Survey On Text Clustering In Distributed Network

Chithra Purushothaman
Dept. of Computer Science & Engg.
Sree Buddha College of Engg.
Alappuzha, Kerala, INDIA

Lakshmi S
Dept. of Computer Science & Engg.
Sree Buddha College of Engg.
Alappuzha, Kerala, INDIA

***Abstract*— Text clustering has emerged as a widely used technique with the increase in large number of documents that is getting accumulated day by day in various fields like news groups, government organizations, Internet and digital libraries. Text clustering is the process of grouping similar documents into clusters .Text clustering is an important technique for improving the quality of information retrieval in both centralized and distributed environment. Most of the existing text clustering algorithms is designed for central execution, which are not work well on distributed environment. Therefore, many specialized algorithms were proposed for distributed and peer to peer to network. This paper briefly describes some of the approaches for text clustering in distributed network.**

Keywords— document clustering, distributed clustering, p2p network, text clustering.

I. INTRODUCTION

Data clustering is a data mining technique that performs the grouping of similar data objects into clusters, such that objects in the same cluster are similar, and those in different clusters are dissimilar. The objective of clustering is to partition an unstructured set of objects into clusters (groups). In clustering, groups are not predefined. Grouping is accomplished by finding similarities between data according to characteristics found in the actual data. Clustering has been used in many different areas and there exist a multitude of different clustering algorithms for different settings. To use most clustering algorithms two things are necessary:

- an object representation,
- a similarity (or distance) measure between objects.

A clustering algorithm finds a partition of a set of objects that fulfills some criterion based on these conditions. Clustering is an unsupervised learning method. The result (the clustering, the partition) is based solely on the object representation, the similarity measure and the clustering algorithm. If these correspond to the users understanding the result might well be an intuitive and useful clustering. Clustering can be applied to many types of data. It is very useful in the text domain, where the objects to be clusters can be of different granularities such as documents, paragraphs, sentences or terms. In document (text) clustering the objects are texts or documents. It is the process of grouping similar documents into clusters based on their textual content. Documents in one cluster belong to a certain topic, while different clusters represent different topics. To represent these, the vector space model is commonly used.

Text clustering is an established technique for improving quality in information retrieval, for both centralized and distributed environments. Most existing text clustering algorithms are designed for central execution. They require that clustering is performed on a dedicated node, and are not suitable for distributed network. Centralized approach operate on a principle of gathering all data into a central site, then running an algorithm against that data for text clustering. Such an approach is fundamentally inappropriate for the distributed and P2P environment. In fact, the long response time, lack of proper use of distributed resource, and the fundamental characteristic of centralized clustering algorithms do not work well in distributed environments.

P2P networks are gaining growing status in many distributed applications such as file-sharing, web caching, network storage, searching and indexing of relevant documents and P2P network threat analysis. It enables a collection of nodes (peers) to share computer resources in a decentralized manner. Collectively the peers already store a huge amount of widely varying data collected from different sources. If this data, distributed over large number of peers, can be integrated, it represents a very valuable data repository that, upon mining, may give very exciting and useful results. Clustering in such network is challenging because data is inherently distributed and no participant has the capacity to collect and process all data. Therefore, many specialized algorithms for distributed and P2P clustering have been developed. This paper briefly describes various approaches proposed for distributed networks.

II. DOCUMENT CLUSTERING

Document clustering groups similar documents that form a coherent cluster, while documents that are different have separated apart into different clusters. Clustering of text documents plays a vital role in efficient Document Organization, Summarization, Topic Extraction and Information Retrieval. The definition of a pair of documents being similar or different is not always clear and normally varies with the actual problem setting. For example, when clustering research papers, two documents are regarded as similar if they share similar thematic topics. When clustering is employed on web sites, we are usually more interested in clustering the component pages according to the type of information that is presented in the page. For instance, when dealing with universities" web sites, we may want to separate

ISSN: 2278 – 7798

professors "home pages from students" home pages, and pages for courses from pages for research projects. This kind of clustering can benefit further analysis and utilize of the dataset such as information retrieval and information extraction, by grouping similar types of information sources together. Accurate clustering requires a precise definition of the closeness between a pair of objects, in terms of either the pair wised similarity or distance. A variety of similarity or distance measures have been proposed and widely applied, such as Euclidean distance, cosine similarity, Jaccard coefficient, Pearson correlation coefficient. Given the diversity of similarity and distance measures available, their effectiveness in text document clustering is still not clear.

Given a set S of n documents, we would like to partition them into a pre-determined number of k subsets S_1, S_2…. S_k, such that the documents assigned to each subset are more similar to each other than the documents assigned to different subsets. Document clustering techniques mostly rely on single term analysis of the document data set, such as the Vector Space Model. To achieve more accurate document clustering, more informative features including phrases and their weights are particularly important in such scenarios.

- Document clustering is particularly useful in many applications such as automatic categorization of documents, grouping search engine results, building taxonomy of documents, and others.
- Each document in a corpus corresponds to an m-dimensional vector d, where „m" is the total number of terms.
- Document vectors are often subjected to some weighting schemes, such as the standard Term Frequency-Inverse Document Frequency (TF-IDF), and normalized to have unit length.

A. Document Preprocessing Steps:

Preprocessing is a very important step since it can affect the result of a clustering algorithm. So it is necessary to preprocess the data sensibly. Preprocessing has the several steps that take a text preprocessing of documents Creating feature vectors using document as input and output as a set of tokens to be used in feature vector.

- Tokenization: A document is treated as a string (or bag of words), and then partitioned into a list of tokens.
- Removing stop words: Stop words are frequently occurring, insignificant words. This step eliminates the stop words.
- Stemming word: This step is the process of conflating tokens to their root form.

B. Document Representation:

Generating N-distinct words from the corpora and call them as index terms (or the vocabulary). The document collection is then represented as a N-dimensional vector in term space. Computing Term weights Term Frequency Inverse Document Frequency. Compute the TF-IDF weighting.

C. TFIDT Analysis:

By taking into account these two factors: term frequency (TF) and inverse document frequency (IDF) it is possible to assign weights to search results and therefore ordering them statistically. Put another way a search result's score Ranking is the product of TF and IDF: TFIDF = TF * IDF where:

- TF = C / T where C = number of times a given word appears in a document and T = total number of words in a document.
- Document IDF = D / DF where D = total number of documents in a corpus, and DF = total number of documents containing a given word.

III. SIMILARITY MEASURES

The similarity measure reflects the degree of closeness or separation of the target objects and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand, and there is no measure that is universally best for all kinds of clustering problems. Moreover, choosing an appropriate similarity measure is also crucial for cluster analysis, especially for a particular type of clustering algorithms. Recalling that closeness is quantified as the distance/similarity value, we can see that large number of distance/similarity computations are required for finding dense areas and estimate cluster assignment of new data objects. Therefore, understanding the effectiveness of different measures is of great importance in helping to choose the best one. The following are the different similarity measures:

A. Cosine Similarity

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is bounded in [0, 1].

$$\cos\theta = \frac{A.B}{|A|\,|B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Where, A- term count value in the document1 and B- term count value in the document2. The value of the cosine similarity lies between 0-1.the value 0 represents the documents are not similar and 1 represents the documents are similar.

B. Jaccard Coefficient

The Jaccard coefficient, also known as Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects.

ISSN: 2278 – 7798

$$T(A,B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B}$$

Where, A- term count value in the document1 and B- term count value in the document2.

F. Pearson coefficient

$$r = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2} \sqrt{\sum_i (y_i - \overline{y})^2}}$$

IV. RELATED WORK

Several approaches have been proposed for P2P text clustering. K-Means is one of the most frequently used clustering algorithms because of its low complexity and high clustering quality, particularly for text clustering. The basic K-Means algorithm can be summarized as follows: (1) Select k random starting points as initial centroids for the k clusters. (2) Assign each document to the cluster with the nearest centroid. (3) Recomputed the centroid of each cluster as the mean of all cluster documents. (4) Repeat steps 2-3 until a stopping criterion is met, e.g., no documents change clusters anymore. The direct distribution of K-Means algorithm for document clustering in large network is not efficient. So its distributed versions are used.

Researches on P2P clustering introduces an exact local algorithm for monitoring k-means clustering described in [36]. The k-means monitoring algorithm monitors the distribution of centroids across peers and thus make the k-means process aware of when to update the clusters. The K-means monitoring algorithm consists of two phases:

- First phase is monitoring the data distribution which is carried out by an exact algorithm.
- Second phase is centroid computation by a centralization approach.

However this algorithm cannot be used to solve distributed clustering.

Eisenhardt et al. [1] proposed one of the first P2P clustering algorithms. The approach is based on two algorithms.

- K-Means clustering algorithm is used to do the categorization of the documents.
- A probe/echo mechanism is used to distribute the task through the P2P network and propagate the results back to the initiator of the clustering.

Each synchronization round corresponds to a K-Means iteration. Each site carries out the following algorithm at each iteration. One site initiates the process by marking itself as engaged and sending a probe message to all its neighbors. The message also contains cluster centroids currently maintained at the initiator site. the first time a node receives a probe (from a neighbor site p with centroids Cp), it marks itselfs engaged, sends a probe message(along with Cp) to all its neighbors (expect the origin of the probe), and updates the centroids in Cp using its local data as well as computing a weight for each centroid based on the number of data points associated with each cluster. If a site receives an echo from a cluster. If a site receives an echo from a neighbor p (with centroids Cp and weights Wp), it merges Cp and weights Wp with its current centroids and weights. Once a site has received either a probe or echo from all neighbors, it sends an echo along with its local centroids and weights to the neighbor from which it received its first probe. Once the initiator node has received echos from all its neighbors, it can generate centroids which take into account all datasets at all sites.

The algorithm distributes K-Means computation by broadcasting the centroid information to all peers. Due to this centroid broadcasting, it leads to heavy traffic and congestion in the network and it does not scale to large networks.

Hsiao and King[2] propose a similarity information retrieval system called Meteoroggragh for structured P2P overlays without relying on message flooding. Meteorograph characterizes an item as a vector in the vector space model and stores the item in a single structured overlay. To map items into the structured overlay, each item in Meteorograph is transformed to a single value called the absolute angle. Two items are similar if they share some common keywords, and the two corresponding vectors in the vector space have a very small angle. By controlling the locations, represented by absolute angles, in which items are stored, Meteorograph can rapidly locate a search item. Moreover, it can aggregate similar items together at nearby locations in the overlay. This approach avoids broadcasting by employing a DHT to index all clusters using manually selected terms. This approach requires extensive human interaction for selecting the terms, and the algorithm cannot adapt to new topics.

Hammouda and Kamel [3] propose a hierarchical topology for distributing K-Means. Clustering starts at the lowest level of the hierarchy, and the local solutions are aggregated until the root peer is reached. Central to this hierar chical architecture design is the formation of neighborhoods. A neighborhood is a group of peers forming a logical unit of isolation. Peers in a neighborhood can communicate directly but not with peers in other neighborhoods. Each neighborhood has a supernode. Communication between neighborhoods is achieved through their respective supernodes. This model reduces flooding problem usually encountered in large P2P network. The Heirarchical Peer to Peer Clustering (HP2PC) algorithm is a distributed iterative clustering process. It is a centroid based clustering algorithm, where a set of centroids that describe the data set in that neighborhood.

In HP2PC, each neighbor converges to a set of centroids that describe the data set in that neighborhood. Other neighborhoods, either on the same level or at higher levels of the hierarchy, may converge to another set of centroids. Once a neighborhood converges to a set of centroids, those centroids are acquired by the supernode of that neighborhood. The supernode, in turn as part of its higher level neighborhood, collaborates with its peers to form a set of centroids for its neighborhood. This process continuous hierarchically until a set of centroids is generated at the root of the hierarchy. This algorithm has the disadvantage that clustering quality

ISSN: 2278 – 7798

decreases noticeably for each aggregation level, because of the random grouping of peers at each level. Therefore, quality decreases significantly for large networks.

Januzaj et.al [10] proposed clustering the data locally and extracting suitable representatives out of these clusters. These representatives are sent to a global server where the complete clustering based on the local representatives is restored. This approach is characterized by carrying out local clustering quickly and independently from each other.

In [4], another distributed clustering algorithm is proposed, where peers exchange cluster summaries containing extracted key phrases from the cluster. The algorithm called Core Phrase is proposed for extracting keyphrases from multidocument sets or clusters, with no prior knowledge about the documents. The algorithm is based on finding a set of core phrases that best describe a multi-document set. The algorithm works by first intersecting all documents together to generate a list of candidate key phrases for describing the topic of the documents. The extracted candidate keyphrases are then analyzed for frequency, span over the document set, and other features. Each phrases is assigned a score based on its features, then the list is ranked and the top phrases are output as the descriptive topic of the document cluster. Even though the cluster summaries are compact, each peer needs to send them to all other peers, requiring O(n2) messages for a network of size n. This approach is therefore only suitable for very small networks.

Two approximate local algorithms LSP2P and USP2P for P2P K-means clustering are described in [6] and [7]. Local synchronization based P2P K-means distribute the centroids using gossiping. The centers at each peer are updated making use of the information received from their immediate neighbors. This algorithm produces highly accurate clustering results but no analytical guarantees on this clustering accuracy is provided. So a second algorithm Uniform Sampling based P2P K-means is developed. Probabilistic guarantees are provided through sampling. USP2P assumes the network as static and found to achieve high accuracy. Both these algorithms are based on the assumption that data distribution among the peers is uniform. So it may not work well for large size networks. Also since text collections in P2P networks may not be uniformly distributed, these algorithms do not suit for text collection.

A frequently used technique [9] focus on constructing an index over a Distributed Hash Table(DHT) that maps terms to documents, and enables locating the most similar documents for each term. DHT [8] is used for the distribution of the inverted index over all participating peers. Each peer analyzes its own collection, and extracts a set of terms, normally after performing basic stemming and filtering of stopwords. For each extracted term, the peer executes a DHT lookup to locate the responsible peer in the network, and posts there its contact details, with the term and term score.

V. CONCLUSION

In this paper we have summarized the research work that has done related to the clustering document clustering and similarity measure. Group of independent servers (usually in close proximity to one another) interconnected through a dedicated network to work as one centralized data processing resource. Clusters are capable of performing multiple complex instructions by distributing workload across all connected servers. Clustering improves the system's availability to users, its aggregate performance, and overall tolerance to faults and component failures.

REFERENCES

[1] M. Eisenhardt, W. Muller, and A. Henrich, “Classifying documents by distributed P2P clustering.” in INFORMATIK, 2003.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] H.C. Hsiao and C.T. King, “Similarity discovery in structured P2P overlays,” in ICPP, 2003.

[3] K. M. Hammouda and M. S. Kamel,” Hierarchically distributed peer-to-peer document clustering and cluster summarization,” IEEE trans. Knowl. Data Eng., vol. 21, no. 5, pp. 681-698, 2009.

[4] K. M. Hammouda and M. S. Kamel,”Distributed collaborative web document clustering using cluster keyphrase summaries,” Information Fusin, vol. 9, no. 4, pp. 465-480, 2008.

[5] L. T. Nguyen, W. G. Yee, and O. Frieder, “Adaptive distributed indexing for structured peer-to-peer networks,” in CIKM, 2008.

[6] S.Datta, C. Ginnella, and H. Kargupta, “K-Means clustering over a large, dynamic net,” in SDM, 2006.

[7] S. Datta, C. R. Giannella, and H. Kargupta, “Approximate distributed K-Means clustering over a peer-to-peer network,” IEEE TKDE, vol. 21, no. 10, pp. 1372–1388, 2009.

[8] O. Papapetrou, W. Siberski, and W. Nejdl, “PCIR: Combining DHTs and peer clusters for efficient full-text P2P indexing,” Computer Networks, vol. 54, no. 12, pp. 2019–2040, 2010.

[9] O. Papapetrou, W. Siberski, and N. Fuhr,”Decentralised Probablistic Text Clustering,” IEEE trans. Knowl. Data Eng., vol. 24, no. 10, 2012.

[10] E. Januzaj, H.P Kriegal, and M.Pfeifle, “Towards effective and efficient distributed clustering large data sets”, ICDM, Melburne, 2003

ISSN: 2278 – 7798