ISSN:0975-9646

Assessment of Document Similarity with Clustering Using Multi View-Points

Sindhudarshini S[#1], Suresh G S[*2]

PG Scholar[#], Associate Professor[*], Dept of CSE,
Channabasaweshwara Institute of Technology
Gubbi, Tumkur, Karnataka, India.

Abstract: **Clustering methods will have some clustering relationship between the documents or objects on which it can be applied. Similarity is a measure to provide similar documents which can be defined precisely or completely. Clustering is a data mining or text mining techniques which is used to analyze datasets by dividing it into meaningful groups. The objects have certain relationship among objects in data sets. The existing clustering algorithms with respect to data mining use single viewpoint similarity measure for partitioned clustering of documents. The drawback is that the cluster cannot reveal the complete set of relationships among documents which says it doesn't make use of fully informative assessment. In this paper a new measure called multi viewpoint similarity is used to find the similarity between the documents. By using multi viewpoints descriptive evaluation can be achieved. The actual study reveals that the theory "multi viewpoint similarity can bring about more instructive relationships among the documents and thus more meaningful clusters are formed" is used in the real time applications where text documents are to be explored or organized regularly.**

Keywords: **data mining, text mining, similarity measure, multi viewpoint similarity measure, clustering methods, single viewpoint.**

I. INTRODUCTION

Data mining is the practice of examining large pre-existing databases in order to generate new information. It is a study of extracting useful patterns from large datasets. Data mining has various techniques such as association rule mining, classification, clustering, regression, sequential pattern discovery, deviation detection. Here clustering is used. Clustering is one of the descriptive data mining techniques which group the similar documents together.

In other words it is a process of grouping the documents into same cluster which are similar to each other than to those in other clusters. Major clustering methods can be classified into the following categories: density-based methods, hierarchical methods, grid-based methods, model-based methods and partitioning methods. This paper focuses on partitioned clustering. K-means is one of the partitioned clustering which is mainly used in industry. It takes two arguments first is the data set or objects and the second is the number of clusters to be formed. One of the application in which K-means algorithm used is credit card fraud detection. In this application it generates clusters offline and makes a model. The important quality of K-means algorithm is that it combines with other algorithm for finest results. It is simple and effective clustering algorithm. The drawback is it is sensitive to cluster size, initialization, and low performance comparatively. Clustering is an optimization process which forms highest quality clusters. K-means has sum of squared-error objective function that uses Euclidean distance.

II. PRIOR WORK

Document clustering is one of the text mining techniques which groups documents into some categories in such a way that there is maximization of intra-cluster document similarity and inter-cluster dissimilarity. Each document which we regarded as clustering is represented as an m-dimensional vector d where m represents the total number of terms present in the given document. These Document vectors are the result of some sort of weighting schemes like TF-IDF (Term Frequency-Inverse Document Frequency). There exist so many approaches for document clustering. It encompass non-negative matrix factorization, information theoretic co-clustering and probabilistic model based method and so on. The cons of these approaches is doesn't use specific measure in finding document similarity. It is found from literature that one of the popular measures is Euclidean distance:

$$Dist(di, dj) = \|di - dj\| \quad (1)$$

Euclidean distance measure is used in k-means algorithm. The motivation of the k-means algorithm is to minimize the distance, according to Euclidean measurement, between documents in the clusters. The centroid is computed as:

$$Min \sum_{r=1}^{k} \sum_{die Sr} \|d_i - c_r\|^2 \quad (2)$$

Another similarity measure which is used in Hi-dimensional documents is cosine similarity measure. It is also used in Spherical K-means (Variance of K-means). Cosine similarity is used to maximize the cosine similarity between cluster's centroid and the documents in the cluster .The difference between Euclidean distance and cosine

similarity measure is that the former concentrates on vector directions while the latter concentrates on vector magnitudes. Another similarity measure is graph partitioning approach. One of the methods in graph partitioning approach is Min-Max cut algorithm which focuses on minimizing centroid function.

$$Min \sum_{r=1}^{k} \frac{D_r^t D}{\|Dr\|^2} \qquad (3)$$

Other graph partitioning methods are Normalized Cut and Average Weight is also used for document clustering. They use pair wise and cosine similarity value for document clustering. Graph partitioning builds nearest neighbour graph first and results in clusters. This clustering is based on Jaccard coefficient which is computed as:

$$Sim_{ejacc}(u_i,u_j) = \frac{ui\,uj}{\|ui\|2+\|uj\|2-ui\,uj} \qquad (4)$$

In Jaccard coefficient both magnitudes and directions is considered compared to cosine similarity and Euclidean distance. This paper focuses on multi viewpoint based similarity measure.

III. MULTI-VIEWPOINT SIMILARITY MEASURE

The main aim for clustering documents is achieved by using multi viewpoint similarity which finds the similar documents. Existing algorithms makes use of one point of reference for clustering text documents whereas multi viewpoint uses more than one point of reference. In this approach similarity between documents is calculated as:

$$Sim(di,dj) = 1/(n-nr\sum Sim(di-dh,dj-dh)) \qquad (5)$$

$$dt,dj \in Sr\ dh \in {}^{S}/_{Sr}$$

The description of this approach is as consider two point di and dj in cluster Sr. Similarity between those two points is viewed from a point dh which is outside the cluster. This similarity is equal to the product of cosine angle between the points. A postulation on which this explanation is based on is dh is not the same cluster as di and dj. When distances are smaller the probability are higher that the dh is in the same cluster. Nevertheless various viewpoints are helpful in increasing the truthfulness of similarity measure there is an opportunity of having that give negative result. Though the possibility of such drawback can be mistreated provided plenty of documents to be clustered.

A series of algorithms are proposed to achieve MVS (multi viewpoint similarity). The following is a procedure for building similarity matrix of MVS. [1]

Procedure BUILDMVSMATRIX (*A*)
2: **for** *r* ← 1: *c* **do**
3: *DS\Sr* ←_*di*/□*Sr di*
4: *nS\Sr* ← |*S* \ *Sr*|
5: **end for**
6: **for** *i* ← 1: *n* **do**
7: *r* ← class of *di*
8: **for** *j* ← 1: *n* **do**
9: **if** *dj* □ *Sr* **then**
10: *aij* ← *dti dj* − *dti DS\Sr nS\Sr* − *dt j DS\Sr nS\Sr* + 1
11: **else**
12: *aij*←*dti dj*−*dti DS\Sr* −*dj nS\Sr* −1 −*dt j DS\Sr* −*dj nS\Sr* −1+1
13: **end if**
14: **end for**
15: **end for**
16: **return** *A* = *{aij} n*×*n*
17: **end procedure**

Algorithm 1: procedure for building MVS similarity matrix

From the consition it is understood that when di is considered closer to dl, still the dl can be considered being closer to di as per MVS. The following algorithm is used for validation purpose. [1]

Require: 0 < *percentage* ≤ 1
1: **procedure** GETVALIDITY (*validity,A, percentage*)
2: **for** *r* ← 1: *c* **do**
3: *qr* ← _*percentage* × *nr*
4: **if** *qr* = 0 **then** _*percentage* too small
5: *qr* ← 1
6: **end if**
7: **end for**
8: **for** *i* ← 1: *n* **do**
9: *{aiv*[1]. . . *aiv*[*n*] *}* ←Sort *{ai*1, . . . , *ain}*
10: *s.t. aiv*[1] ≥ *aiv*[2] ≥ . . . ≥ *aiv*[*n*] *{v*[1], . . . , *v*[*n*]*}* ← permute *{*1, . . . , *n}*
11: *r* ← class of *di*
12: *validity* (*di*) ← |*{dv*[1], . . . , *dv*[*qr*] *}*∩ *Sr*| *qr*
13: **end for**
14: *validity* ←_*n i*←1 *validity* (*di*) *n*
15: **return** *validity*
16: **end procedure**

Algorithm 2: procedure for getting validity score

D ← New document
for each sentence s in D do
w1 ← first word in s
if w1 is not in G, then
Add w1 to G
end if
L ← Empty list {L is a list of matching phrases}
for each word wi ∈ {w2,w3,. . . ,wk} in s do
if wi is not in G, then
Add wi to G
end if
if (wi−1,wi) is an edge in G, then
Retrieve a list of document entries from wi document table that have a sentence on the edge (wi−1,wi)
Extend previous phrase matches in L for sentences that continue along (wi−1,wi)
Add new phrase matches to L
else
Add edge(wi−1,wi) to G
end if
Update sentence path in nodes wi−1 and wi
end for
end for
Output matching phrases list L

Algorithm 3: Document index graph

The final validity is calculated by averaging overall the rows of A. When the validity score is higher, then suitability is more for clustering

IV. SYSTEM DESIGN

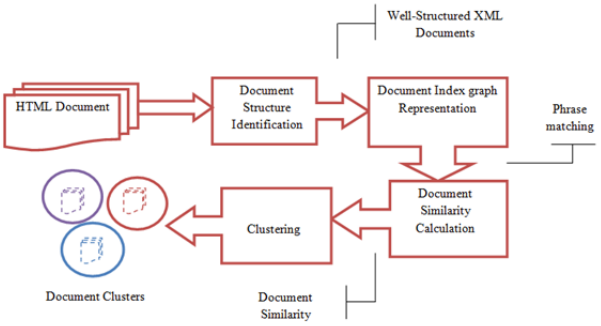

Fig 1: System Architecture.

Initially for the database various set of text documents has been chosen. Text documents have different set of information. In next step keyword identification is performed to choose different keywords. This keyword identification is performed from each text document. Next feature space is constructed in which feature extraction of all the text documents has been performed. Once the feature of the documents is constructed then based on the features of all text documents similarity computation is performed. In this similarity computation it is unwavering that how much each text document is related or similar to other text documents. Once the similarity computation is over then the clustering of the documents is performed. In this clustering process documents of similar type is grouped together to form a cluster. Whole of this process is called mapping documents to cluster.

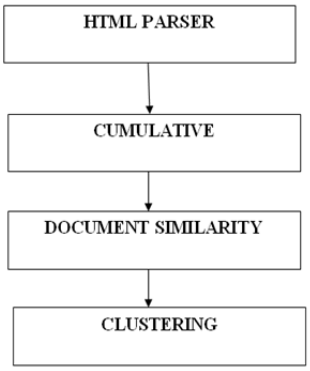

Fig 2: design layout

HTML Parser:

a. When document enters the process state, parsing will be the first step.
b. Parsing is nothing but identification or separation of phrases in a HTML document.
c. In HTML Parser, the raw file is read and it is parsed through all the nodes in the tree structure.

Cumulative document:

a. By finding the references of base document with the entire input document, we can sum all the documents which contain the similar phrases.
b. Henceforth all the documents with their phrases are identified, starting from the base document.

Document Similarity:

Similarity between the documents can be found by the similarity measure:

a. The weights in the cosine-similarity are found from the TF-IDF measure between the phrases of the two documents.
b. It can be done by computing the term weights involved.
c. TF= C/T, where C=number of times a given word appears in a document and T= total number of words in a document, and TF=Term Frequency.
d. IDF= D/DF, where D=total number of documents in a corpus, and DF=total number of documents containing in a given word, and IDF=Inverse document frequency.
e. TFIDF=TF*IDF

Clustering:

a. The similar documents are grouped together in a cluster, if their similarity measure is less than a specified threshold [9].

V. RESULTS

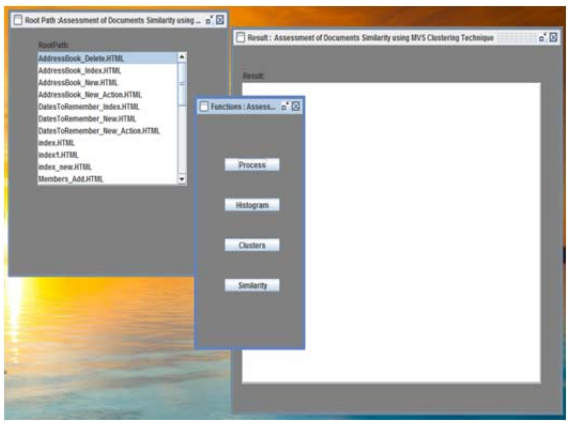

Fig 3: document pre-processing

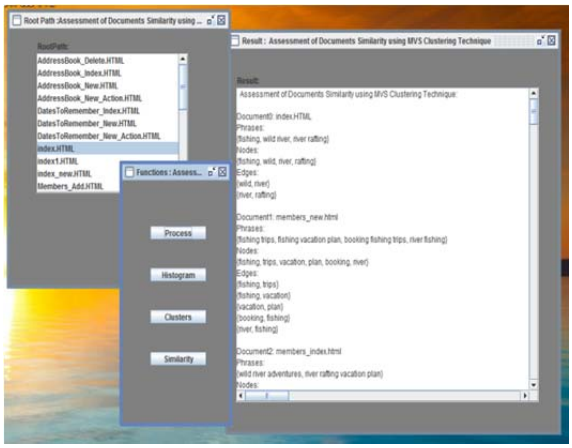

Fig 4: Process

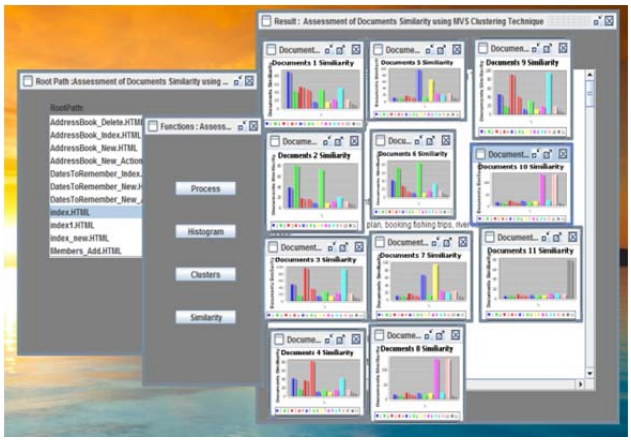

Fig 5: Histogram

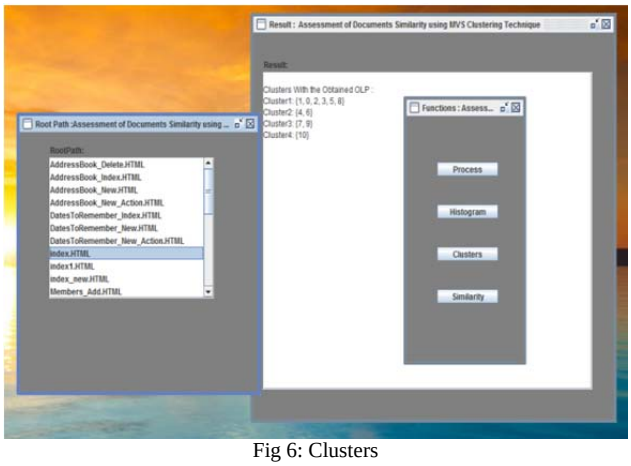

Fig 6: Clusters

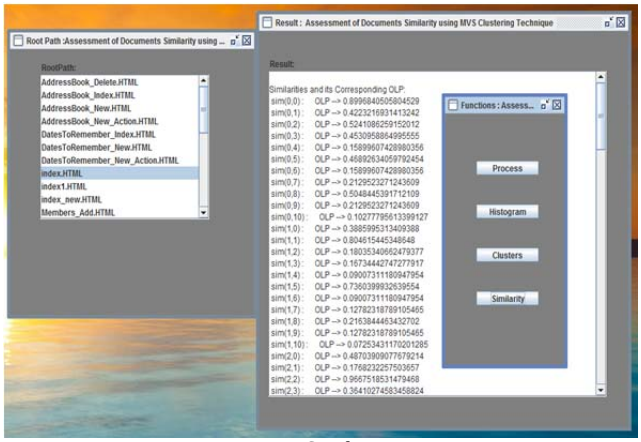

Fig 7: Similarity

VI. CONCLUSION

In this paper, we propose a novel document similarity assessment with clustering using multi view-points. Given a data set, the ideal scenario would have a set of criteria to select an accurate clustering algorithm to apply. Choosing different dimensional space frequency levels leads to different accuracy rate in the clustering results. The similarity measure is capable of providing effective assessment and bestows high quality clusters. The proposed scheme is tested with large data sets with various evolution metrics. The results reveals that the clustering algorithm provides performance that is better than many state-of-the-art clustering algorithm. Similarity measure from multiple viewpoints is the main contribution of this paper. There are many numbers of future researches to extend and improve this work. One is that this work might continue to improve on the efficiency of similarity calculation strategies.

ACKNOWLEDGMENT

We are grateful to express sincere thanks to our faculties who gave support and special thanks to our department for providing facilities that were offered to us for carrying out this paper.

REFERENCES

[1] Clustering with Multiviewpoint-Based Similarity Measure Duc Thang Nguyen, Lihui Chen, Senior Member, IEEE, and Chee Keong Chan, IEEE transactions on knowledge and data engineering, vol. 24, no. 6, june 2012

[2]. I. Guyon, U.V. Luxburg, and R.C. Williamson, "Clustering: Science or Art?," Proc. NIPS Workshop Clustering Theory, 2009.

[3]. I. Dhillon and D. Modha, "Concept Decompositions for Large Sparse Text Data Using Clustering," Machine Learning, vol. 42, nos. 1/2, pp. 143-175, Jan. 2001.

[4]. S. Zhong, "Efficient Online Spherical K-means Clustering," Proc. IEEE Int'l Joint Conf. Neural Networks (IJCNN), pp. 3180-3185, 2005.

[5]. A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with Bregman Divergences," J. Machine Learning Research, vol. 6, pp. 1705-1749, Oct. 2005.

[6]. E. Pekalska, A. Harol, R.P.W. Duin, B. Spillmann, and H. Bunke, "Non-Euclidean or Non-Metric Measures Can Be Informative," Structural, Syntactic, and Statistical Pattern Recognition, vol. 4109, pp. 871-880, 2006.

[7]. M. Pelillo, "What Is a Cluster? Perspectives from Game Theory," Proc. NIPS Workshop Clustering Theory, 2009.

[8]. D. Lee and J. Lee, "Dynamic Dissimilarity Measure for Support Based Clustering," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 6, pp. 900-905, June 2010.

[9]. A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Clustering on the Unit Hypersphere Using Von Mises-Fisher Distributions," J. Machine Learning Research, vol. 6, pp. 1345-1382, Sept. 2005.

[10]. W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization," Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Informaion Retrieval, pp. 267-273, 2003.