ISSN:0975-9646

Generalized Tree Based Document Cluster Using Hybrid Similarity

Gaurav Dwivedi
Student, M.Tech
CSE, VNSIT Bhopal

Prof. Amit Kumar Nandanwar
Computer Science and
Engineering, VNSIT Bhopal

Abstract-the Web has undergone a tremendous growth regarding both content and users. This has lead to an information overload problem in which people are finding it increasingly difficult to locate the right information at the right time. Recommender systems have been developed to address this problem, by guiding users through the big ocean of information. Until now, recommender systems have been extensively used within e-commerce and communities where items like movies, music and articles are recommended. More recently, recommender systems have been deployed in online music players, recommending music that the users probably will like.

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. Automatic document clustering has played an important role in many fields like information retrieval, data mining, etc. The aim of this thesis is to improve the efficiency and accuracy of document clustering.

All documents and data are in digital form reason of easy maintaining, faster access and compact storage. To access relative document easily document clustering is used. Document clustering creates segments collection of textual documents into subgroups using similar contents. The purpose of document clustering is to meet human interests in information searching and understanding. An effective feature phrase of document is more informative feature for improving document clustering.

Keywords Suffix tree, Similarity Measure, Document Clustering, Feature Extraction.

1.1 INTRODUCTION

In this paper Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. In other words, the goal of a good document clustering scheme is to minimize intra-cluster distances between documents, while maximizing inter-cluster distances (using an appropriate distance measure between documents). A distance measure (or, dually, similarity measure) thus lies at the heart of document clustering. Clustering is the most common form of unsupervised learning and this is the major difference between clustering and classification. No super-vision means that there is no human expert who has assigned documents to classes. In clustering, it is the distribution and makeup of the data that will determine cluster membership. Clustering is sometimes erroneously referred to as automatic classification; however, this is inaccurate, since the clusters found are not known prior to processing whereas in case of classification the classes are pre-defined. In clustering, it is the distribution and the nature of data that will determine cluster membership, in opposition to the classification where the classifier learns the association between objects and classes from a so called training set, i.e. a set of data correctly labeled by hand, and then replicates the learnt behavior on unlabeled data.

Document clustering is particularly useful in many applications such as automatic categorization of documents, grouping search engine results, building taxonomy of documents, and others. For this Hierarchical Clustering method provides a better improvement in achieving the result. Our paper presents two key parts of successful Hierarchical document clustering. The first part is a document index model, the Document Index Graph, which allows for incremental construction of the index of the document set with an emphasis on efficiency, rather than relying on single-term indexes only. It provides efficient phrase matching that is used to judge the similarity between documents. This model is flexible in that it could revert to a compact representation of the vector space model if we choose not to index phrases. The second part is an incremental document clustering algorithm based on maximizing the tightness of clusters by carefully watching the pair-wise document similarity distribution inside clusters.

1.2 Document Clustering

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.

1.3 Suffix Tree Clustering (STC)

a novel clustering algorithm designed to meet the requirements of post-retrieval clustering of Web search results. STC is unique in treating a document as a string, not simply a set of words, thus making use of proximity information between words. STC relies on a suffix tree to efficiently identify sets of documents that share common phrases and uses this information to create clusters and to succinctly summarize their contents for users. First, we describe the desired characteristics for a post-retrieval clustering algorithm and motivate the STC algorithm. We then describe the suffix tree data structure: its definition, characteristics and construction algorithms. Next, we describe the STC algorithm and its complexity. Finally, we detail some of the characteristics of STC.

the suffix tree of the string "I know you know I know". Internal nodes are marked as circles, leaves as rectangles. There are six leaves in this example, numbered from 1 to 6. The terminating character is not shown in this diagram.

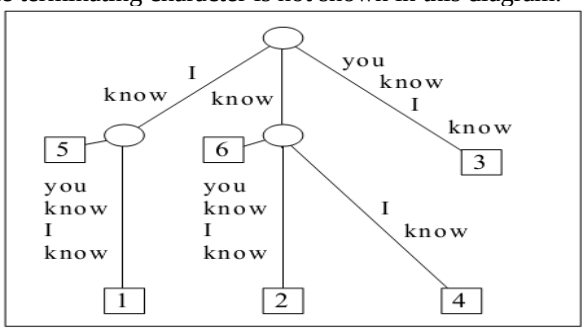

Figure 1.1: Example of a suffix tree

The suffix tree of the string "I know you know I know". There are six leaves in this example, marked as rectangles and numbered from 1 to 6. The terminating characters are not shown in this diagram. In a similar manner, a suffix tree of a set of strings, called a generalizedsuffix tree, is a compact trie of all the suffixes of all the strings in the set:

A generalizedsuffix tree T for a set S of n strings Sn, each of length mn, is a rooted directed tree with exactly 6mnleaves marked by a two number tuple (k,l) where k ranges from 1 to n and l ranges from 1 to mk. Each internal node, other than the root, has at least two children and each edge is labeled with a nonempty sub-string of words of a string in S. No two edges out of a node can have edge labels beginning with the same word. For any leaf (i,j), the concatenation of the edge labels on the path from the root to leaf (i.j) exactly spells out the suffix of Si that starts at position j, that is it spells out Si [j..mi].

Figure 1.2 is an example of the generalized suffix tree of a set of three strings – "cat ate cheese", "mouse ate cheese too"and "cat ate mouse too". The internal nodes of the suffix tree are drawn as circles, and are labeled athrough ffor further reference. Leaves are drawn as rectangles. The first number in each rectangle indicates the string from which that suffix originated; the second number represents the position in that string where the suffix starts. Each string is considered as having a unique terminating character, which is not shown in this diagram.

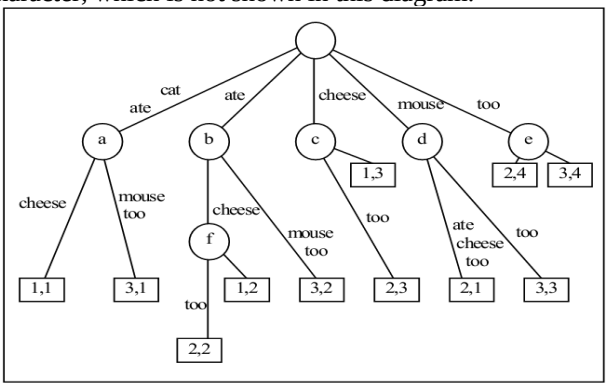

Figure 1.2: Example of a generalized suffix tree

The generalized suffix tree of the three strings "cat ate cheese", "mouse ate cheese too "and " cat ate mouse too". The internal nodes of the suffix tree are drawn as circles, and are labeled a through f for further reference. There are 11 leaves in this example (the sum of the lengths of all the strings) drawn as rectangles. The first number in each rectangle indicates the string from which that suffix originated; the second number represents the position in that string where the suffix starts.

2. PREVIOUS WORK

Periodic pattern mining or periodicity detection has a number of applications, such as prediction, forecasting, detection of unusual activities, etc. The problem is not trivial because the data to be analyzed are mostly noisy and different periodicity types (namely symbol, sequence, and segment) are to be investigated. Accordingly, we argue that there is a need for a comprehensive approach capable of analyzing the whole time series or in a subsection of it to effectively handle different types of noise (to a certain degree) and at the same time is able to detect different types of periodic patterns; combining these under one umbrella is by itself a challenge.

All clustering methods have to assume some cluster relationship among the data objects that they are applied on. Similarity between a pair of objects can be defined either explicitly or implicitly. we introduce a novel multiviewpoint-based similarity measure and two related clustering methods. The major difference between a traditional dissimilarity/similarity measure and ours is that the former uses only a single viewpoint, which is the origin, while the latter utilizes many different viewpoints, which are objects assumed to not be in the same cluster with the two objects being measured. Using multiple viewpoints, more informative assessment of similarity could be achieved

3. PROPOSED ALGORITHM

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 pairwise similarities of documents.

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.

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. 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.

4. RESULT ANALYSIS

Table 5.1. Performance Comparison of Different Clustering Methods Using OHSUMED Data Corpus

Cluster No.	Latent semantic indexing (LSI)		Locality Preserving Indexing (LPI)		Correlation Preserving Indexing (CPI)		Affinity based STC	
	Avg Acc (%)	+/- (%)	Avg Acc (%)	+/- (%)	Avg Acc (%)	+/- (%)	Avg Acc (%)	+/- (%)
1	67.87	12.2	71.12	12.7	75.06	13.3	79.57	10.61
2	54.53	10.1	58.65	9.92	60.84	9.52	64.32	9.5
3	47.96	8.32	52.57	7.86	56.2	8.88	61.39	6.22
4	42.72	6.93	44.45	7.62	47.45	7.02	58.27	6.71
5	38.78	6.08	40.78	6.55	44.58	6.6	57.63	7.06
6	36.72	4.61	39.81	5.41	41.59	5.19	58.08	6.31
7	35.66	4.61	37.64	4.63	39.77	4.77	50.61	5.37
Avg	46.32	7.54	49.29	7.81	52.21	7.90	61.41	7.40

Table 4.2. Performance Comparison of Different Clustering Methods Using 20News Group Dataset Corpus

	Kmeans		Spectral		Locality preserving indexing		correlation preserving indexing		Affinity based STC	
	Avg Acc (%)	+/- (%)	Avg Acc (%)	+/- (%)	Avg Acc (%)	+/- (%)	Avg Acc (%)	+/- (%)	Avg Acc (%)	+/- (%)
NG1/NG2	74.14	16.1	92.8	3.93	94.56	4.56	97.08	4.41	98.43	1.57
NG2/NG3	63.92	11.49	78.7	9.38	80.18	13.1	83.03	11.4	86.75	13.25
NG8/NG9	67.78	14.18	78.2	10.23	81.06	11.89	86.18	12.07	90.29	9.71
NG10/NG11	64.5	10.87	68.7	8.78	74.86	12.87	76.38	12.36	84.37	16.67
NG1/NG15	73.56	13.51	90.1	5.98	91.56	7.56	95.18	7.07	96.65	3.35
NG18/NG19	65.38	9.57	71	10.32	75.94	12.78	81.42	12.01	84.81	15.19

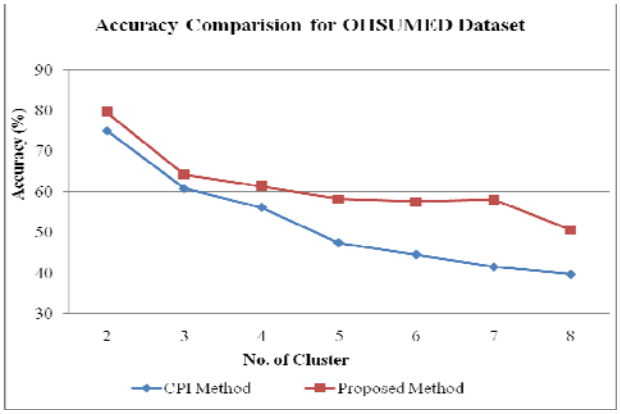

Figure 4.1. The average accuracy comparison over 2 to 8 clusters

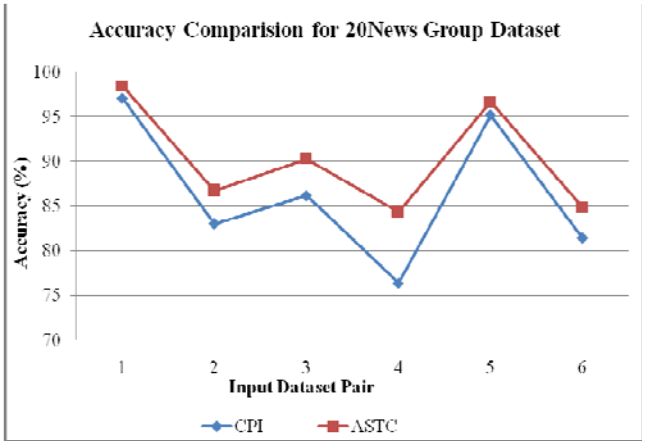

Figure 4.2. The average accuracy comparison using 20NewsGroup dataset.

5.CONCLUSION

In this paper proposed suffix tree data structure is used for identify phrases within documents. Also consider that there are other competent ways to recognize and take out phrases from the documents. In really, the phrases in documents are not dependent to the phrase withdrawal techniques and tools. For the primary instant, vectors of phrases tf-idf weights are utilized for performed document similarities and are confirmed to be very successful in clustering documents. This work has presented a well approach to expand the practice of tf-idf weighting scheme: the term tf-idf weighting method is proper for estimating the importance of not only the keywords however also the phrases of document for document clustering purpose.

The model of the suffix tree may be new for document similarity and relatively simple, but the execution is much complex. To get better performance for the phrase-based document similarity, this work examines both the hypothetical data structure investigation and also the clustering approaches optimization by using affinity propagation clustering technique. Hence results for proposed method are effectively improving the performance on compare to existing techniques such as CPI based method. These experiments are proven that for large datasets. The phrase-based document similarity is a highly accurate and efficient practical document clustering solution.

REFERANCE

[1] Faraz Rasheed, Mohammed Alshalalfa, and Reda Alhaj, "Efficient Periodicity Mining in Time Series Databases Using Suffix Trees", IEEE Transactions On Knowledge And Data Engineering, Vol. 23, No. 1, Pages: 79 – 94, IEEE, 2011.

[2] Duc Thang Nguyen, Lihui Chen, and Chee Keong Chan, "Clustering with Multiviewpoint-Based Similarity Measure", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 6, Pages: 988 - 1001, June 2012.

[3] Luís Filipe da Cruz Nassif and Eduardo Raul Hruschka, "Document Clustering for Forensic Analysis: An Approach for Improving Computer Inspection", IEEE Transactions on Information Forensics and Security, Vol. 8, No. 1, Pages: 46 - 54, January 2013

[4] Mohammad Khabbaz, Keivan Kianmehr, and Reda Alhajj, "Employing Structural and Textual Feature Extraction for Semistructured Document Classification", IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications And Reviews, Vol. 42, No. 6, Pages: 1566 – 1578, November 2012.

[5] Taiping Zhang, Yuan Yan Tang, Bin Fang, "Document Clustering in Correlation Similarity Measure Space", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 6, Pages: 1002 - 1013, June 2012.

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

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

[8] Sahu Neeraj, Thakur G. S., "Hesitant Distance Similarity Measures for Document Clustering", World Congress on Information and Communication Technologies, Pages: 430 – 438, IEEE, 2011.

[9] Rafi Muhammad, Maujood Mehdi, Fazal Murtaza Munawar, Ali Syed Muhammad, "A comparison of two suffix tree-based document clustering algorithms", Pages: 244-249, IEEE, 2010.

[10] Silverstein C. and Pedersen J. O. "Almost-constant time clustering of arbitrary corpus subsets". In Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97), Pages: 60-66, 1997.

[11] Hearst M. A, "The use of categories and clusters in information access interfaces". Natural Language Information Retrieval, Kluwer Academic Publishers, Pages: 333-374, 1998.

[12] Weiner, P., "Linear pattern matching algorithms". In Proceedings of the 14th Annual Symposium on Foundations of Computer Science (FOCS'73), Pages: 1-11, 1973.

[13] Landau G. M. and Vishkin U. "Fast Parallel and serial approximate string matching". Journal of Algorithms, Vol 10, No. 1, Pages: 57-169, 1989.

[14] Ehrenfeucht A. and Haussler D., "A new distance metric on strings computable in linear time", Discrete Applied Math, Vol. 40, Pages: 191–203, 1988.

[15] Rodeh M., Pratt V. R. and Even, S., "Linear algorithm for data compression via string matching". Journal of the ACM, 28(1): Pages: 16-24, 1981.