovered and then centroids of these clusters are calculated. The centroids calculated for each cluster are treated as feature vectors for that cluster.

K-means clustering algorithm with a newly proposed distance metric, i.e. design specification distance measure function was put forward by Doreswamy and Hemanth K.S. [3] to improve the clustering accuracy. Even after 50 years of the invention, K-means clustering algorithm is the most popular approach for clustering web pages. In that case where data distribution is not known, K-means algorithm is the easiest algorithm for document clustering. This algorithm starts with some specific number of clusters initially. Some k numbers of clusters are presumed in each case. Then, the centres for each of these clusters are fixed. In the next step, each data point is assigned to a specific cluster. The position of the centroid of the cluster is fixed to the mean of all the data points that belong to the cluster. These steps are repeated till it converges. In this way, the k-means algorithm works. The k-means algorithm operates such that it minimizes the squared error between the centroid of a cluster and other data points that belong to that cluster. Distance is said to be the quantitative degree that calculates the logical separation of two data points in a cluster on the basis of measurable characteristics. Distance is a measure that reflects how similar or different the two documents of the cluster are. A function D that is defined on the Cartesian product, i.e. $M \times M$ of a set R with non-negative real values is called as distance. The point in a single dataset is expressed by the term m_i and n_i. The distance measure function can be defined on two documents A and B as,

$$D(A,B) = \left[\sum_{i=1}^{n} |a_i - b_i|^2 \right]^{\frac{p}{3}} \quad (2)$$

Where, n is the number of documents in the cluster. Suppose, the documents are $A = (a_1, a_2 \ldots a_n)$ *and* $B = (b_1, b_2 \ldots b_n)$ and p is a user defined parameter.

B.Shanmugapriya et al. [9] recommended a new modified projected K-means clustering approach in combination with effective distance measure. The traditional K-means algorithm is generalized in new approach and the motive of the algorithm is to work on the high-dimensional data. The newly proposed algorithm enhances an objective function comprehensively. The newly proposed effective distance measure is used by the objective function of the new algorithm to get more appropriate results in clustering of the high dimensional data. In the case where all the dimensions of high-dimensional data are not considered, the value of the objective function is decremented. To resolve this, the virtual dimensions are also considered along with the objective function. The two requisites for the data values on the virtual dimensions assure that the objective function gains the lowest value whenever the clusters are found.

3. BACKGROUND

Various concepts are used in the proposed system. The concepts such as Vector space model, TF-IDF and cosine similarity model are discussed here.

3.1 Vector space model

Vector space model (VSM) is a most commonly used and popular algebraic model. Text documents can be represented as vectors of identifiers using this model. In our case, the documents can be represented as multidimensional vectors of keywords. The keywords can be extracted from the document in Euclidean space. Each keyword is assigned a weight. The weight assigned with the keyword shows how relevant the keyword is in the document. Therefore, a document can be represented in the vector form as, $D_j = [w_{1j}, w_{2j}, w_{3j}, w_{4j}..., w_{nj}]$. Here, W_{ij} represents the weight of the keyword i in document j.

3.2 TF-IDF

TF-IDF is generally numeric statistic for the documents that shows how relevant a word is in the document in a collection or corpus. The number of times a term occurs in a document is called as term frequency.

Inverse Document Frequency (IDF) measures how rare a term is in the whole collection. Sometimes, some terms are very common in the document or they don't have much significance in the document. Inverse document frequency factor deals with diminishing the weight of such terms that appear very frequently in the document. The terms that appear rarely are given more importance. The inverse document frequency regarding a term t is defined as follows:

Where, N- total no of documents and df- the no of documents the term appears in.

The concepts of term frequency and inverse document frequency are united in our approach. Each term is assigned a weight, i.e. tf-idf. It can be formulated as:

$$\text{tf-idf} = \text{tf} * \text{idf}. \quad (3)$$

3.3 Cosine-Similarity Measure

Cosine similarity measures the extent of similarity between the vectors. In our case, the documents are represented as the vectors of the keywords. Every keyword is assigned a particular weight based on its tf-idf value. It is powerful and one of the most popular technique for measuring similarity as compared to other techniques. Cosine similarity is the judgement of orientation and not the magnitude. If the angle between the two vectors is 0^0, then the two vectors are very similar. As the angle varies from 0^0 to 90^0, the similarity decreases. The cosine similarity of two vectors can be formulated as:

$$\text{cosine-similarity}(q,d) = \frac{q.d}{||q||.||d||} \quad (4)$$

Where, q represents the query vector and d represents the document vector. Also ||q|| - length of the query vector and ||d|| - length of the document vector.

4. PROPOSED SYSTEM

There are two main phases of the proposed system. First is cluster formation and the other is ranking the clusters.

4.1 Cluster formation

Input to the system is Web pages and the use query. The expected output is well-formed clusters.

The steps involved in formation of the clusters are as follows:

1. Extraction of web pages:
   The user query has to be submitted to a search engine. The first n pages of the search engine results are to be extracted.

2. Preprocessing the documents:
   The extracted web pages have to be preprocessed. The documents pre-processing includes following steps:
   a. Remove the stop-words and other unwanted words.
   b. Do stemming using porter algorithm.
   c. Save each processed page as a document D_i, where i = 1, 2, 3, …, N

3. Now, consider each keyword as a transaction and the document in which the keyword appears must be considered as an element of the transaction.

4. Calculate tf for each distinct keyword as:

$$\text{tf} = \frac{\text{No of times the keyword appears in the document}}{\text{Total no of keywords in the document}} \quad (5)$$

   Also, calculate the idf for each distinct keyword as:

$$\text{idf} = \log_{10} \frac{\text{total no of documents}}{\text{no of documents the keyword appears in}} \quad (6)$$

5. Calculate tf and idf for each keyword. Also calculate tf-idf value for each keyword and add the entries into the tf-idf table.

6. Now, calculate threshold as given in equation no 1.

7. Generate frequent candidate itemsets (S) as:
   for all keywords till,
   $0 < \min \{tf*idf_{D1}, tf*idf_{D2}, \ldots, tf*idf_{DN}\} <= \text{threshold}$
   for all frequent candidate itemsets, i.e. S and for each S, n>=2

   Now mark those frequent candidate itemsets (rows) for elimination for which min $\{ tf*idf_{D1}, tf*idf_{D2}, \ldots, tf*idf_{Dn}\}$ > threshold.

   Mark documents (columns) for elimination if min { tf*idf n frequent candidate item set_1 , tf*idf n frequent candidate item set_2 , ... , tf*idf n frequent candidate item set_N} >threshold.

8. Form the final clusters (C_i) where i = 1, 2, 3,. . . ,M. Each cluster has a group of similar documents. And the number of documents belonging to a cluster may vary cluster to cluster.

4.2 Ranking of cluster documents:

Input to this step is the output of the step one, i.e. clusters formed from the corpus in step one and another input is the user query words. The use query can also be represented as the query vector, W_e.

Output of the step is the ranked clusters.

The steps involved in this step are as follows:

1. Calculate the cosine similarity for every pair of documents in the clusters formed in step 1.

2. Preprocess the user query string.
   a. Remove the stopwords and unwanted words.
   b. Apply porter stemming algorithm.

3. Calculate the similarity factor for each document in the cluster. The similarity factor can be formulated as:

$$\sum_{m=1,m\neq k}^{N}\left(\frac{\text{No of keywords common to } D_k \text{ and } D_m}{\text{Total unique words in } D_k \text{ and } D_m}\right) \times \text{cosinesim}(D_k, D_m) \quad (7)$$

4. Each document in the cluster is ranked at this step. The rank can be calculated as:

$$\text{Rank}(D_k) = \left(\sum_{e=1}^{p} (\text{Tf*Idf})W_e, D_k\right) * \text{Simfact}(D_k) \quad (8)$$

   Here, $(tf*idf)_{We,}$ D_k shows the tf*idf of the query words regarding the document D_k.

5. Sort the documents in every cluster according to their ranked values.

5. EXPERIMENTAL RESULTS

The dataset we are using here is the online web pages. We submit the user query to the search engine and extract top N pages. We use this data as raw data. Then, we perform the basic steps of pre-processing on the documents. All the stopwords and unwanted words from the documents are removed. Then, the porter stemming algorithm is applied. Thus, we get the data on which we perform the steps in proposed approach.

On the results that we get after the application of the proposed algorithm, we tested the F-measure factor. The F-Measure calculates the average of the precision and recall and thus measures the system performance. It computes the harmonic mean of Precision and Recall. F-measure weighs Precision and Recall evenly and thus reflects overall algorithm performance under consideration.

While calculating F-measure, every cluster is considered to be the result of a query and each class to be the desired resulting set of documents for a query. Then, the recall and precision of the cluster for each given class is calculated. The F-measure for cluster j and class i is computed as follows:

$$F_{ij} = \frac{2*\text{Precision}(i,j)*\text{Recall}(i,j)}{\text{Precision}(i,j)+\text{Recall}(i,j)} \quad (9)$$

Here, we calculate the F-measure of some of the clusters formed after submitting a user query. The dataset 1, dataset 2, dataset 3 and dataset 4 are the set of snippets that we get after executing the queries for the words "Microsoft", "pepsi", "ipod", "Maharashtra". Figure 2 shows the F-measure values of different datasets..

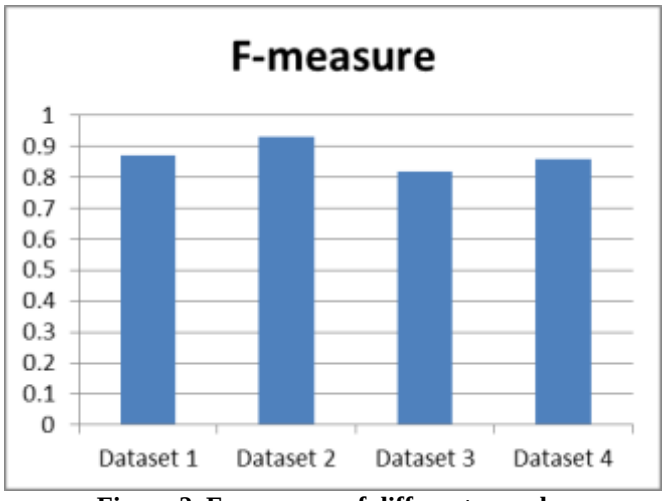

Figure 2. F-measures of different samples

Precision and recall values for clusters formed out of dataset 1(snippets for query word 'Microsoft") are calculated. The precision and recall values for clusters for the dataset 1 is shown in the Figure 3.

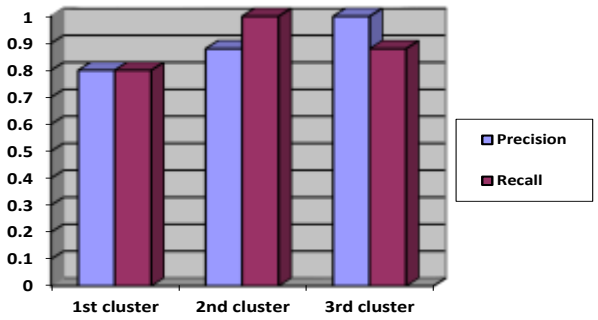

Figure 3. Precision and recall values of different clusters

We now examine the introduced system by changing the basic parameter value, i.e. minimum support. We are calculating the threshold value on the basis of minimum support as mentioned in equation no 7. Also, we are using the value of threshold to eliminate the less frequent documents on the basis of comparing the tf-idf values of the keywords of the document as per the steps of the modified algorithm.

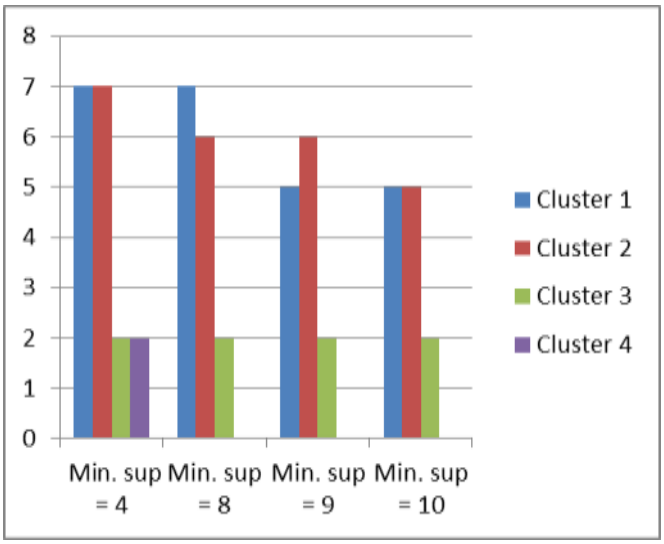

Figure 4. No. of clusters for different values of minimum support

The changes in number of clusters and the number of documents in a cluster after varying the values of minimum support are shown in the Figure 4.

6. CONCLUSION

This paper proposes an approach called Tf-Idf based Apriori. An equation for computing the threshold is formulated. When it is combined with the proposed Tf-Idf, it identifies frequent itemsets from the set of documents. This threshold is used to discard rows and columns of tf-idf table created. The concept of frequent candidate itemset generation in the proposed approach is similar to the concept of generation of frequent itemset generation and candidate itemsets in traditional apriori algorithm. In experimental work, we take the user query as the input and submit it to the search engine. Then, top n documents are extracted. Then, we have applied the modified algorithm on the dataset and also, we use the concept of cosine similarity. Thus, after the completion of this step, we get the clusters.

While ranking the documents, the cosine similarity measure is applied between every pair of documents in a cluster. After this, the similarity factor is calculated for each document. Using the value of similarity factor and the tf-idf value of query words in each document, the rank values are calculated for each document in a cluster. Then these values are sorted and the documents are arranged properly in a cluster according to their rank values. Thus, we get properly ranked clusters. It is observed that on an average 81% of documents have been ranked in an appropriate order in each cluster. Ranking of documents will make it easy for the user the user to get the required document at the very start of each cluster and thus, will reduce his search process. This approach can be extended by taking into account the documents that are not included in initial clusters due to strong association rule.

REFERENCES

[1] Patrick Pantel and D. Lin. Discovering Word Senses from Text, SIGKDD'02, July 23-26, 2002, Edmonton, Ablerta, Canada .

[2] J.Chen, O.R.Zaiane and R.Goebel. An Unsupervised Approach to Cluster Web Search Results based on Word Sense Communities, 2008 IEEE/WIC/ACM, International Conference on Web Intelligence and Intelligent Agent Technology.

[3] Doreswamy and Hemanth K.S. A Novel Design Specification (DSD) based K-mean clustering performance Evaluation on Engineering material's database, IJCA, Vol 55, No.15, Oct-2012.

[4] Mansaf Alam and Kishwar Sadf.Web search result clustering using heuristic search and latent semantic indexing, IJCA, Vol 44, No. 15, April 2012.

[5] Li p, Wang B and Jin W. Improving web document clustering through employing user-related tag expansion techniques, Journal of Computer Science and Technology 27(3):554-556 May-2012. DOI 10.1007/ S11390-012-1243-4.

[6] Ingyu Lee and B-Won. An Effective web document clustering algorithms based on bisection and merge, Artif Intell Rev (2011) 36-69-85, DOI 10.1007/S10462-011-9203-4.

[7] R.Thiyagarajan, K.Thangavel and R.Rathipriya. IJCA, Vol-86, No. 14, Jan-2014.

[8] Khaled M and Mohamed S. Efficient phrased-based document indexing for web document clustering, IEEE Transaction on Knowledge and Data Engineering, Vol 16, No. 10, October-2004.

[9] B. Shanmugapriya and M.Punithavalli. A modified projected K-means clustering algorithm with effective distance measure, International Journal of Computer Application, Vol 44, No. 8, April-2012.

[10] Naresh Barsagade. Web usage mining and pattern discovery: A survey paper, CSE8331, Dec 2003.

H. C. Chaudhari received her B.Tech. degree in Computer Science and Technology from Usha Mittal Institute of Technology, SNDT, Mumbai, Maharashtra, India in 2011, pursuing M.Tech. Degree in Computer Science and Engineering from Government College of Engineering, Amravati, Maharashtra, India. Her areas of interest include data mining. Presently, she is engaged in web document clustering.

Prof. K. P. wagh has received his diploma in computer engineering from government polytechnic, Jalgaon, Maharashtra, India and B.E. degree in computer science and engineering from Government College of

Engineering, Aurangabad. Also he has received his M.E. degree in computer science and engineering from Walchand College of Engineering, Sangli. Presently, he is working as assistant professor in Information technology department at Government College of engineering, Amravati.

Dr. P. N. Chatur has received his M.E degree in Electronics Engineering from Government College of Engineering Amravati, Maharashtra, India and Ph.D. degree from Amravari University. He has published twenty papers in international journals. His area of research includes Artificial Neural Network, Data Mining, Data Stream Mining and Cloud Computing. Currently, he is Head of Computer Science and Engineering & Electronics Engineering Department at Government College of Engineering Amravati, Maharashtra, India and is engaged with large database mining analysis and stream mining.