International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471
Vol. 3 Issue 1, January-2014, pp: (24-32), Impact Factor: 1.147, Available online at: www.erpublications.com

Document Clustering Approaches using Affinity Propagation

Aditi Chaturvedi[1], Dr. Kavita Burse[2], Rachana Mishra[3]
[1]M. Tech. Scholar, CSE Dept. OCT, Bhopal, India
[2]ECE Dept., OCT, Bhopal, India
[3]CSE Dept., OCT, Bhopal, India

Abstract: Document clustering as an unsupervised approach extensively used to navigate, filter, summarize and manage large collection of document repositories like the World Wide Web (WWW). Recently, Document clustering is the process of segmenting a particular collection of texts into subgroups including content based similar ones. The purpose of document clustering is to meet human interests in information searching and understanding. Nowadays all paper documents are in electronic form, because of quick access and smaller storage. So, it is a major issue to retrieve relevant documents from the larger database. This work will study the key challenges of the clustering problem, as it applies to the text domain. Also will discuss the key methods used for text clustering, and their relative advantages.

Keywords: Document Clustering, Feature Extraction, Document Representation Model, Suffix Tree, Similarity Measure.

Introduction

Document clustering has long been studied as a post retrieval document visualization technique to provide an intuitive navigation and browsing mechanism by organizing documents into groups, where each group represents a different topic. In general, the clustering techniques are based on four concepts: data representation model, similarity measure, clustering model, and clustering algorithm. Most of the current documents clustering methods are based on the Vector Space Document (VSD) model. The common framework of this data model starts with a representation of any document as a feature vector of the words that appear in the documents of a data set. A distinct word appearing in the documents is usually considered to be an atomic feature term in the VSD model, because words are the basic units in most natural languages (including English) to represent semantic concepts. In particular, the term weights (usually tf-idf, term-frequencies and inverse document-frequencies) of the words are also contained in each feature vector. The similarity between two documents is computed with one of the several similarity measures based on the two corresponding feature vectors, e.g., cosine measure, Jaccard measure, and Euclidean distance.

Clustering is the process of organizing data objects into a set of disjoint classes called clusters. Objects that are in the same cluster are similar among themselves and dissimilar to the objects belonging to other clusters. Document clustering is the task of automatically organizing text documents into meaningful clusters or group, In other words, the documents in one cluster share the same topic, and the documents in different clusters represent different topics.

Clustering is an example of unsupervised classification. Classification refers to a procedure that assigns data objects to a set of classes. Unsupervised means clustering does not depend on predefined classes and training examples while classifying the data objects.

Clustering is a crucial area of research, which finds applications in many fields including bioinformatics, pattern recognition, image processing, marketing, data mining, economics, etc. Cluster analysis is one of the primary data analysis tools in data mining. Clustering algorithms are mainly divided into two categories: Hierarchical algorithms and Partition algorithms. A hierarchical clustering algorithm divides the given data set into smaller subsets in hierarchical fashion. A partition clustering algorithm partition the data set into desired number of sets in a single step.

A divisive clustering starts with one cluster of all data points and recursively splits into the most appropriate clusters. The process continues until a stopping criterion is achieved. There are two main issues in clustering techniques. At first, finding the optimal number of clusters in a given dataset and secondly, given two sets of clusters, computing a relative measure of goodness between them. For both these purposes, a criterion function or a validation function is usually applied. In conventional clustering objects that are similar are allocated to the same cluster while objects differ are put in different clusters. These clusters are hard clusters. In soft clustering an object may be in more than two or more clusters.

1. Collection of Data includes the processes like crawling, indexing, filtering etc which are used to collect the documents that need to be clustered, index them to store and retrieve in a better way, and filter them to remove the extra data, for example, stop words.

2. Preprocessing consists of steps that take as input a plain text document and output a set of tokens (which can be single terms or n-grams) to be included in the vector model. These steps typically consist of:

   2.1 Filtering is the process of removing special characters and punctuation that are not thought to hold any discriminative power under the vector model. This is more critical in the case of formatted documents, such as web pages, where formatting tags can either be discarded or identified and their constituent terms attributed different weights [1].

   2.2 Stemming the process of reducing words to their base form, or stem. For example, the words “connected", “connection", “connections" are all reduced to the stem “connect." Porter's algorithm is the de facto standard stemming algorithm.

   2.3 Stopword removal A stopword is defined as a term, which is not thought to convey any meaning as a dimension in the vector space (i.e. without context). A typical method to remove stopwords is to compare each term with a compilation of known stopwords. Another approach is to first apply a part-of-speech tagger and then reject all tokens that are not nouns, verbs, or adjectives.

   2.4 Frequent word removal Pruning removes words that appear with very low frequency throughout the corpus. The underlying assumption is that these words, even if they had any discriminating power, would form too small clusters to be useful. A pre-specified threshold is typically used, e.g. a small fraction of the number of words in the corpus. Sometimes words which occur too frequently (e.g. in 40% or more of the documents) are also removed.

Problem Formulation

Given a document collection D={d1,d2,d3,….dN} which contains N documents, there is need to sub-group [2] the documents based on the semantic of the text contents present in a Document , assuming we require K such sub-groups, the clustering process generates C={c1,c2,…ck} clusters, with each ci being non empty.
Document clustering is still a developing field which is undergoing evolution. It started off on the popular vector based approach where documents were treated as a bag of words and clustering criteria was the presence of common words in the documents. Several modifications were applied on this method to improve this method as the result set would only provide us information on what words were present in a group of documents, not the actual content or context of the documents. There was a need of more intuitive ways of clustering that would provide us sound knowledge of the content present inside the documents.

Related Work

There are three factors that effectively influence the cluster’s quality as well as clustering performance. These are namely data representation, similarity measure and the clustering technique that merges the clusters using the similarity measure [3-6]. Vector Space Model (VSM) is a data representation scheme used in most of the Web clustering techniques. In this model a document is represented as a multi dimensional vector of words where each word represents one dimension. Each word is assigned a weight based on TFIDF scheme. The similarity between two documents is usually computed using the various similarity measures such as cosine similarity, Pearson correlation coefficient or Jaccard Coefficient etc[7, 8]. Clustering methods based on this scheme consider only the single word representation of documents hence ignoring most valuable word proximity information [4]. Moreover, clustering techniques based on vector space model do not support incremental processing [9]. Capenato et. al. stated in their survey of web clustering engines that incremental processing improves the efficiency when applied in the clustering methods [10]. The most related work that takes into account the information about proximity of words and phrase based analysis in an incremental way is Suffix Tree clustering (STC) [11].
STC is an incremental linear time (O(n)) algorithm[18] that classifies the documents sharing phrases or the suffix of a phrase into one cluster. The group of these documents is identified by constructing a Suffix tree which represents various phrases and their all possible suffixes. A Suffix Tree is a data structure widely used in diverse applications [12]. A suffix Tree is a Compact trie data Structure representing all the suffixes of a text phrase [12], where a phrase means a sequence of words not the grammatical structure of a sentence [10]. The major advantage of STC is that it makes an efficient use of phrases rather than considering just words and it allows clusters to overlap which is quite practical in the real world as a document can contain information regarding multiple topics and can belong to more than one cluster [1].
Various efforts are made to improve the efficiency of Suffix Tree based clustering by identifying the problems in the original STC. One effort is done by Jongkol et. al. who proposed the Semantic Suffix Tree(SSTC)[9], by combining the semantic similarity, in the original suffix Tree. They used the Wordnet to derive the semantic similarity and used it along with the string matching in the construction of suffix Tree. Their purpose was to identify the phrases having same context but consisting of different words, for example, the context of two phrases ”doctor and nurse” and ”physician and nanny” is similar but Suffix Tree considers them differently. Another extension of STC is the Semantic Hierarchical Online Clustering (SHOC) algorithm [13], which used Suffix Arrays to extract the frequent phrases and singular value decomposition (SVD) technique to identify the cluster contents. Lingo algorithm [14] is further extension to the SHOC algorithm that combines the common phrases with the latent semantic analysis to categorize the search results into coherent clusters. A newer version of Lingo is the Semantic Lingo algorithm [15] that applied synonym in the result

snippets to increase the cluster quality. To solve the problem of large clusters with poor quality in STC, Chim et. al. proposed NTSC algorithm. NTSC combined the vector space model with the Suffix Tree to calculate similarity between pair wise documents to increase the quality of large size clusters[5, 6]. Jianhua wang et. al. proposed a cluster merging algorithm that combined the cosine similarity with the non overlapping parts of the clusters to take into consideration the similarity between the non overlapping parts of the clusters[12].
Daniel Crabtree et. al. improved upon the STC by efficiently identifying the relation between the user query and the clusters. They used the Cluster Description Similarity along with the cluster overlapping in a new cluster splitting method to overcome the problem of cluster chaining [16]. To improve the clustering performance in STC, Daniel et. al. devised Extended Suffix Tree Clustering (ESTC)[1] to select a small number of clusters from a larger set of Clusters returned by the STC. They introduced a new cluster ranking function based on heuristics and a new cluster selection algorithm. Their work is based on a new cluster scoring method to select only the top ranked most relevant clusters. Archana Kale improved the cluster labeling by assigning the words of phrases (on average 3 words for each cluster) having highest frequencies to the cluster as labels [17]. Jiangning et. al. extended the STC for the Chinese web page clustering by introducing the Chinese synonyms in the suffix tree to improve cluster quality[18].

Hierarchical Clustering is preferred over non-hierarchical Clustering because in non-hierarchical clustering a central point, also called centroid, needed to be choosen randomly and the distance from that point is calculated to group documents (with less distance) in one cluster[19]. Finding this central point poses a big challenge. That is why non hierarchical methods are not very popular. A comparative work on both the methods is done by [20]. Florian Beil et. al. introduced two clustering algorithms FTC(non-hierarchical) and HFTC(hierarchical) in [20], based on the concept of frequent Term Set and analyzed their behavior.

They used the association rule mining to identify the frequent terms in documents to group them into clusters. Agglomerative Hierarchical Clustering (AHC) is a widely used bottom up clustering algorithm. Many researchers have efficiently used this method in a information retrieval and data and Web Clustering [21]. There is a group of methods that is used for Agglomerative hierarchical Clustering. Most common of these methods are Single Linkage, Complete Linkage, Group Average Linkage and Wards Method. All of these methods differ in the approach of similarity calculation that guides the selection of the most similar pair of clusters.

Recent research has developed new methods for estimating the semantic distance between two terms or words. Rudi L. Cilibrasi et. al. introduced a new similarity measure, called Normalized Google Distance(NGD) [21], to effectively capture the semantic similarity between words and phrases based on information distance and Kolmogorove Complexity. Later on, Alberto J Evangelista et. al. reviewed the work of Rudi L. Cilibrasi to improve their distance function through elimination of random data[22]. We adopted this method to estimate the similarity between two clusters rather than just two words.

Document Clustering Algorithms

Clustering is thus preformed after the documents matching the query are identified. Consequently, the set of thematic categories is not fixed they are created dynamically depending on the actual documents found in the results. Secondly, as the clustering interface is part of a search engine, the assignment of documents to groups must be done efficiently and on-line. For this reason it is unacceptable to download the full text of each document from the Web – clustering ought to be performed based solely on the snippets returned by the search service.

A. Agglomerative Hierarchical Clustering (AHC) Algorithm

The basic process of hierarchical clustering:

1. If you have n items them make n clusters and assign each item to a cluster. Each cluster should have just one item.
2. Find the most similar pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute similarities between the new clusters and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N

The main problem with AHC [23], that, they are very slow with large amount of data provided and also very sensitive with halting criterion that is, by mistake it can merge valuable clusters into one cluster. Also they do not scale well. They can never undo what was previously done. With outliers it performs poorly.

B. K-Means Algorithm

This algorithm is based on the center locations [24]. It first finds out the k cluster center location. Then each data point finds out which center is closest to it. Each center finds the centroid of the points and jumps to there. The main benefit of K-means algorithm is that, it is capable to produce overlapping clusters. Its main disadvantage is that it is

most effective when the desired clusters are approximately spherical with respect to the similarity measure used. There is no reason to believe that documents should fall into approximately spherical clusters.

C. Suffix Tree Clustering (STC)

STC includes 2 main steps [25]. First it searches for all sets of documents that share a common phrase. They are found by suffix tree data structure. In second step we merge these phrases into cluster. The merge process is dependent on the percentage of the documents that contain both phrases. It also allows overlapping clusters.STC uses simple cluster definition. Also, STC is a fast incremental linear time algorithm which makes it suitable for web search clustering. It is faster than K Means. The main benefit of Suffix Tree Clustering is that it uses phrases to provide concise and meaningful descriptions of groups. But needs some thresholds for cluster formation and they turn out particularly difficult to tune. Its main disadvantage is it removes longer high quality phrases and use only shorter phrases. Finally, if a document does not include any of the extracted phrases or just some parts of them, it will not be included in the results although it may still be relevant

.

D. Semantic Hierarchical Online clustering (SHOC)

The Semantic Online Hierarchical Clustering [26] is a web search results clustering algorithm that uses variation of the Vector Space Model called Latent Semantic Indexing (LSI) and uses phrases in the process of clustering. Unlike STC, SHOC improves the quality of label. STC gives incomplete labels while SHOC gives complete phrases. With SHOC documents can belong to several clusters. SHOC includes two key concepts: Complete phrases and definition of continuous clusters. It should meet the three requirements: Semantic, Hierarchical, and Online. It has three steps:

1. Data collection and cleaning
2. Feature extraction and
3. Identifying and organizing clusters.

Problems with SHOC

The problem with SHOC [26] is that it provides only vague comments on the values of thresholds of their algorithm and the method which is used to label the resulting clusters. It uses the singular value decomposition. So it may create unintuitive, random continuous clusters. It might be because of the input snippets used.

E. Lingo Algorithm

The Lingo algorithm is used by the Carrot2 web searcher and is based on complete phrases and LSI [27]. Lingo is an enhancement of SHOC and STC and unlike most of the algorithms, it first discover descriptive names for the clusters and then, assigns the documents into appropriate clusters. One disadvantage with this algorithm is that the topic separation phase usually requires algebraic transformations that demand a lot of computing time, using Singular Value Decomposition

COMPARISON OF WEB DOCUMENT CLUSTERING ALGORITHMS

Algorithm	Cluster Diversity	Cluster labels	Scalability	Time Complexity	Advantages	Disadvantages
Agglomerative Hierarchical Clustering (AHC) [23]	not very robust towards outliers	Most frequent terms	Low	Single link and group average: O(n2) Complete link: O(n3)	Simple	-Slow when applied to large document collections. -Sensitive to halting criterion. -Poor performance in domains with many outliers.
K- means [24]	Low, small (outlier) clusters rarely highlighted	One-word only, may not always describe all documents in the cluster	Low, based on similar data structures as Lingo	O(nkt)(k:initialclusters,t: iterations)	-Efficient and simple. -Suitable for large datasets.	-Very sensitive to input parameters.
Suffix Tree Clustering (STC) [25]	Low, small (outlier) clusters rarely highlighted	Shorter, but still appropriate	High	O(n)	-Incremental -Uses phrases to provide concise and meaningful description of groups.	-Snippets usually introduce noise. -Snippets may not be a good description of a web page.

Semantic Online Hierarchical Clustering (SHOC) [26]	Low	Label that describe the cluster	High	O(n)	-Uses Latent Semantic Indexing (LSI) and phrases in the process of clustering. -Uses suffix array to identify complete phrases. -Allows overlapping clusters. -Provides a method of ordering documents	-Provides only vague Comments on the values of thresholds of the algorithm and the method which is used to label the resulting clusters.
Lingo [27]	High, many Small (outlier) Clusters highlighted	Longer, often more descriptive	Low. For More than about 1000 documents, Lingo clustering will take a long time and large memory	O(n)	-Readable cluster Labels. Overlapping clusters. -Cluster accuracy	-Unable to generate a Hierarchical structure of clusters. -The implementation of lingo is fairly computationally expensive.

Advanced Clustering Techniques

A. Personalized Web Search Engine Using Suffix Tree Clustering

This work [28] proposed system utilizes clustering and re-ranking algorithms in order to organize the web documents and provide an order to the results displayed to the user. Web crawlers are utilized to get the links, images and allied information from the World Wide Web. The fetched documents are further clustered using suffix tree clustering algorithm, which enhances the performance of the web search engine. The results are organized using Page Re-Rank algorithm which considers hyperlink and link structure information to bring an order to the web. The system creates a semantic profile of the user by monitoring and analyzing the users search history. The web documents are fetched by using crawlers [1] and are further organized in to clusters which aid the user in finding the required details faster. Extended suffix tree clustering algorithm is proposed which analyzes the retrieved web documents semantically descriptive meaningful phrases as cluster labels.

B. Hesitant Distance Similarity Measures

Paper [29] presents new approach, Hesitant Distance Similarity Measures for Document Clustering. The proposed Hesitant Distance Similarity Measures approach is based on Fuzzy Hesitant Sets. In this paper we have used fifty Similarity Measures from f1 to f50. The steps, Document collection, Text Pre-processing, Feature Selection, Indexing, Clustering Process and Results Analysis are used.

C. Clustering with New Ranking and Similarity Measures

Paper [30] introduces a new method for ranking base clusters and new similarity measures for comparing clusters. Our STHAC technique combines the Hierarchical Agglomerative clustering method with phrase based Suffix Tree clustering to improve the cluster merging process. Experimental results have shown that STHAC outperforms the original STC as well as ESTC(our precious extended version of STC) with 16% increase in F-measure. This increase in F-measure of STHAC is achieved due to its better filtering of low score clusters, better similarity measures and efficient cluster merging algorithms. STHAC, Suffix Tree based Hierarchical and Agglomerative Clustering that has made four key contributions to the conventional. Suffix Tree Clustering to improve overall cluster quality and hence clustering performance.

D. Multiviewpoint-Based Similarity Measure

Similarity between a pair of objects can be defined either explicitly or implicitly. Paper [31] proposes a Multiview point-based Similarity measuring method, named MVS. Theoretical analysis and empirical examples show that MVS is potentially more suitable for text documents than the popular cosine similarity. Based on MVS, two criterion functions, IR and IV , and their respective clustering algorithms, MVSC-IR and MVSC-IV , have been introduced. Compared with other state-of-the-art clustering methods that use different types of similarity measure, on a large number of document data sets and under different evaluation metrics, the proposed algorithms show that they could provide significantly improved clustering performance. The key contribution of this paper is the fundamental concept of similarity measure from multiple viewpoints. Future methods could make use of the same principle, but define alternative forms for the relative similarity in (10), or do not use average but have other methods to combine the relative similarities according to the different viewpoints. Besides, this paper focuses on partitional clustering of documents. In the future, it would also be possible to apply the proposed criterion functions for hierarchical clustering algorithms.

E. **Structural and Textual Feature Extraction for Semi-structured Document**

Paper [32] addresses XML document classification by considering both structural and content-based features of the documents. This approach leads to better constructing a set of informative feature vectors that represents both structural and textual aspects of XML documents. Author proposed a new framework for document classification, in particular XML document classification, based on extracting features from different aspects of the document. Extracted a feature vector that represents the document in a compact format; it captures valuable information that can be used later on to build an accurate classifier using any of the well known classification techniques. It also proposed a fast, robust, accurate, and novel approach for content-based document classification. Our proposed model is easy to implement for real world applications.

F. **Correlation Similarity Measure**

Paper [33] presents a new document clustering method based on correlation preserving indexing. It simultaneously maximizes the correlation between the documents in the local patches and minimizes the correlation between the documents outside these patches. Consequently, a low dimensional semantic subspace is derived where the documents corresponding to the same semantics are close to each other. Extensive experiments on NG20, Reuters, and OHSUMED corpora show that the proposed CPI method outperforms other classical clustering methods. Furthermore, the CPI method has good generalization capability and thus it can effectively deal with data with very large size.

Proposed Work

We focus our work on how to combine the advantages of two document models in document clustering. As a result of our work, a phrase-based document similarity is presented in this paper. By mapping each node of a suffix tree (excludes the root node) into a unique dimension of an M-dimensional term space (M is the total number of nodes except the root node), each document is represented by a feature vector of M nodes. Consequently, we find a simple way to compute the document similarity: First, the weight (tf-idf) of each node is recorded in building the suffix tree, and then the cosine similarity measure is used to compute the pair wise similarities of documents.

The document similarities are mainly decided by the overlap nodes (phrases), which appear in at least two different documents in the document set. All other nodes or phrases are trivial to the phrase-based document similarities, with a slight effect on the overall quality of document clustering.

A. **The Phrase-Based Document Similarity**

Throughout this paper, we use the symbols N, M, and k to denote the number of documents, the number of terms, and the number of clusters, respectively. We use the symbol D to denote the document set of N documents that we want to cluster, the C1, C2, . . . ,Ck to denote each one of the k clusters. In text-based information retrieval, a document model is a concept that describes how a set of meaningful features is extracted from a document. Most of the current document clustering methods uses the VSD model to represent documents. In the model, each document d is considered to be a vector in the M-dimensional term space. In particular, we usually employ the term tf-idf weighting scheme [4], [16], in which each document can be represented as

$$\vec{d} = \{w(1,d), w(2,d), \ldots, w(M,d)\}$$

Where $w(i,d) = (1 + \log tf(i,d)) \cdot \log(1 + N/df(i))$, $tf(i,d)$ the frequency of the ith is term in the document d, and is the number of documents containing the ith term.
In the VSD model, the cosine similarity is the most commonly used measure to compute the pair wise similarity of two document di and dj, which is defined as

$$sim_{i,j} = \frac{\vec{d_i} \bullet \vec{d_j}}{|\vec{d_i}| \times |\vec{d_j}|}.$$

B. **Suffix Tree Document Model**

The STD model considers a document d as a string consisting of words w1w2 . . .wm, not characters. The suffix tree of a document d is a compact trie containing all suffix substrings of the document d. Fig. 2 is an example of a suffix tree composed from three documents. The nodes of the suffix tree are drawn in circles. There are three kinds of nodes in the suffix tree: the root node, internal nodes, and leaf nodes. Each internal node has at least two children. Each edge is labeled with a nonempty substring of a document called a phrase. Then, each leaf node in the suffix tree designates a suffix substring of a document; each internal node represents a common phrase shared by at least two suffix substrings. The similarity of two documents is defined as the more internal nodes shared by the two documents, the more similar the documents tend to be.

In Fig. 2, each internal node is attached to an individual box. The numbers in the box designate the documents that have traversed the corresponding node. Each upper number designates a document identifier, the number below designates the traversed times of the document. (In the implementation of our approach, the node data structure has a list storing the numbers directly.)

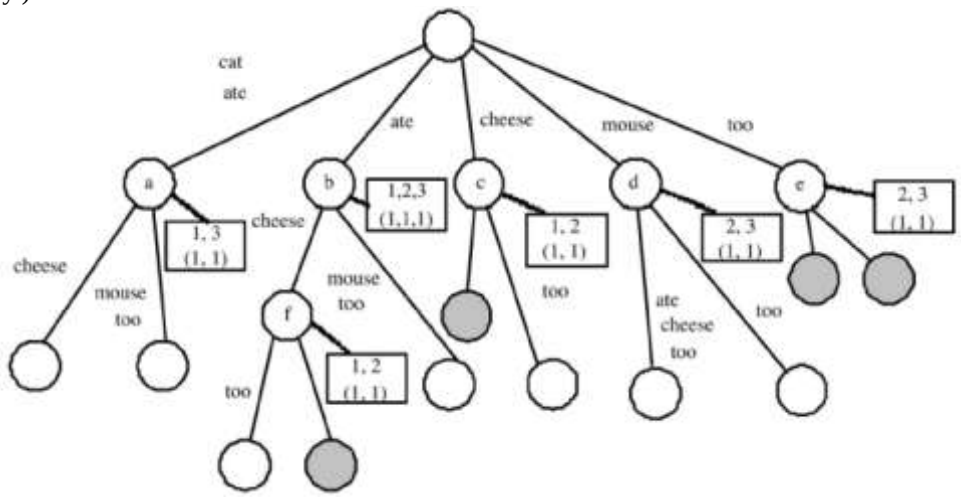

Fig. 2. The suffix tree of tree documents "cat ate cheese," "mouse ate cheese too," and "cat ate mouse too."

C. The Phrase-Based Document Similarity Based on the STD Model

As mentioned in the previous section, there are three different kinds of nodes in a suffix tree. In particular, there exist some leaf nodes labeled with an empty phrase (usually a NULL in the suffix tree implementations). They are generated by Ukkonen's algorithm [34], which is used to build suffix tree, and denote the end of the corresponding documents. For example, in Fig. 2, the four leaf nodes represented by gray circles are such nodes. We call these leaf nodes "terminal nodes" in our work. With the exception of the terminal nodes and the root node, each node in the suffix tree, either an internal node or a leaf node represents a nonempty phrase that appears in at least one document in the data set. The same phrase might occur in different edges of the suffix tree. For instance, there are three different edges labeled with the same phrase of "cheese" in the suffix tree of Fig. 2. The definition of the phrase-based document similarity is simple and understandable: By mapping each node v labeled by a nonempty phrase into a feature of M-dimensional term space, each document d can be represented as a feature vector of the weights of M node terms in the VSD model as illustrated by (1). It is very easy to understand that the document frequency of each node df(v) is the number of the different documents that have traversed node v; the term frequency tf(v,d) of a node v with respect to document d is the total traversed times of the document d through node v. In the example of Fig. 2, the df of node b is df(b)=3, the tf of the node with respect to the document 1 is tf(b,1)=1 (assuming the document identifiers of three documents to be 1, 2, 3). Therefore, we can calculate the weight of node b with respect to document 1 as . After obtaining the term weights of all nodes, it is easy to apply traditional similarity measures such as the cosine similarity to compute the similarity of any two documents. In this paper, the cosine similarity measure is used to compute the pairwise similarities of all documents. Let vectors, and denote two documents dx and dy, where xi and yi are the weights of corresponding node term vi, respectively. Then, the similarity of two documents is calculated by cosine similarity.

D. Affinity clustering based on similarity

Data mining, or exemplars, is traditionally found by randomly choosing an initial subset of data points and then iteratively refining it, but this only works well if that initial choice is close to a good solution. Affinity propagation [35] is a new algorithm that takes as input measures of similarity between pairs of data points and simultaneously considers all data points as potential exemplars. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. hence affinity propagation to solve a variety of clustering problems and found that it uniformly found clusters with much lower error than those found by other methods, and it did so in less than one-hundredth the amount of time. Because of its simplicity, general applicability, and performance, we believe affinity propagation will prove to be of broad value in science and engineering.

Clustering data by identifying a subset of representative examples is important for processing data clustering and detecting patterns in data [36]. Such "exemplars" can be found by randomly choosing an initial subset of data points and then iteratively refining it, but this works well only if that initial choice is close to a good solution. We devised a method called "affinity propagation," which takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a high-quality set of exemplars and corresponding clusters gradually emerges. We used affinity propagation to cluster images of faces, detect genes in microarray data, identify representative sentences in this manuscript, and identify cities that are efficiently accessed by airline travel. Affinity propagation found clusters with much lower error than other methods, and it did so in less than one-hundredth the amount of time.

E. **Proposed Algorithm**

Input: Dataset files
Output: Clustered group of files IDs.

Clustering Process
Step 1: Dataset Preprocessing
Stemming
Stop word removal
Frequent word removal
Step 2: All Dataset one Matrix conversion where every row represents each document.
Step 3: Unique words list creation from Dataset matrix.
Step 4: Generalized Suffix tree creation for dataset matrix for all data files content based phrase suffix tree.
Step 5: TF and IDF extraction from generalized suffix tree for all unique keywords.
Step 6: Document vector creation using TF and IDF find by suffix tree.
Step 7: Similarity matrix generation from similarity matrix generated by step 6.
Step 8: Similarity matrix is passed to affinity propagation for efficient clustering of documents.
Step 9: Step 8 Generates cluster grouped of documents IDs.

Conclusion

In this survey we had projected various clustering approaches and algorithms in document clustering .The area of document clustering have many issues, which need to be solved. We hope, the paper gives interested readers a broad overview of the existing techniques. As a future work, improvement over the existing systems with better results which offer new information representation capabilities with different techniques like search result clustering, collection clustering and co-clustering can be attempted

References

[1]. Crabtree, D., Gao, X., Andreae, P.: "Improving web clustering by cluster selection". In: Proceedings of 2005 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 172–178 (2005)
[2]. Muhammad Rafi, Mehdi Maujood ,Murtaza Munawar Fazal, Syed Muhammad Ali, "A comparison of two suffix tree-based document clustering algorithms", IEEE, 2010.
[3]. Hammouda, K., Kamel, M.: "Efficient document indexing for web document clustering". IEEE Transactions on Knowledge and Data Engineering 16(10), 1279–1296 (2004)
[4]. Hammouda, K., Kamel, M.: "Phrase-based document similarity based on an index graph model". In: Proceedings of 2002 IEEE International Conference on Data Mining ICDM, pp. 203–210 (2002)
[5]. Chim, H., Deng, X.: "A new suffix tree similarity measure for document clustering". In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 121–130. ACM, New York (2007)
[6]. Chim, H., Deng, X.: "Efficient phrase-based document similarity for clustering". IEEE Transactions on Knowledge and Data Engineering 20(9), 1217–1229 (2008)
[7]. Huang, A.: "Similarity measures for text document clustering", pp. 49–56 (2008)
[8]. Joydeep, A.S., Strehl, E., Ghosh, J., Mooney, R.: "Impact of similarity measures on web-page clustering". In: Workshop on Artificial Intelligence for Web Search, AAAI, pp. 58–64 (2000)
[9]. Janruang, J., Guha, S.: Semantic suffix tree clustering. In: First IRAST International Conference on Data Engineering and Internet Technology, DEIT (2011)
[10]. Carpineto, C., Osinski, S., Romano, G., Weiss, D.: A survey of web clustering engines. ACM Computing Surveys 41, 1–38 (2009)
[11]. Zamir, O., Etzioni, O.: Grouper: A dynamic clustering interface to web search results. In: Proceedings of the Eighth International World Wide Web Conference, pp. 283–296. Elsevier, Toronto (1999)
[12]. Wang, J., Li, R.: A New Cluster Merging Algorithm of Suffix Tree Clustering. In: Shi, Z., Shimohara, K., Feng, D. (eds.) Intelligent Information Proceesing III. IFIP AICT, vol. 228, pp. 197–203. Springer, Boston (2006).
[13]. Zhang, D., Dong, Y.: Semantic, Hierarchical, online Clustering of Web Search Results. In: Yu, J.X., Lin, X., Lu, H., Zhang, Y. (eds.) APWeb 2004. LNCS, vol. 3007, pp. 69–78. Springer, Heidelberg (2004)
[14]. Osinski, S., Weiss, D.: A concept-driven algorithm for clustering search results. IEEE Intelligent Systems 20, 48–54 (2005)
[15]. Sameh, A.: Semantic web search results clustering using lingo and wordnet. International Journal of Research and Reviews in Computer Science (IJRRCS) 1(2) (June 2010)
[16]. Crabtree, D., Andreae, P., Gao, X.: Query directed web page clustering. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006, pp. 202–210. IEEE Computer Society, Washington, DC, USA (2006)
[17]. Kale, A., Bharambe, U., SashiKumar, M.: A new suffix tree similarity measure and labeling for web search results clustering. In: 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET), pp. 856 –861 (2009)
[18]. Wu, J., Wang, Z.: Search results clustering in chinese context based on a new suffix tree. In: Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops, pp. 110–115. IEEE Computer Society, Washington, DC, USA (2008)
[19]. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 14, pp. 849–856. MIT Press (2001)

[20]. Beil, F., Ester, M., Xu, X.: Frequent term-based text clustering. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 436–442. ACM, New York (2002)

[21]. Cilibrasi, R., Vitanyi, P.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007)

[22]. Kjos-hanssen, B., Evangelista, A.J.: Google distance between words. Computing Research Repository abs/0901.4 (2009).

[23]. Miyamoto, S. ; Dept. of Risk Eng., Univ. of Tsukuba, Tsukuba, Japan ; Terami, A., "Semi-supervised agglomerative hierarchical clustering algorithms with pairwise constraints", International Conference on Fuzzy Systems (FUZZ), Pages 1 – 6, IEEE, 2010.

[24]. Singh, V.K. ; Tiwari, N. ; Garg, S., "Document Clustering Using K-Means, Heuristic K-Means and Fuzzy C-Means", International Conference on Computational Intelligence and Communication Networks (CICN), Page(s): 297 – 301, IEEE, 2011

[25]. Jongkol Janruang, Guha, S., "Applying Semantic Suffix Net to suffix tree clustering", 3rd Conference on Data Mining and Optimization (DMO), Pages 146 – 152, IEEE, 2011.

[26]. Dell Zhang, Yisheng Dong, "Semantic, Hierarchical, Online Clustering of Web Search Results", Advanced Web Technologies and Applications, Pages 69-78, Springer, 2004.

[27]. Xiuqin Lin ; Qianhao Zhang ; Gengyu Wei, "The clustering algorithm for Chinese texts based on Lingo" Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Volume: 2, Pages 1187–1190, IEEE, 2011.

[28]. Ajitha Annadurai, "Architecture of Personalized Web Search Engine Using Suffix Tree Clustering", Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011), Page 604-608, IEEE, 2011.

[29]. Neeraj Sahu, G. S. Thakur, "Hesitant Distance Similarity Measures for Document Clustering" IEEE, 2011.

[30]. Phiradit Worawitphinyo, Xiaoying Gao, and Shahida Jabeen, "Improving Suffix Tree Clustering with New Ranking and Similarity Measures", ADMA 2011, Part II, LNAI 7121, pp. 55–68, Springer, 2011.

[31]. Duc Thang Nguyen, Lihui Chen and Chee Keong, "Clustering with Multiviewpoint-Based Similarity Measure",IEEE Transactions On Knowledge and Data Engineering, VOL. 24, NO. 6, Pages 988-1001, IEEE, 2012.

[32]. Mohammad Khabbaz, Keivan Kianmehr, and Reda Alhajj, "Employing Structural and Textual Feature Extraction for Semi-structured Document Classification", IEEE Transactions on Systems, Man, and Cybernetics, VOL. 42, NO. 6, Page 1566-1578, IEEE, 2012.

[33]. Taiping Zhang, Yuan Yan Tang, Bin Fang and Yong Xiang, "Document Clustering in Correlation Similarity Measure Space", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, NO. 6,Page 1002-1013, IEEE 2012.

[34]. E. Ukkonen, "On-Line Construction of Suffix Trees", Algorithmica, vol. 14, no. 3, pp. 249-260, 1995.

[35]. http://www.psi.toronto.edu/index.php?q=affinitypropagation.

[36]. Hung Chim and Xiaotie Deng, "Efficient Phrase-Based Document Similarity for Clustering", IEEE Transactions on knowledge and data engineering, vol. 20, no. 9, pp 1217-1229, 2008 IEEE.