Affinity Propagation Based Document Clustering Using Suffix Tree

Aditi Chaturvedi
CSE Dept.
OCT, Bhopal, India

Dr. Kavita Burse
CSE Dept.
OCT, Bhopal, India

Rachna Mishra
CSE Dept.
OCT, Bhopal, India

Abstract

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. This paper proposes phrase content based document similarities based on suffix tree. Content based similarity used term frequency (tf) and inverse document frequency (idf) weighting method to compute similarity of documents. Proposed affinity propagation clustering approach is very effective on clustering the documents of standard document OHSUMED dataset by comparison with existing document clustering methods.

1. Introduction

Document clustering has been deliberated as a post recovery document visualization method to provide a spontaneous navigation and browsing mechanism by arranging documents in the form of sets, where each set represents a dissimilar topic. The four concepts on which clustering technique are based are namely similarity measure, clustering model clustering algorithm and data representation model. Now a day's vector space document (VSD) model is mostly used as document clustering method. The general outline of this data model starts with a depiction of any document as a feature vector of the words that come out in the documents of a data set. The dissimilar word coming out in the documents is regularly measured to be an atomic attribute term in the VSD model, because to symbolize semantic concepts words are the basic units in most ordinary languages (including English) to. Terms of documents are important feature for clustering which is used by term weights normally called tf-idf, tf stands for term frequency and idf stands for document inverse frequencies. Any documents can be checked for similarity using any feature from several features, such as cosine similarity measure, Jaccarrd measure and euclidian distance.

2. RELATED WORK

Lots of work have been done in field of documents clustering and applied for many search engines. The main factors involve for document clustering that impact on quality also performance of clustering. First one is Data representation, second similarity measure and clustering techniques that perform clustering using similarity measure matrix [3-6]. Many clustering techniques used vector space model (VSM) for data representation scheme. These models consider a document as each word represents one dimension in multi dimension words representation. Every word has weight by calculating TF and IDF for particular word. Similarities of two documents can be calculates by such as cosine similarity, pearson correlation coefficient or jaccard coefficient etc. [7, 8].

Main drawback of this method is that it measure similarity for documents only single word of documents so lots of important information has been ignored [4]. Apart this these clustering technique for VSM also do not support incremental processing [9]. Web clustering engines survey improves incremental processing efficiently by implement clustering techniques [10]. The main work that uses the information about equal of words and phrase based analysis in an incremental method is Suffix Tree clustering (STC) [11].

STC is an efficient algorithm which perform processing within linear time O(n) [18] for clustering the documents allotment phrases or the suffix of a phrase into single cluster. The Suffix tree is constructed for group of documents to identify various phrases and their all possible suffixes. A Suffix Tree data structure is widely used in various applications [12]. A suffix Tree is a shot memory occupied tree data Structure that considering all the suffixes of a text phrase [12], here phrases represent sequence of word other than grammar of sentence [10]. Main feature of STC is uses phrase efficiently rather than just considering just word which effect clustering for real world. Because that any document can have lots of topics thus a document can belongs to several clusters [1].

A variety of work have done to get better organization of Suffix Tree based clustering by recognized the harms in the unusual STC. A good work has proposed the Semantic Suffix Tree Clustering (SSTC) [9], by merge the semantic similarity, of the suffix Tree. This technique finds similarity with help of Wordnet and used it with the string similar in the construction of suffix Tree. The objective to find similar phrase but different words, such as, the context of following phrases “doctor and nurse” and “physician and nanny” does not contain common words but they are same in meaning, Suffix Tree recognized these words differently. One more extension of STC is the Semantic Hierarchical Online Clustering (SHOC) scheme is given in paper [13], this method applied Suffix tree algorithm on array of frequent phrases for clustering purpose. Lingo algorithm [14] is also good addition to the SHOC algorithm this merge phrase with latent semantic analysis for categorize cluster for efficient searching apply. Lingo is expanded to Lingo, the Semantic Lingo algorithm [15] this increases cluster by using synonym within results snippets. NTSC proposed algorithm is introduced to solve the problem of large clusters with poor quality in STC. NTSC consist advantage of vector space model within Suffix Tree to calculate similarity between pair wise documents it helps to make effective clusters to worth of large size clusters [5, 6]. Another algorithm proposed for cluster merging that usages both the cosine similarity with the non-overlie parts of the clusters to take into consideration the similarity between the non overlapping parts of the clusters [12]. STC has been making improved by effectively find relation between user query and clusters. This method discover cluster description similarity with not only cluster overlie to overcome cluster chaining problem. As mention in [16], this helps to maintain cluster performance.

Apart of this correlation preserving indexing method is presented in paper [17]. This method maximizes correlation between documents in local patch with and minimizing correlation between documents these patches. A low dimensional semantic subspace is resultant where the documents parallel to the same semantics are close to each other. Broad simulation has been done using this method on various datasets such as NG20 and OHSUMED corpora. These experiments are proving the proposed CPI method outperforms with comparison to other traditional methods. Also CPI method has ability and hence it can effectively deal with data is very large size.

3. PROPOSED WORK

This work is focus on combining advantage of different document models for clustering purpose. Here proposes phrase based document clustering by link of each phrase with suffix tree in unique matrix dimension. This matrix having M-dimensional term space where M is represents total number of terms nodes of suffix tree other than root. Every document represented by vector of M nodes. With help of these vector calculates documents similarity. First of all weight for tf and idf is measure for each phrase term in suffix tree, hence every document is identified with these weight vectors. Then cosine similarity is applied using these weight vector is used to similarity for every pair of documents.

The document similarities are totally depends on the common nodes or phrases, which are common in any two different documents present in the total dataset. Other nodes or phrases are insignificant to the phrase-based document similarities, with a minor cause on the on the whole efficiency of clustering.

3.1 The Phrase-Based Document Similarity

This section discussed traditional similarity by using phrase; here symbols represents N denote the number of documents, M represents number of terms and k to the number of clusters. Here symbol D represents the overall document set of N documents that required to be cluster, the C1, C2, . . . , Ck where every symbol represents each one of the k clusters. In text-based information gathering, a document model is a used to express a set of important features is mined from a document set. Mainly of the existing clustering methods uses the VSD model to signify data. These models correspond to each document d is collection of a vector in the M-dimensional term space. These schemes normally uses term tf-idf weighting scheme [4], [16] where every document define as

D= {w(1,d), w(2,d),………….,w(M, d) eq. (1)

Where w(1,d)=(1+logtf(I, d).log(1+N/df(i)), tf(i, d) is ith the frequency of the ith is term in the document d, and is the number of documents containing the ith term.

With help of these vectors of cosine similarity is measured for every pair of documents present in dataset. Cosine similarity of two documents di and dj can be given as

$$\boldsymbol{Sim_{i,j}} = \frac{\vec{d_i}\cdot\vec{d_j}}{|\vec{d_i}|\times|\vec{d_j}|} \quad \text{eq. (2)}$$

3.2 Suffix Tree Document Model

The STD model generates using a document d as a text of words w1, w2 . . .wm, excepting characters. The suffix tree of a of a document d for dataset D is a dense tree joining all suffix substrings of the document d. Fig. 1 demonstrate an example of a generalized suffix tree by combining three documents. In the suffix tree nodes are represents within the circles. This tree have three types of nodes first is root node, intermediate nodes and in last of edge leaf node. Internal nodes are compulsory to have at least two child nodes. Every edge in tree joining nodes have phrases of those documents. Each leaf node leads to suffix substring of respective document d. every intermediate node must have two common suffix of documents join common content. Similarity of two documents can be drived from more intermediate nodes are common for these two documents. More common intermediate nodes mean documents have larger similarity.

In Fig. 1, every inside node is connected to an entity box. The numbers in the box assign the documents that have pass through the equivalent node. All higher digit allocate a document id, the digit lower allocate the pass through times of the document.

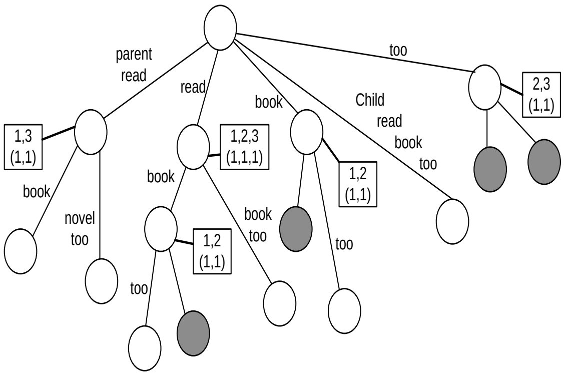

Figure 1. The suffix tree of tree documents "Parents read book. Child read book too. Parent read novel too."

3.3 The Phrase-Based Document Similarity Based on the Suffix tree

As given three types of nodes are there in suffix tree to join nodes interconnect. This suffix tree is generated by ukkonen's algorithm [18] for suffix tree generation. This basic algorithm generates suffix tree for given strings, string is consider as collection of characters. But here nodes are represent phrase of documents which is used to construct suffix tree, and represent the end of the relative documents. For example, in Fig. 1, the four leaf nodes correspond to by dark circles are such nodes. These nodes are known as leaf nodes or "terminal nodes" in proposed work. Each intermediate node apart to leaf nodes and root node in the suffix tree represents a not null string as phrase that consists of at least one document in the data set D. The common phrase may come about in dissimilar edges of the suffix tree.

For instance, present are three different edges labelled with the common phrase of "read" in the suffix tree given in Fig. 1. The meaning of the phrase-based document similarity is easy and clear: mapping of each node v labelled by a phrase string into a feature of M-dimensional term matrix, every document d may be correspond to as a feature vector of the weights of M node terms as demonstrate by eq. 1.

Document frequency of each node df(v) can be given as the how many of the diverse documents that have current node v; the term frequency tf(v,d) of a node v for relative document d is define as the total times of the document d appear while reaching root node from leaf through node v. In the example of given in figure, the df of node b is df(b)=3, the tf of the node with for the document 1 is tf(b,1)=1 where the document id of given documents are 1, 2, 3. Hence calculate the weight of node b as respect to document 1. When term weight has been obtained of every node, tradition similarity measure can be applied easily, such as the cosine similarity to derive the similarity of any two documents present in dataset D. In this paper, the cosine similarity measure is performed for pair wise similarity measure within documents for overall documents set.

3.4 Affinity clustering based on similarity

Data mining, or exemplars, is usually set up by randomly decided for a primary subset of data files and then by iterations refines selected points for best suited results. But this only works well if that initial choice is close to a good solution otherwise a lot time takes for good solution.

Novel algorithm proposed affinity propagation [19] that takes input measures of similarity among couple of data points and concurrently judges all data points as probable exemplars. Real-valued information's are switch among data points until a high-quality position of exemplars and equivalent clusters steadily appears. Therefore affinity propagation is good method for solving various clustering problems and generates that for consistently found clusters with lesser error than those establish by other methods. Since its easiness, broad applicability, and performance, considers affinity propagation will prove to be of wide value in science and engineering.

Clustering is important process for information retrieval and detecting patterns from data collections [20]. Such "exemplars" are selected by random manner choosing a preliminary subset of data files and then by repeating and selecting better alternative points, refining it to reach better final data point. But this method performs well only if that preliminary selection is near to a good solution. So author advised a method "affinity propagation," which takes as input measures of similarity among pairs of data. Messages are exchanged among data points until a high-quality set of exemplars and corresponding clusters gradually emerges.

3.5 Proposed Algorithm

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

```
Step 1: Dataset Pre-processing
        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.
Step 10: Calculate accuracy of cluster by eq. (3).
```

4. Simulations and Results

4.1 Evaluation Metrics

To evaluate proposed method with existing methods, the accuracy (AC) metric is calculated used to measure the clustering performance [9]. The AC metric can be given as below eq 3:

$$AC = \sum_{i=1}^{n} \delta(\mathrm{si}, \mathrm{map}(\mathrm{ri}))\,/n \quad \text{eq. (3)}$$

Where ri is the cluster label generated by proposed algorithm, si is the label make available by the body dataset providers, n is the total documents used for clustering purpose, δ(x, y) is the function that return mapping original label and generated label. This function return one if x=y and return zero otherwise, and map(ri) is the permutation mapping function that maps cluster label ri to the equivalent label from the data label by proposed method.

4.2 Document Representation

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] , in which each document can be represented as mention in eq. 1

D= {w(1,d), w(2,d),…………,w(M, d)

Where w(1,d)=(1+logtf(I, d).log(1+N/df(i)), tf(i, d) is the frequency of the ith term in the document d, and df(i) is the number of documents containing the ith term.

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 mention in eq. 2

$$\boldsymbol{Sim_{i,j}} = \frac{\vec{d_i}\cdot\vec{d_j}}{|\vec{d_i}|\times|\vec{d_j}|}$$

4.3 Results

Experiments were performed using Matlab 7.10.1 on OHSUMED data sets. We compared the proposed algorithm with other competing algorithms under same experimental setting. The experimental results obtained when number of cluster are set to from 2 to 8 for OHSUMED data sets, the number of nearest. In all experiments, proposed algorithm performs better than or competitively with other algorithms. The details of experiments can be described as follows.

The OHSUMED collection includes medical abstracts from the MeSH categories of the year 1991. In the literature [21], the first 20,000 documents were divided into two halves, 10,000 for training and 10,000 for testing. The specific task was to categorize the 23 cardiovascular diseases categories. In our experiment, we use this subset for document clustering. After removing those documents appearing in multiple categories, the testing data contains more than 14,000 documents with unique cluster labels.

The experiments were performed with the number of clusters ranging from 2 to 8. For each given c (i.e., the number of clusters), 50 document sets with different clusters were randomly chosen from the corpus. Hundred runs were performed for each set of documents. Table 1 shows the means and standard deviations. It can be seen from Table 1 that the average accuracy for suffix tree clustering, for all the cases of c=2;...;8, proposed algorithm consistently outperforms the other existing algorithms.

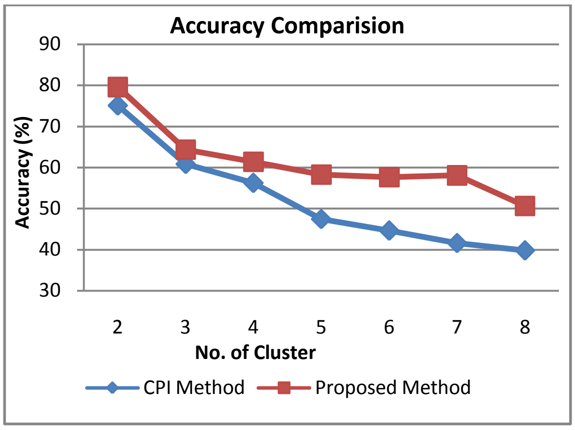

Figure 2. The average accuracy comparison over 2 to 8 clusters

Table 1. Performance Comparison of Different Clustering Methods Using OHSUMED Data Corpus

Cluster Number	CPI		Proposed Algo	
	Average Accuracy (%)	+/- (%)	Average Accuracy (%)	+/- (%)
2	75.06	13.3	79.57	10.61
3	60.84	9.52	64.32	9.5
4	56.2	8.88	61.39	6.22
5	47.45	7.02	58.27	6.71
6	44.58	6.6	57.63	7.06
7	41.59	5.19	58.08	6.31
8	39.77	4.77	50.61	5.37
Average	52.21	7.90	61.41	7.40

5. Conclusion

This work 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.

6. 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 Proceeing 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] 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.

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

[19] http://www.psi.toronto.edu/index.php?q=affinity propagation.

[20] 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.

[21] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," LS VIII-Report LS8-Report 23, Universitat Dortmund, 1997.

.