ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Sentence-Similarity Based Document Clustering Using Birch Algorithm

Dr.A.Vijaya Kathiravan, P.Kalaiyarasi

Assistant Professor in Computer Applications, PG and Research Department of Computer Science,

Government Arts College [Autonomous], Salem, Tamil Nadu, India

Research Scholar, PG and Research Department of Computer Science, Government Arts College [Autonomous],

Salem, Tamil Nadu, India

ABSTRACT: Document clustering is the process of automatically grouping the related documents into clusters. Instead of searching entire documents for relevant information, these clusters will improve the efficiency and avoid overlapping of contents. Relevant document can be efficiently retrieved and accessed by means of document clustering. When compared with hard clustering, birch algorithms allow patterns to belong to all the clusters with differing degrees of membership. Birch algorithm is important in domains such as sentence clustering, since a sentence is related to more than one theme or topic present within a document or set of documents. In our proposed system, birch clustering algorithm operates on cluster Start with initial threshold and inserts points into the tree. Results obtained while applying the algorithm to sentence clustering tasks demonstrate that birch algorithm is capable of identifying overlapping clusters of semantically related sentences and its performance improvement can be proved by comparing with k-means. Performance measures document clustering and its application in document summarization.

KEYWORDS: document clustering, flexirank algorithm, birch algorithm, similarity measure, accuracy, document similarity

I. INTRODUCTION

Clustering is nothing but given a number of objects or individuals, each of which is described by a set of numerical measures, devise a classification scheme for grouping the objects into a number of classes such that objects within classes are similar in some respect and unlike those from other classes. The number of classes and the characteristics of each class are to be determined.

Clustering is an automatic learning technique aimed at grouping a set of objects into subsets or clusters. The goal is to create clusters that are coherent internally, but substantially different from each other. In plain words, objects in the same cluster should be as similar as possible, whereas objects in one cluster should be as dissimilar as possible from objects in the other clusters.

BIRCH is designed for clustering a large amount of numerical data by integration of hierarchical clustering (at the initial micro clustering stage) and other clustering methods such as iterative partitioning (at the later macro clustering stage). It overcomes the two difficulties of agglomerative clustering methods: (1) scalability and (2) the inability to undo what was done in the previous step.

BIRCH introduces two concepts, clustering feature and clustering feature tree (CF tree), which are used to summarize cluster representations. These structures help the clustering method achieve good speed and scalability in large databases and also make it effective for incremental and dynamic clustering of incoming objects.

II. DOCUMENT CLUSTERING

Document clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering is generally considered to be a centralized process.

Copyright to IJIRCCE DOI: 10.15680/ijircce.2015.0305055

Examples of document clustering include web document clustering for search users. The application of document clustering can be categorized to two types, online and offline. Online applications are usually constrained by efficiency problems when compared to offline applications.

Document clustering technique

1) Birch (Balenced Itrative Reducing and Clustering using Hierarchies) hierarchical clustering

BIRCH is designed for clustering a large amount of numerical data by integration of hierarchical clustering (at the initial micro clustering stage) and other clustering methods such as iterative partitioning (at the later macro clustering stage). It overcomes the two difficulties of agglomerative clustering methods: (1) scalability and (2) the inability to undo what was done in the previous step.

2) Partitional clustering

Given *D*, a data set of *n* objects, and *k*, the number of clusters to form, a partitioning algorithm organizes the objects into *k* partitions (*k* _ *n*), where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are “similar,” whereas the objects of different clusters are “dissimilar” in terms of the data set attributes.

3) Hierarchical clustering

Connectivity based clustering, also known as hierarchical clustering, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

Connectivity based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance to) to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances) or UPGMA ("Unweighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

III. LITRATURE REVIEW

Given an integer K, K-means partitions the data set into K non overlapping clusters. It does so by positioning K "centroïds” or "prototypes" in densely populated regions of the data space. Each observation is then assigned to the closest centroid ("Minimum distance rule"). A cluster therefore contains all observations that are all closer to a given centroid than to any of the other centroids (lower image of the illustration). Limitations for Handling Empty Clusters: One of the problems with the basic K-means algorithm given earlier is that empty clusters can be obtained if no points are allocated to a cluster during the assignment step. If this happens, then a strategy is needed to choose a replacement centroid, since otherwise, the squared error will be larger than necessary. Another one limitations for Outliers: When outliers are present, the resulting cluster centroids (prototypes) may not be as representative as they otherwise would be and thus, the SSE will be higher as well [1].

In [3] K-mean clustering user need to specify the number of cluster in advanced. K-mean clustering algorithm performance depends on initial centroids that why the algorithm doesn’t have guarantee for optimal solution. K-means has several limitations which are listed below:

1. Scalability: It scales poorly computationally.
2. Initial means: The clustering result is extremely sensitive to the initial means.
3. Noise: Noise or outliers deteriorates the quality of the clustering result.

The algorithm is simple and has nice convergence but there are number of problems with this. Some of the weaknesses of k-mediods are

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The result is circular cluster shape because based on distance.

Hierarchal clustering one type for Clustering Using Representatives (CURE) clusters are represented by a fixed number of well-scattered points instead of a single centroid. Second, the representatives are shrunk toward their cluster centers by a constant factor. At each iteration, the pair of clusters with the closest representatives is merged. The use of multiple representatives allows CURE to deal with arbitrary-shaped clusters of different sizes, while the shrinking dampens the effects of outliers and noise. CURE uses a combination of random sampling and partitioning to improve scalability.

The single-link hierarchical method measures the similarity between two clusters by the similarity of the closest pair of data points belonging to different clusters. Unlike the centroid/medoid-based methods, this method can find clusters of arbitrary shape and different sizes. However, it is highly susceptible to noise, outliers, and artifacts.

In this paper [10] document – document similarity matrix and multiple kernel c-means algorithm based on clustering for information retrieval when performing document clustering using k-means algorithm, two major problems can happen. The first problem is finding the similarity among data. The second problem is that it would require more iteration.

BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of the CF tree, which removes sparse clusters as outliers and group sentence dense clusters into larger ones.

IV. THE PROPOSED WORK

In the proposed system, initially preprocessing is done for the document in which stop words and stem words are removed. Stop words are removed by means of comparing with the database that contains stop words. Stem words are removed through Porter stemming Algorithm. Porter's algorithm provides how the words can be reduced to their root words. Once after preprocessing, similarity between sentences is calculated using Text rank measure. Similarity calculation is mainly based on number of terms common between two sentences by number of words present in both sentences. Based on sentence similarity, sentences with highest Page Rank value are taken through Page Rank algorithm. Page Rank algorithm provides the importance of sentence i.e. how many times the sentence appears in the document .Then birch algorithm is applied.

A. The proposed system architecture

a. Preprocessing: Preprocessing is nothing but subject (data) to preliminary processing

b. Remove stop words: Preprocessing for dataset is done to remove the stop words and stem words which are considered as less important and to improve quality and efficiency of data. Many of the most frequently used words in English are useless in Information Retrieval (IR) and text mining. These words are called 'Stop words'. Stop-words, which are language-specific functional words, are frequent words that carry no information (i.e., pronouns, prepositions, conjunctions). Examples of such words include 'the', 'of',' and', 'to', etc. These stop words are get stored in the database. Dataset (famous quotations) is loaded in to another database. Here stop words in data set (famous quotation) is removed by comparing with the stop word database.

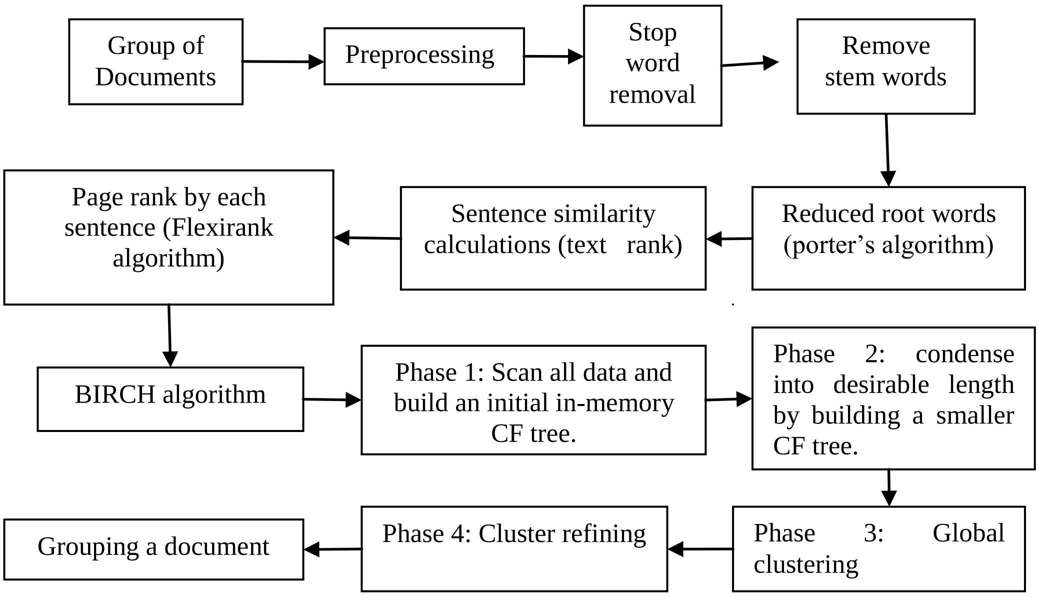
Group of Documents
Preprocessing
Stop word removal
Remove stem words
Reduced root words (porter's algorithm)
Sentence similarity calculations (text rank)
Page rank by each sentence (Flexirank algorithm)
BIRCH algorithm
Phase 1: Scan all data and build an initial in-memory CF tree.
Phase 2: condense into desirable length by building a smaller CF tree.
Phase 3: Global clustering
Phase 4: Cluster refining
Grouping a document

Fig 1: The proposed system architecture

c. Remove stem words: Stemming or lemmatization is a technique for the reduction of words into their root. Many words in the English language can be reduced to their root word or base form e.g. agreed, agreeing, and agreement belong to agree.

d. Porter's stemming: The porter stemming algorithm is a process for removing suffixes form words in English. Removing suffixes automatically is an operation which is an operations which is especially useful in the field of information retrieval.

e. Sentence similarity calculations: The ability to accurately judge the similarity between natural language sentences is critical to the performance of several applications such as text mining, question answering, and text summarization. Given two sentences, an effective similarity measure should be able to determine whether the sentences are semantically equivalent or not, taking into account the variability of natural language expression. Similarity between two sentences is provided by Text rank measure [18].

f. Page ranking calculations in a sentence for flexirank algorithm. The FlexiRank algorithm operates on a set of web pages returned by a web crawler and gives a ranking of the pages as output.

It operates according to the following steps:

- Select attributes based on user demand: Based on the users' demand the algorithm chooses a set of properties of a web page. Some properties are chosen irrespective of the users' demand. Examples of such mandatory properties are Relevance weight, Hub weight and Authority weight. The other attributes are chosen based on user demand to provide an accurate ranking. Examples of such optional attributes are number of hyperlinks, number of images, properties of anchor text, etc.
- Measure the attributes: The selected attributes are measured for each web page.
- Calculate rank: The rank is calculated by taking a weighted average of the measured values. The weight assigned to each attribute is based on users' demand.

g. Birch based document clustering steps

Phase:

1. Scan all data and build an initial in-memory CF tree.
2. Phase 2: condense into desirable length by building a smaller CF tree.

3. Phase 3: Global clustering
4. Cluster refining – this is optional, and requires more passes over the data to refine the results

Phase 1

1. Start with initial threshold and insert points into the tree
2. If run out of memory, increase threshold value, and rebuild a smaller tree by reinserting values from older tree and then other values
3. Good initial threshold is important but hard to figure out
4. Outlier removal – when rebuilding tree remove outliers

Phase 2

1. Optional
2. Phases 3 sometime have minimum size which performs well, so phase 2 prepares the tree for phase 3.
3. Removes outliers, and grouping clusters.

Phase 3

1. Problems after phase 1:
– Input order affects results
– Splitting triggered by node size
2. Phase 3:
– cluster all leaf nodes on the CF values according to an existing algorithm
– Algorithm used here: agglomerative hierarchical clustering

Phase 4

1. Optional
2. Do additional passes over the dataset & reassign data points to the closest centroid from phase 3
3. Recalculating the centroids and redistributing the items.
4. Always converges (no matter how many time phase 4 is repeated)

h. K-means clustering evaluation

It is a partition clustering algorithm and it is very effective in smaller datasets. First select k initial centers based on desired number of clusters. The user can specify k parameter value. Each data point is assigned to nearest centroid and the set of points assigned to the centroid is called a cluster. Each cluster centroid is updated based on the points assigned to the cluster. The process will be repeated until the centroids remain the same or no point changes clusters. In this algorithm mostly Euclidean distance is used to find distance between data points and centroids. The main drawback of K-means algorithm is the quality of the clustering results highly depends on random selection of the initial centroids.

V. RESULT AND DISCUSSION

The dataset used here is famous quotation dataset where a large number of documents are available for usage and they are analyzed offline. The performance evaluate for Birch Algorithm and K-means algorithm.

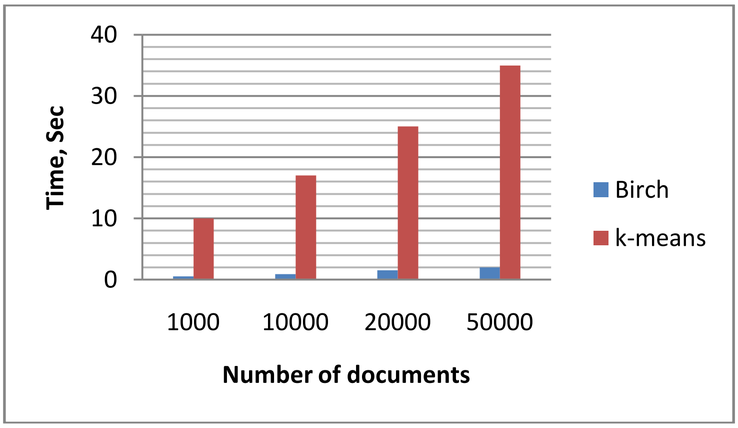

Fig2: Performance comparison between Birch and k-means based on entropy

When number of documents increases, time and second is less in k-means when compared with birch. The performance level achieved in birch is 90% more than k-means.

VI. CONCLUSION

When compared to the existing work, the proposed work avoids content overlap and able to achieve superior performance to k-means algorithms when externally evaluated on a challenging data set of famous quotations. Birch hierarchical clustering algorithm that can be applied to any relational clustering problem, and its application to several non-sentence data sets has shown its performance to be comparable to k-means benchmarks. Like any clustering algorithm, the performance of birch will ultimately depend on the quality of the input data, and in the case of sentence clustering this performance may be improved through development of better sentence similarity measures, which may in turn be based on improved word sense disambiguation, etc.

REFERENCES

[1] Kehar singh, Dimple Malik, Naveen Sharma "Evolving Limitations In K-means Algorithm In Data Mining And Their Removal", IJCEM International Journal of Computational Engineering And Management Volume 12, April 2011, ISSN(online):2230-7893.

[2] Pritesh Vora, Bhavesh Oza "A Survey On K-mean Clustering And Particle Swarm Optimization" International Journal Of Science And Morden Engineering (IJISME) ISSN: 2319-6386, Volume 1, issue-3, Feb 2013.

[3] Read T.Aldahdooh, Wesam Ashour "DIMK- means Distance Based Initialization Method For K-means Clustering Algorithm" Intelligent Systems And Applications, 2013, o2,41-51, published online January 2013 in MCES.

[4] Raghuvira Pratap.A, K.Suvarna Vani, J.Rama Devi, Dr.K.Nageswara Reo "An Efficient Density Based Improved K-mediods Clustering Algorithm" (IJACSA) International Journal Of Advanced Computer Science And Applications, vol2, no.6, 2011.

[5] Monica Sood, Shilpi Bansal "K-medoids Clustering Technique Using Bat Algorithm" International Journal of Applied Information Systems (IJAIS), ISSN: 2249-0868, Volume 5, no 8, June 3013.

[6] Deepti Siso Dia, Lokesh Singh, Sheetal Sisodia, Khushboo Sexena "Clustering Techniques- A Brief Survey Of Different Clustering Algorithms" International Journal of Latest Trends In Engineering Technology (IJLTET) 2004.

[7] George Karypis, Eui-Hong (Sam) Han, Vipin Kumar, "Ghameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling", to 0018-9162/99/$10@1999 IEEE.

[8] Poonam Yadav "Document- Document Similarity Matrix and Multiple-kernel C-means Algorithm Based Web Document Clustering for Information Retrieval" International Journal of Advanced Research in Computer and Communication Engineering vol3, issue 10, Oct 2014.

[9] Pankaj Jajoo "Document Clustering" 2008.

[10] Deepi Gupta, Nidhi Tyagi, Kornal Kumar Bhatia "Retrieval of Web Document Using A Fuzzy Hierarchical Clustering" International Journal of Computer Applications (00975-8887) volume 5 no 6 Aug 2010.

[11] Teiwant Singh, Mr.Manish Mahajan "Performance Comparison of Fuzzy C-means with Respect to Other Clustering Algorithm" International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 4, Issue 5, May 2014.

[12] Sumit Goswami and Mayank Singh Shishodia "A Fuzzy Based Approach to Text mining and Document Clustering" 2011.

[13] K.sathiyakumari, V.Pream Sudha, G.Manimekalai "Unsupervised Approach for Document Clustering Using Modified Fuzzy C Means Algorithm" International Journal of Computer and Organization Trends, volume issue3 – 2011.

[14] Chun – Cling Chen, Frank S.C. Tseng, Tyne Liang "Mining Fuzzy Frequent Item sets for Hierarchical Document Clustering" Information Processing and Management, 46 (2010) 193-211.

[15] G.thilagavathi, J.Anitha, K.Nethra "Sentence-Similarity Based Document Clustering using Fuzzy Algorithm" International Journal of Advance Foundation and Research in Computer (IJAFRC) volume 1, issue 3, march 2014, ISSN 2348-4853.

[16] Tejwant Singh, Mr. Manish Mahajan "Performance Comparison of Fuzzy C Means with Respect to Other Clustering Algorithm" International Journal of Advanced Research in Computer Science and Software Engineering, vol4, issue 5, may 2014, ISSN:2277 128X.

[17] Ilya Karpor, Alexander Goroslevsking, "Application of BRICH to Text Clustering" Preceding of the 14th all Russian Conference Digital Libraries: Advanced Methods and Technologies, digital collections-RCDL 2012, October 15-18.

[18] NidalIsmael, MahMoud, WesamAshour, "Improved Multi Threshold Birch Clustering Algorithm" International Journal of Artificial Intelligence and Applications for Smart Devices, Volume 2, no 1,(2014), pp 1-10.

[19] Yogita Rani, Manju, Harish Rohil, "Comparative Analysis of BIRCH and CURE Hierarchical Clustering Algorithm using weka 3.6.9", The SIJ Transactions on Computer Science Engineering and its Applications (CSEA), Volume 2, no 1, January –February 2014.

[20] Debajyoti Mukhopadhyay, Pradipta Biswas, "FlexiRank: an Algorithm Offering Flexibility and Accuracy for Ranking the Web pages" 2003.

[21] Tian Zhang, Ramakrishnan, Miron Livng, "BIRCH: A new data Clustering Algorithm and its Applications" Data mining Knowledge Discovery, issue 2, pp 141-182, 1997, springer.

[22] Gupta Mamta, Rajarai Anand, "Comparison of Algorithms for Document Clustering" Computational Intelligence and Communication Networks (CICN),2014 International Conference on 14-16 Nov 2014, pp 541-545, print ISSN 978-1-4799-6928-9 [IEEE].