2015 International Conference on Soft Computing Techniques and Implementations- (ICSCTI)
Department of ECE, FET, MRIU, Faridabad, India, Oct 8-10, 2015

Frequent Term Based Text Document Clustering: A New Approach

Manoj Kumar
Assistant Professor
Department of Information and Technology BBDNITM Lucknow, India
manoj.brnwl82@gmail

D K Yadav
Associate Professor
Department of Computer Science
MNNIT Allahabad
Allahabad, India
dky@mnnit.ac.in

Vijay Kumar Gupta
Assistant Professor
Department of Computer Science
BBDNIIT Lucknow
vijaythesoft84@gmail.com

Abstract— Document clustering is used to organize the documents into groups. VSM (Vector Space Model) is a technique used to represent the document as a vector. Working with VSM to cluster the documents is easier.

The main problem with text documents clustering is very high dimensionality of data. A term in the document represents a dimension. To reduce the dimensions of the document vector space, it is preprocessed. The main techniques involved are stemming and term filtering for dimensions reduction of document vectors. After dimensions reduction, term frequency vectors corresponding to each document are generated, where each cell in the term frequency vector represents frequencies of a term. Using proposed method in the paper, each pair of term frequency vectors are compared to find out the similarity value between every two corresponding documents. In this way, three similarity matrices minimum_match, maximum_match and average_match are generated which are further used in various clustering techniques to produce clusters. Clusters produced using proposed approach are compared with that of clusters produced based on cosine similarity in terms of F-measure. Higher values of F-measure for clusters produced using proposed method shows that proposed algorithm is better.

Keywords— K-means, VSM, Vector Space Model, data mining, classification, clustering, document clustering, term frequency, cosine similarity, F-measure, inverse document frequency

I. INTRODUCTION

Clustering divides the objects into meaningful groups. Clustering is unsupervised learning whereas classification is supervised learning. Document clustering is automatic document organization.

For clustering the documents two basic techniques used are partitional and hierarchical clustering. Partitional clustering decomposes the data points into a set of disjoint groups. K-means is a type of partitional clustering algorithm. In ideal cases, the clusters do not overlap. The hierarchical clustering results into a graph called dendrogram. Figure 1 shows the hierarchical clustering of 30 documents where, horizontal axis represents observations and vertical axis represents similarity value. The best part of dendrogram is that it can be cut at any level to get the desired no. of clusters. Hierarchical methods are usually classified into Divisive and Agglomerative methods.

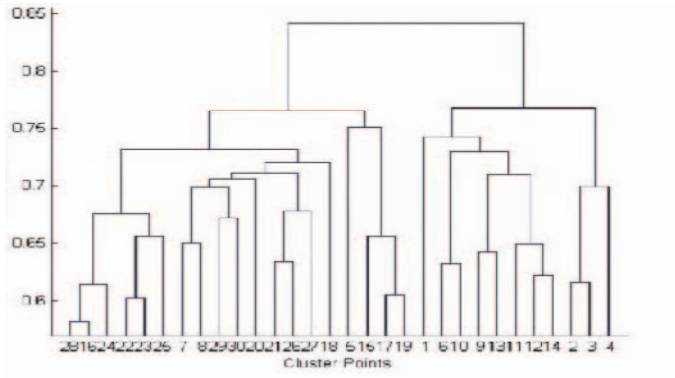

Fig 1: An example of Hierarchical cluster structure

The Vector Space Model and Document Clustering

VSM (Vector Space Model) was introduced in early seventies as a model for automatic indexing. VSM has now become the standard model for document clustering. In VSM a vector represents the frequency of each term in the document.

$$\overrightarrow{D_{tf}} = (tf_1, tf_2, \dots\dots\dots, tf_n) \quad (1)$$

Where tf_i is the number of times i^{th} term occurs in the document. The *tf-idf* score in document *j* for a term at i^{th} position is computed as

$$tf - idf(i,j) = tf(i,j) * idf_i \quad (2)$$

Where *tf(i,j)* is the term frequency for term *i* in document *j*. idf_i is the inverse document frequency for a term t_i expressed as

$$\mathrm{idf_i} = \frac{\log|D|}{|\mathrm{d}: \mathrm{t_i} \in \mathrm{d}|} \quad (3)$$

|D| is the total no. of documents and $|d: t_i \varepsilon d|$ is the no. of documents in which term t_i exists.

idf (inverse document frequency) gives more importance to the terms that are rare across the documents and less to those that are more common. The *tf-idf* weighting scheme will increase the weight of terms that have frequent occurrence in a smaller number of documents. *tf-idf* lowers the weight of those terms that are frequently occurring over the entire set.

Offline document clustering process can be divided into four stages outlined below:

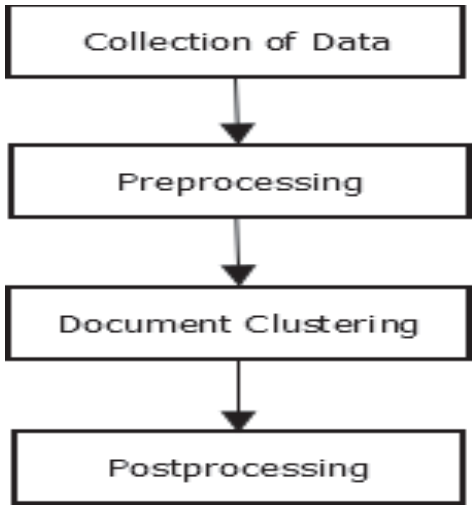

Fig 2: Process of Document Clustering.

Collection of Data involves collecting documents, indexing and storing them. The main techniques involved for collecting the documents are indexing, crawling, and filtering etc. Preprocessing is done to reduce the dimensions of the document. Documents can be represented in many ways like, Graphical Model, Vector-Model etc. Preprocessing converts data in such a form that is easier to be clustered. The steps involved in preprocessing are term filtering and stemming. After preprocessing, the proposed clustering algorithm is applied that is the real focus of this paper. In post-processing clusters prepared in previous steps are applied to various recommendation applications.

II. EVALUATION OF CLUSTER QUALITY

In cluster analysis evaluation of the cluster produced is one of the most important issues. Evaluating the clustering result shows how well the clustering is performed and how good the produced clusters are. Evaluation of the clusters produced is the most difficult task in cluster analysis. Two ways for evaluating the clusters are External and Internal measures.

In internal measure objective function used with the goal of maximizing intra-cluster similarity (similarity between documents within a cluster) and minimizing the inter-cluster similarity (similarity between documents from different clusters). No external knowledge is required in this. But for external quality measures external knowledge about the data is required. Two key measures of external evaluation are Entropy and F-measure. The main focus in the paper is F-measure. Entropy discussion is left purposely for the readers of the paper.

F-measure: F-measure is used to test accuracy of test performed. It uses both recall and precision for this purpose. P (Precision) for i^{th} cluster and j^{th} class is precision (i, j) $=p_{ij}$. *R* (Recall) for i^{th} cluster and j^{th} class is *recall (i,j)*= m_{ij}/mj

m_j = no. of objects in class *j*.

F-measure of cluster *C* is:

$$F - measure = \frac{2 * P * R}{P + R} \tag{4}$$

Where,

$$R = \frac{TP}{TP + FN} \tag{5}$$

$$P = \frac{TP}{TP + FP} \tag{6}$$

Where,

- TP (True Positive): Similar documents that fall in same cluster (Correctly Identified)
- FP (False Positive): Dissimilar documents that fall in same cluster (Incorrectly Identified)
- TN (True Negative): Dissimilar documents that fall in different clusters (Correctly Rejected)
- FN (False Negative): Similar documents that fall in different clusters (Incorrectly Rejected)

Higher value of F-measure indicates better clustering.

III. RELATED WORK

V. Kumar et al. [2] showed the comparison between K-means and Agglomerative hierarchical clustering. Agglomerative hierarchical clustering showed better performance than K-means. Florian Beil et al. [10] proposed algorithms FTC (Frequent Text Clustering) for flat clustering and HFTC (Hierarchical Frequent term based Text Clustering) for hierarchical clustering. P. Ponmuthuramalingam et al. [7] proposed effective dimensionality reduction technique which includes stemming, stopwords, adverbs, verbs and sometimes non-noun verbs removal. Fung B.C.M et al. [6] discussed about the frequent itemsets, which are related with association rule mining. Yanjun Li et al. [8] proposed FCDC; frequent concepts based clustering algorithm rather than based on frequent itemsets. This uses the semantic relationship between words to create concepts.

IV. PROPOSED APPROACH

Cosine similarity between documents in document vector space is the measure of cosine of the angle between them and this is independent of the magnitude between them. Not only is the magnitude of each term of each document considered but also the angle between the documents. This is the most commonly used method to compute similarity between the documents. To compute cosine similarity term frequency vector of the documents is used.

$$similarity = cos(\theta) = \frac{A.B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (7)$$

Length Normalization: Length normalization of each document vector by its Euclidean distance turns the vector into unit vector. Thus, eliminates the length information of the original document. Document length normalization is a way of penalizing the term weights for a document in accordance with its length. Some of the normalization techniques used in information retrieval systems are Cosine normalization, Byte Length normalization etc.

Cosine normalization is the commonly used technique in the vector space model. To calculate cosine normalization L_2 *norm* is calculated first.

$$L_2 norm = ||\vec{x}|| = \sqrt{\sum_i x_i^2} \quad (8)$$

Cosine similarity between two length normalized vectors $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ is dot product of $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$.

$$cos(\overrightarrow{d_1}, \overrightarrow{d_2}) = \overrightarrow{d_1}.\overrightarrow{d_2} = \sum_{i=1}^{|V|} d_{1(i)} * d_{2(i)} \quad (9)$$

Proposed method also uses term frequency vector of each document for the purpose of calculating similarity matrices for the proposed algorithm. In the proposed method, match vector space is normalized by dividing the vector by size of the smaller, larger or average of the two vectors. Match vector between two term frequency vectors is a vector which counts the number of times corresponding term matches between two term frequency vectors.

Term frequency vector algorithm Alg.1 is used to calculate the term frequency of each frequent term in a document. Frequent terms are those terms that occur in at least thresold percentage of total documents. In this way the term frequency vectors corresponding to each document is generated which is further used in proposed method.

minimum_match, maximum_match and average_match

The term frequency vector generated in the previous algorithm is used in the algorithm Alg.2 to calculate the similarity between each pair of term frequency vector and generate the three similarity matrices minimum_match, maximum_match and average_match. Three different matrices are generated based on the size of vectors to be matched. Similarity matrix can be used in many clustering techniques to produce the clusters for e.g. hierarchical, k-means, bisecting k-means, fuzzy c means clustering etc. In this paper the similarity matrices are used in k-means and agglomerative hierarchical clustering to produce the clusters. But k-means is used for evaluation purposes only.

Alg. 1 Generateing the Term Frequency Vectors

Input: FTL: Frequent Term List

colFTL: FTL.length

NoF: Number of Files

IV: Initial Vectors

Output: TV:Term Frequency Vectors

for $i = 1 \rightarrow colFTL$ **do**

  for $j = 1 \rightarrow NoF$ **do**

    $freq_i$= count the frequency of the term in IV_j

    $TV_j \leftarrow TV_j + freq_i$

  end

end

V. EXPERIMENTAL EVALUATIONS

In this work, two data subsets, Reuters Transcribed Subset (RTS) and Mini Newsgroups (MN) of two standard datasets Reuters-21578 and 20 Newsgroups respectively are used. The proposed algorithm, Alg.2 is applied on these datasets. It shows the step-by-step procedure required to generate three similarity matrices. In proposed approach, datasets are pre-processed first in order to remove stopwords, generate frequent term vectors etc., and then apply the proposed algorithm.

Reuters Transcribed Subset (RTS)

Reuters-21578 is the very common dataset used for document clustering. 20 files each from 10 classes from Reuters-21578 are selected to create RTS.

Its details are as follows:

- Number of unique documents = 201
- Number of categories = 10
- Number of unique words after term filtering = 5,817

Mini Newsgroups

This is a very popular dataset used in machine learning methods, and data mining methods etc. Mini Newsgroups is a

subset composed of 100 articles from each newsgroup. Its details are as follows:

- Number of unique documents = 2,000
- Number of categories = 20
- Number of unique words after term filtering = 24,104

Alg. 2 Generateing the Normalized Match Terms Vectors

Input: NoF: number of files

TV: Term Vectors

Output: MTV: Match Terms Vectors

for $i = 1 \rightarrow NoF$ **do**

  for $j = 1 \rightarrow NoF$ **do**

    match ← count the match between corresponding terms of TV_i and TV_j

    sum ← calculate the total count for TV_i and TV_j

    length1=TV_i.totalCount

    length2=TV_j.totalCount

    avgLength=$\frac{length1+length2}{2}$

    if $length1 \leq length2$ **then**

      minLength=length1

      maxLength=length2

    end

    else

      minLength=length2

      maxLength=length1

    end

    $minimum\_match(TV_i, TV_j) = \frac{match}{minLength}$

    $maximum\_match(TV_i, TV_j) = \frac{match}{maxLength}$

    $average\_match(TV_i, TV_j) = \frac{match}{avgLength}$

  end

end

VI. RESULT

Higher F-measure values of the clusters produced using proposed algorithm show that the proposed method is better than cosine similarity. Dimension reduction is less for lower thresold and is more for higher thresold but in case of more dimensionality reduction the resulting document will have less no. of frequent terms resulting into less information about the dataset.

REUTERS TRANSCRIBED SUBSET

Table 1: F-measure Values for RTS dataset

Thresold	cosine similarity	minimum _match	maximum _match	average _match
5	0.483	0.53	0.521	0.573
10	0.512	0.551	0.656	0.618
15	0.536	0.567	0.604	0.495
20	0.58	0.579	0.698	0.609
25	0.574	0.672	0.562	0.578
30	0.604	0.589	0.745	0.695

Table 1 represents the F-measure values of the final clusters based on cosine similarity and proposed algorithm for "Reuters Transcribed Subset" dataset.

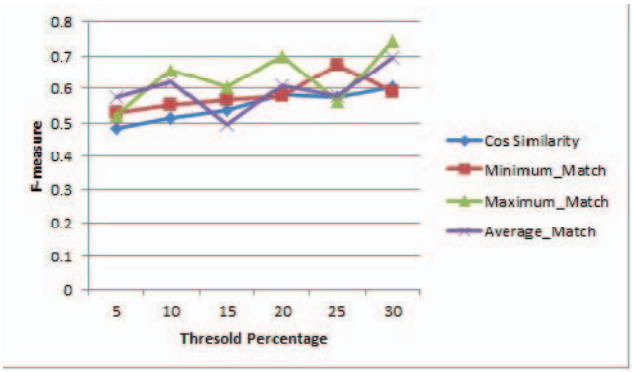

Fig 3: Comparison of F-measure for RTS dataset

The F-measure chart in Figure 3 shows the proposed algorithm is better than cosine similarity for the "Reuters Transcribed Subset" dataset.

Mini Newsgroups

Table 2: F-measure Values for Mini Newsgropups.

Thresold	cosine similarity	minimum _match	maximum _match	average _match
5	0.486	0.56	0.512	0.718
10	0.53	0.633	0.483	0.521
15	0.567	0.604	0.636	0.662
20	0.479	0.598	0.507	0.689
25	0.572	0.662	0.588	0.678
30	0.621	0.745	0.604	0.695

Table 2 represents the F-measures values of the final clusters produced based on cosine similarity and proposed algorithm for "Mini Newsgroups" dataset.

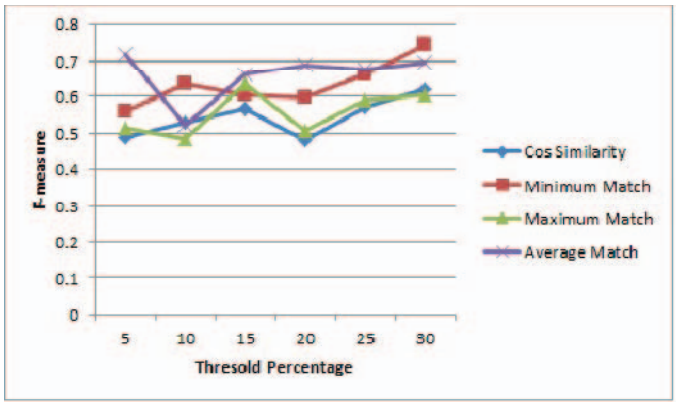

Fig 4: Comparison of F-measure for Mini Newsgroups.

The F-measure chart in Figure 4 shows the proposed algorithm is better than cosine similarity for "Mini Newsgroups" dataset.

VII. CONCLUSION & FUTURE SCOPE

In this paper some existing algorithms have been investigated and a new one is proposed. The conclusion is that it is difficult to get a general clustering algorithm, which can work optimally for all types of datasets. The algorithm is evaluated on two different standard datasets Reuters Transcribed Subset and Mini Newsgroups. The proposed algorithm is compared with the standard similarity measure cosine similarity in respect of these datasets. The F-measure chart of clustering in Figure 3 and Figure 4 based on proposed similarity values shows that the proposed algorithm performs better clustering with higher values of F-measure than one based on cosine similarity. This means the proposed algorithm with varying thresold is better or at least comparable to the cosine similarity for the chosen datasets

The algorithm proposed in this paper is at its very starting stage and there may be many possible improvements that can be implemented. Semantic similarity between the documents can be taken into account. Also proposed algorithm can be applied on some other clustering techniques i.e. bisecting k-means, fuzzy c-means etc. This can also be tested on some large datasets.

REFERENCES

[1] Han J., Kamber M., "Data Mining: Concepts and Techniques," Morgan Kaufmann (Elsevier), 2006.
[2] Steinbach M., Karypsis G., Kumar V., "A Comparison of Document Clustering Techniques," KDD-2000 Workshop on Text Mining, 2000, 203-215.
[3] L. LIU, J. KANG, J. YU, Z. WANG, "A Survey of Document Clustering Techniques," Proceeding of NLP-KE'O5, 2005 IEEE.
[4] Agrawal R., Srikant R, "A fast algorithm for mining association rules," VLDB 94, Santiago de Chile, Chile, 1994, pp.487-499.
[5] Beil F., Ester M., Xu X., "Frequent Term-based Text Clustering," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp.436- 442.
[6] Fung B.C.M., Wang K. and Ester M, "Hierarchical Document Clustering using Frequent Item sets," Proceedings of SIAM International Conference on Data Mining, 2003, pp.180-304
[7] Ponmuthuramalingam.P. Devi.D, "Effective Dimension Reduction Techniques for Text Documents," International Journal on Computer Science and Network Security (ICSNS), Vol 10, No. 7, 2010.
[8] Yanjun Li, Soon M. Chung, John D. Holt, "Text Document clustering based on frequent word meaning sequences," In Data and Knowledge Engineering 2008 pp. 381-404.
[9] Porter M.F., "An Algorithm for Suffix Stripping," Program, Vol. 14, no. 3, pp. 130-137, 1980.
[10] Beil F., Ester M., Xu X., "Frequent Term-Based Text Clustering," Proc. 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD 2002), Edmonton, Alberta, Canada.
[11] Z. Hong, L. J. ling, "Text Clustering Method Based on the Iteration Convergence of Initial Centers," 2012 International Conference on Computer Science and Information Processing (CSIP).
[12] N. K. Nagwani, S. Verma, "A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm," International Journal of Computer Applications (0975 8887) Volume 17 - No.2, March 2011
[13] P. Ponmuthuramalingam, T. Devi, "Effective Term Based Text Clustering Algorithms," International Journal on Computer Science and Engineering (IJCSE) Vol. 02, No. 05, 2010, 1665-1673.
[14] W. Zhang, T. Yoshida, X. Tang, Q. Wanga, "Text clustering using frequent itemsets," Elsevier: Knowledge-Based Systems 23 (2010) 379388.
[15] X. Wang, J. Cao, Y. Liu, S. Gao, X. Deng, "Text Clustering Based on the Improved TFIDF by the Iterative Algorithm," 2012 IEEE Symposium on Electrical and Electronics Engineering (EEESYM).
[16] "http://archive.ics.uci.edu/ml/datasets/Reuters+Transcribed+Subset"
[17] "http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html"