International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Efficient Text Clustering for Distributed Network

Chithra Purushothaman[1], Lakshmi S[2]

[1, 2]Department of Computer Science & Engineering, Sree Buddha College of Engineering, Alappuzha, Kerala, India

Abstract: ***Text clustering is an important technique for improving the quality of information retrieval in both centralized and distributed environment. Most of the existing texts clustering algorithms are designed for central execution, which are not work well on highly distributed environment. In this paper, an algorithm called probabilistic text clustering for distributed network such as peer to peer network is proposed. This algorithm achieves high scalability for assigning documents to clusters. It enables a peer to compare each of its documents only with very few selected clusters, maintain cluster quality.***

Keywords: text clustering, k- means, p2p network, DHT, centroid

1. Introduction

Data clustering is a data mining techniques that perform the grouping of similar data objects into clusters, such that objects in the same cluster are similar, and those in different clusters are dissimilar. Clustering can be applied to many types of data. The focus of this thesis is on clustering text documents, also known as document clustering. 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.

Text clustering is an established technique for improving quality in information retrieval, for both centralized and distributed environments. It is especially useful in highly distributed environments such as distributed digital libraries and peer-to-peer (P2P) information management systems, since these environments operate on large-scale document collections, scattered over the network.

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 large scale distributed networks. Therefore, specialized algorithms for distributed and P2P clustering have been developed such as [1], [2], [3], [4]. But these approaches are either limited to a small number of nodes, or they focus on low dimensional data only.

In a distributed environment, such as P2P network, data sources are distributed over a large, dynamic network. So, clustering in such network is challenging, firstly because the data is distributed and no participant has the capacity to collect and process all data, and secondly because of high churn rate, affecting availability of content and of computational nodes. For these types of systems, we require a P2P algorithm that can cluster distributed and highly dynamic text collections, without overloading any of the participating peers, and without requiring central participation.

In order to reduce the load in such network, we have to reduce the number of comparisons between documents and cluster. Our approach achieves this by applying probabilistic pruning: Instead of considering all clusters for comparison with each document, only a few most relevant ones are taken into consideration. We apply this core idea to K-Means and one of the frequently used text clustering algorithms. In our proposed algorithm, called Probabilistic Text Clustering (PTC) for peer to peer systems reduces the number of required comparisons by providing clustering quality.

2. Related Works

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.

Several works focus on P2P clustering [1], [2], [3], [4]. 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, and 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. 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] avoid 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. This algorithm has the disadvantage that clustering quality decreases noticeably for each aggregation level, because of the random grouping of peers at each level. Therefore, quality decreases significantly for large networks. Already for a network of 65 nodes

organized in three levels, the authors report a drastic drop in quality.

In [4], another distributed clustering algorithm is proposed, where peers exchange cluster summaries containing extracted key phrases from the cluster. Even though these summaries are compact, each peer needs to send them to all other peers, requiring O (n^2) 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 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[9]. DHT 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.

3. Problem Statement

Document clustering in distributed network introduce the problem of information overflow. This can be achieved by reducing the number of comparison between document and cluster.

4. Objectives

The following are the main objectives of our work:

- Compare data only with a few selected clusters providing cluster quality.
- To improve quality of information retrieval
- To display the result according to the highest ranking.

5. Methodology

For text clustering in peer to peer network, we propose an algorithm called probabilistic text clustering (PTC). PTC consists of two activities: cluster indexing and document assignment. Cluster indexing is performed by cluster holders.

Algorithm: PTC: Clustering the documents

1) **for** Document d in document set **do**
   PRESELECTION STEP
2) CandClusters ← CandidateClustersFromDHT()
   FULL COMPARISON STEP
3) RemainingClusters ← FilterOut(CandClusters)
4) **for** Cluster c in RemainingClusters **do**
5) Send termVector(d) to ClusterHolder(c)
6) Sim[c] ← Compute similarity(d,c)
7) **end for**
8) Assign(d, cluster with maximum similarity)
9) **end for**

The peer that is a cluster holder maintains the centroid and document assignment for one cluster. These peers create cluster summaries and index them in the underlying DHT, using the most frequent cluster terms as keys. The second activity, document assignment, consists of two steps, preselection and full comparison. In the preselection step, the peer holding d retrieves selected cluster summaries from the DHT index, to identify the most relevant clusters (Alg. 2, Line 2). Preselection filters out most of the clusters. In the full comparison step, the peer computes similarity score estimates for d using the retrieved cluster summaries. Clusters with low similarity estimates are filtered out (Line 3), and the document is sent to the few remaining cluster holders for full similarity computation (Lines 4-7). Finally, d is assigned to the cluster with the highest similarity (Line 8). This two-stage filtering algorithm reduces drastically the number of full comparisons. Both cluster indexing and document assignments are repeated periodically to compensate churn.

The fig.1 shows the overview of the system. It shows that each peer has three roles. First has document holder, which takes care of clustering its document. Second, it participates in the underlying DHT by holding part of the distributed index. Third, a peer may become a cluster holder, which maintain the centroid and document assignment for one cluster.

A. Inititialisation

Assumes that a peer from the network starts the algorithm by selecting k peers randomly to act as cluster holders. These cluster holders choose one of their documents as initial centroid, and publish the cluster summary to the DHT. Then, the initiating peer uses broadcasting over DHT to notify all participating peers to start clustering.

B. Indexing of Cluster Summaries

Cluster holders are responsible for indexing the summaries of the clusters in the DHT. Each cluster holder periodically recomputes the cluster centroid, using the documents assigned to the cluster at the time. It also recomputes a cluster summary and publishes it to the DHT index, using selected cluster terms as keys. This enables peers to efficiently identify the relevant clusters for their documents. For this identification, it is sufficient to consider the most

frequent terms of a cluster c as keys. We use Topterms to denote the set of terms.

Steps

1. DHT lookup for top terms of d
2. Retrieve all relevant clusters
3. Compare d with top relevant clusters
4. Assign d to best cluster

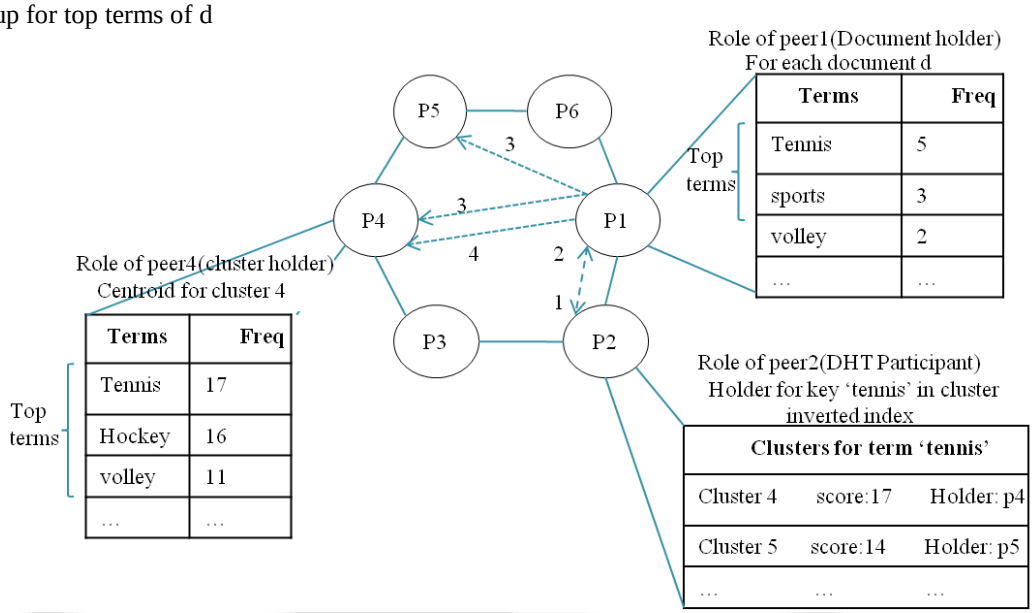

Figure 1: System overview

Document Assignment to Clusters

Each peer is responsible for clustering its documents periodically. Clustering of a document consists of two steps:

(a) the preselection step, where the most promising clusters for the document are detected, and, (b) the full comparison step, where further clusters are filtered out, and the document is fully compare with the remaining clusters and assigned to the best one.

Preselection step. Consider a peer p, clustering document d. Let Topterms (d) denote all terms in d. For each term t in Topterms(d), performs a DHT lookup to find the peer that holds the cluster summaries for t (Fig. 1, Step 1). It then contacts that peer directly to retrieve all summaries published using t as a key (Step 2). All responses are then merged, and list with the retrieved cluster summaries is constructed. This list is referred as the preselection list, and denote as C_{pre}.

Full comparison step. Peer p then sends the term vector of d to the candidate cluster holders for performing full document-cluster comparison, and retrieves the comparison results (Fig. 1, Step 3). To avoid sending the document to all cluster holders in C for comparison, p uses the cluster summaries contained in C_{pre} to filter out the clusters not appropriate for the document at hand.

*C. **Filtering strategy***

Filtering is performed by estimating Jaccard coefficient between a document and each of the clusters. The Jaccard coefficient between document and cluster centroid is defined as

$$\text{Sim}(d, c) = \sum_{t \in d} \frac{TF(t,d)\ X\ TF(t,c)}{|d|^2\ X\ |c|^2 - [TF(t,d) X\ TF(t,c)]}$$

where |d| and |c| are the corresponding document/cluster length, and TF denotes the term frequency of the term in the document/cluster. The Jaccard Coefficient is a similarity measure and ranges between 0 and 1. It is 1 when d and c are similar and 0 when d and c are dissimilar. Jaccard coefficient measures find more coherent clusters.

The filtering process works as follows. Peer p sends the document vector to the first cluster holder in C_{pre}, denoted as $c_{selected}$, and compute the Jaccard coefficient Sim(d, $c_{selected}$). It then removes from C_{pre} the summary of $c_{selected}$. The process is repeated until C_{pre} is empty. The document is finally assigned to the cluster with the highest Jaccard coefficient.

6. Conclusion

Probabilistic Text Clustering (PTC) algorithm is used for efficient clustering of document in distributed network such as P2P network. Clustering in such network is done in such a way that network load cannot be increased. In order to reduce the load in such network, we have to reduce the number of comparisons between document and clusters. This objective is achieved by PTC algorithm.PTC algorithm for a peer to peer system reduces the number of required comparisons by providing clustering quality. Instead of considering all clusters for comparison with each document, only a few most relevant ones are taken into consideration. This algorithm uses Distributed Hash Table (DHT), which is one of the most fundamental and frequently used components in P2P systems. In future, we are try to do clustering of document by considering meaning of words in P2P network.

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.