Improving Suffix Tree Clustering Algorithm for Web Documents

Yan Zhuang
Computer Center
East China Normal University
Shanghai, China
zhuangyan.ecnu@foxmail.com

Youguang Chen
Computer Center
East China Normal University
Shanghai, China
ygchen@cc.ecnu.edu.cn

Abstract—Web document clustering results can help users quickly locate the information they need among the results search engines returned. According to the characteristics of the suffix tree structure and the flaws of similarity calculation in STC algorithm's cluster merging, this paper proposes an improved suffix tree clustering method. The method combines vector space model with Pearson correlation coefficient, calculates the relevant of clusters based on document vector of all clusters, and then utilizes the relevant vectors of clusters and the correlations between them to calculate the similarity for cluster merging, improves the clustering process of documents. Analysis of the experimental results shows that the method outperforms the original STC algorithm on Web documents clustering.

Keywords-Web Document Clustering; Suffix Tree; Suffix Tree Clustering; Space Vector Model; Pearson Correlation Coefficient

I. INTRODUCTION

With the explosive growth of Internet information now, Web search engines have become an important tool for people to query information from the Internet, which provides a platform for people to share knowledge and exchange information, usually follow a certain way to sort all the relevant pages and show search results in the list form. This mechanism which sort results by relevance may only performs well in cases where users well describe query contents accurately and clearly. In reality, most users can not accurately describe their queries clearly, and they usually concentrate query contents into one or two search terms. Under these conditions of low quality queries, search tools can not define what pages or links users are really interested in. It provides users with return results that contain a large number of irrelevant or low correlations of pages and documents, so that the efficiency and accuracy of search are not high. Therefore, in the web information retrieval system, if web document clustering can be carried out effectively, the overall efficiency of the system performance and retrieval will be greatly improved.

To solve these problems, using clustering technology Web documents clustering, generating a representative tag of a word or a phrase for each cluster, and then presenting users with clustering results becomes a more effective solution. Clustering can reveal the internal structure of a Web document set, make similar documents associated together to form clusters which having a greater similarity between the members in each cluster and a smaller similarity between members of different clusters. To a certain extent, it solves the problem of messy online information, allows users to locate the needed information more conveniently and accurately. Automatically clustering search results into different clusters of simple theme or summary description, forming a hierarchy similar to the folder structure is a good solution.

To take full advantage of Web document structure information in the suffix tree clustering algorithm, this paper proposes an improved suffix tree clustering method. Compared with the traditional suffix tree clustering algorithm, the difference in this approach is that, the researchers cope with the document information contained in base clusters to obtain the correlation vector for each base cluster. In the selection and merging process of basic clusters, by utilizing the structural information between the basic clusters in correlation vectors, it considers not only the document overlap between two basic clusters, but also the relevance of other basic clusters. Experiments show that the improved STC algorithm has a better performance than original STC.

II. RELATED WORK

Weiner[1] first proposed the suffix tree algorithm for linear time tree building. On this basis, McCreight[2] proposed a more space-saving algorithms. Ukkonen [3] proposed an improved algorithm for linear tree building, and it was easy to understand and has online features. Gusfield [4] elaborated Ukkonen's suffix tree algorithm. Improved algorithm for suffix tree achievements, making STC algorithm widely used. Zamir[5,6] first proposed a suffix tree clustering algorithm used to search engine search results clustering. By identifying phrase shared between different documents on text clustering, STC algorithm is based on suffix tree model of document set, Compared to the vector space model, which takes into account the near sequential relationship between words, resulting in a better clustering effect.

At this stage, related work of text clustering and suffix tree clustering has a number of extensive and in-depth researches. Wang[7] proposed an approach with significantly lower memory requirements. Wu[8] presented a new STC algorithm which was more suitable for Chinese context. Worawitphinyo's paper[9] introduced a new method for ranking base clusters and new similarity measures for comparing clusters which improved the cluster merging process. Moe[10] modified original STC with improvements in effectiveness and efficiency by

applying a more sophisticated similarity measure. Janruang[11] introduced semantic similarity in clustering process, adopted a heuristic methods to clustering process, reduced suffix tree nodes and branches. Rafi[12] compared two approaches of extracting text phrases, discussed the representation model and similarity calculation method. Janruang[13] proposed a new algorithm Semantic Suffix Tree Clustering (SSTC) to cluster web search results containing semantic similarities.

III. SUFFIX TREE MODEL

Suffix tree is a valid string matching tree hierarchical data structure, is widely used in basic string handling problems. Such as finding the greatest overlap substring matching, similar string, string compression and text compression. STC regard documents as a set of phrases sequence rather than a collection of random words, this method is able to retain the semantic order of words, it can easily select phrases as theme tags for clustering result clusters, and with clear semantic and outstanding readability. While most traditional clustering algorithms regard documents as unordered word sets, ignoring the original order between phrases and sentences, therefore they lost valuable sequence information which contains explicit semantics, making the identification of tags for each category difficult. Generalized suffix tree is an extension of the suffix tree, in the text clustering process, processing unit of generalized suffix tree is a word rather than a character, and corresponding string of words' series is called a phrase. Here are two definitions:

Definition 1. Suffix Tree: A suffix tree T of a string S which contains m words has exactly m leaf nodes, each non-leaf node has at least two child nodes, each edge has a non-empty string S. Two edges derived from one node can't contain the same prefix substring. The string concatenated by edges of the entire path from the root to the leaf is exactly the suffix string starts from the position i of the string S, Expressed as S{i ... m}..

Definition 2. Generalized Suffix Tree: The set of strings S is composed of n strings S_n whose lengths are $m_1, m_2, \ldots, m_n$. The generalized suffix tree of S is a sub-tree which has $\sum_{k=1}^{n} m_k$ leaves, Each leaf is marked as a tuple (k, l), where, $k \in (l, n)$, $l \in (l, m_k)$. Each non-leaf node has at least two child nodes, each edge has a non-empty string of S. Two edges derived from one node can't contain the same prefix substring. The string concatenated by edges of the entire path from the root to the leaf (i, j) is exactly the suffix string starts from the position i of the string S_i, Expressed as $S_i\{j \ldots m_i\}$.

Fig.1 shows a generalized suffix tree built according to the text "Web Document Suffix Tree Clustering", "Web Document Clustering" and "Suffix Tree Clustering" build, Nodes in suffix tree is drawn as circles(from A to I), each node v represents a phrase derived from the root to the node and a basic cluster which contains the phrase. All leaves in the sub-tree which starting from node v are suffix of phrase v_p.There is a box attached to the end of each leaf node, the number in boxes indicates the documentation source of substring v_p. Each non-leaf node represents overlapping phrases shared by at least two suffixes. The more same documents two clusters have, the more similar the two clusters are, and the two clusters are more likely to be merged into a base cluster.

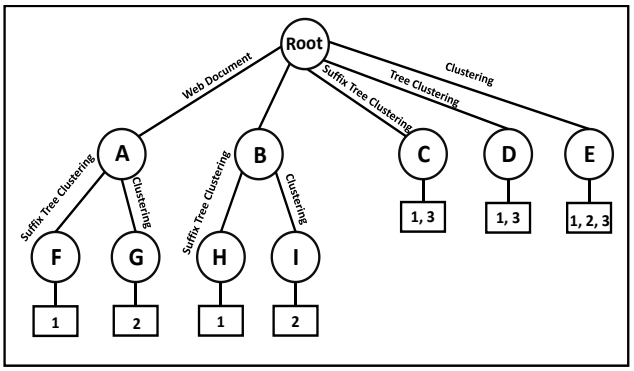

Figure 1. Example of a generalized suffix tree.

IV. IMPROVED SUFFIX TREE CLUSTERING

Similar to STC, improved suffix tree clustering algorithm has three steps: 1) Web document information pre-processing, including the removal of stop-words, the extraction of stem and so on. 2) Suffix tree building and basic cluster selection. 3) Merging basic clusters, to get the final clustering result.

A. Web Document Information Preprocessing

In general, before the document is submitted to the clustering algorithm processing, it must be preprocessed in advance. Preprocessing of the input data is a very important step in information retrieval, efficient documentation preprocessing helps to improve the quality of the cluster algorithm's clustering results.

Before being submitted to the clustering algorithm, the title and content of each page should be preprocessed as a formal document. According to some punctuation, including comma, period, semicolon, question mark and exclamation mark, each document is divided into several sentences. For the sentence words, by removing forms changes such as prefixes and suffixes, all forms of a word are converted to a unique stem, sentences are converted into sequence of strings to increase word description ability. Each sentence boundary is marked by non-word punctuations, tags and non-word elements, Digitals, labels, non-alphanumeric characters and most of the punctuations are removed, original document strings of words and their order are preserved. Therefore, once clustering key phrases are identified, they can be used as catalog directories and tag information to enhance readability. Stop words in predefined list such as articles, auxiliary verbs, pronouns and prepositions are removed in succession.

B. Suffix Tree Building and Basic Cluster Selection

Suffix tree building, namely processing document information sentence by sentence, adding sentences to the suffix tree. Specific process is as follows:

1) Traversal sentence set, fetch all sentences;

2) Traversal the set of sentences, fetch all suffix phrases of sentences;

3) Identify whether a suffix phrase is exists in the suffix tree, if exists, go to step 4), otherwise go to step 5);

4) Add suffix phrases' information to suffix tree matching nodes;

5) Create a new node of suffix phrases, and add the new node to the suffix tree.

Not all nodes can be used as base clusters researchers need, node selection is required to get the basic clusters. In the node selection, each node is assigned a value S(B) according to (1). Where |B| represents the number of documents contained in the basic cluster B , |P| represents the number of words contained in the basic cluster B's corresponding phrase , and f is a linear function where $2 \leq |P| \leq 6$.

$$S(B) = |B| \cdot f(|P|). \quad (1)$$

In order to maintain constness of the next step's basic cluster merging operation, a fixed value of m needs to be confirmed, according to the above calculation results of the ranking, select the first m nodes as candidates for basic cluster clustering.

C. Basic Cluster Merging

After selection of basic clusters, these basic clusters will be merged by the similarity between them to form the clustering result. Traditional similarity calculating mainly considers the overlap of two basic clusters' sentences and documents, such calculation exist two problems as follows: 1) The situation where two basic clusters have a big gap between the number of documents and a relationship of inclusion cannot be dealt effectively, basic clusters which contain similar text but different themes may be merged into a single cluster. One specific example is that when a parent node is selected as the basic class and so as its child node (from the perspective of documentation, the two have the relationship of inclusion) may have these problem. 2) To consider only the overlapping part of the two basic clusters in proportion, ignoring documents whose text is not similar but the theme are the same may exist in the rest of the documents, in such a case, the two basic clusters where most documents share the same theme cannot be merged. Traditional algorithms cannot characterize these properties, because it only takes into account the situation where two basic clusters contain the same documents while the corresponding themes are different. In order to get better clustering results, this paper considers the similarity information between texts of different basic cluster and the relationship between them, using Pearson's correlation coefficient to establish the following similarity calculation model:

$$\mathrm{Sim}(C_i, C_j) = |\rho_{C_i C_j}| = \left|\frac{\mathrm{cov}(C_i, C_j)}{\sigma C_i \sigma C_j}\right| = \left|\frac{\mathrm{E}\left((C_i - \mu C_i)(C_j - \mu C_j)\right)}{\sigma C_i \sigma C_j}\right|$$

$$= \left|\frac{\mathrm{E}(C_i C_j) - \mathrm{E}(C_i)\mathrm{E}(C_j)}{\sqrt{\mathrm{E}(C_i^2) - E^2(C_i)}\sqrt{\mathrm{E}(C_j^2) - E^2(C_j)}}\right|. \quad (2)$$

Where the numerator is the covariance, the denominator is the product of the two variables' standard deviation. Obviously, the standard deviation of C_i and C_i cannot be zero. C_i and C_i are the space vector models of the two basic clusters' documents, which is calculated using the equation:

$$C_i = (c_{i1}, c_{i2}, \ldots, c_{iN}). \quad (3)$$

$$c_{ij} = |c_i \cap c_j|. \quad (4)$$

Where N is the total number of basic clusters. Once the linear relationship between two vectors increases, the correlation coefficient Sim tends to 1, on the contrary it tends to zero. For example, when two vectors are the same, the similarity between them is exactly 1.

In specific calculation process, the data must first be centered, that is subtracted each data with vector's mean value. After centering, their average is zero, $\mathrm{E}(C_i) = \mathrm{E}(C_j) = 0$, then:

$$\mathrm{Sim}(C_i, C_j) = \left|\frac{\mathrm{E}(C_i C_j)}{\sqrt{\mathrm{E}(C_i^2)}\sqrt{\mathrm{E}(C_i^2)}}\right| = \left|\frac{\frac{1}{N}\sum_{k=1}^{N} c_{ik} c_{jk}}{\sqrt{\frac{1}{N}\sum_{k=1}^{N} {c_{ik}}^2}\sqrt{\frac{1}{N}\sum_{k=1}^{N} {c_{jk}}^2}}\right|$$

$$= \left|\frac{\sum_{k=1}^{N} c_{ik} c_{jk}}{\sqrt{\sum_{k=1}^{N} {c_{ik}}^2}\sqrt{\sum_{k=1}^{N} {c_{jk}}^2}}\right| = \left|\frac{\sum_{k=1}^{N} c_{ik} c_{jk}}{||C_i||||C_i||}\right|. \quad (5)$$

That correlation can be seen as the cosine function of the angle between the two centered cluster vectors. Further, when the C_i and C_j vectors are normalized, $||C_i||=||C_j||=1$, the correlation coefficient is the product of the two vectors $\rho_{C_i C_j} = C_i \cdot C_j$.

If the documents of two basic clusters have a little or even no overlap, while their themes of document are very similar, these two basic clusters will not be merged according to the traditional STC algorithm. While the algorithm of this paper deal with such situations, if the themes of their documents are similar, it is likely that the documents will appear in other basic clusters. In the calculation process of the basic clusters' vector space model, the correlation coefficient between the basic clusters' vector increases, it contributes to the merging of this two clusters.

V. Experiments

In order to verify the performance of improved STC algorithm, Experiment id designed to compare the different performance with traditional STC algorithm.

A. Data Set

Researchers used the two Web document data sets used by literature [14], the data sets are Google's relevant results for search query "Jaguar" and "Salsa", including 800 different web documents which are divided into 53 different categories. These two data sets are selected because the Google search results of them contain pages of various topics. Search results of "Jaguar" may contain results about "car", "animal" and "game", and search results of "Salsa" may contain results about "dance", "food" and "foreign". Jaguar dataset contains 420 documents which are divided into 34 different categories, Salsa dataset contains 396 different documents which are divided into 21 different categories. The data sets contain the complete text of Web information which is available online at http://www.danielcrabtree.com/research/wi05/rawdata.zip.

B. Assessment Criteria

Precision and recall of the cluster j for category i are defined as:

$$\text{Precision}(i,j) = \frac{N_{ij}}{N_j}. \quad (6)$$

$$\text{Recall}(i,j) = \frac{N_{ij}}{N_i}. \quad (7)$$

Where N_j is the number of documents included in cluster j, N_i is the number of documents included in category i. N_{ij} is the number of common documents of cluster j and category i.

F-measure of cluster j for category i is defined as:

$$F(i,j) = \frac{2 \cdot Precision(i,j) \cdot Recall(i,j)}{Precision(i,j) + Recall(i,j)}. \quad (8)$$

F-measure of overall clustering results is defined as:

$$F = \frac{\sum_i (|C_i| \cdot F(i))}{\sum_i |C_i|}. \quad (9)$$

Where $|C_i|$ is the number of documents in category i, and F(i) is the F-measure of category i, which takes the maximum in all clusters' F-measure. The greater the F-measure means the better clustering performance.

C. Experiment Results and Analysis

During the suffix tree clustering, researchers took eight different similarity thresholds from 0.3 to 1.0, according to statistics of the first 15 clusters returned by clustering, and compared the performance between original STC algorithm and the improved STC algorithm by F-measure. The results are shown in Fig.2 and Fig.3.

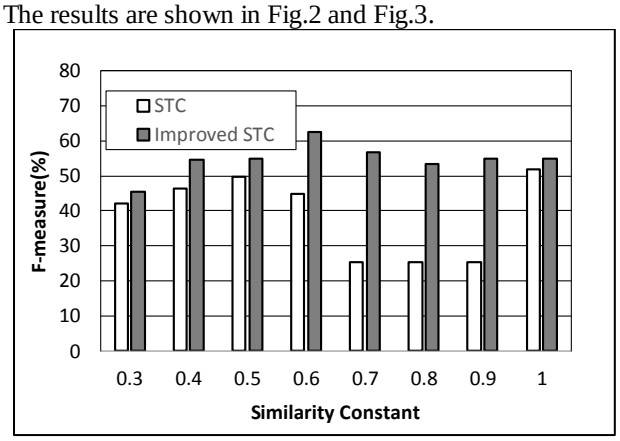

Figure 2. Clustering result of Jaguar case

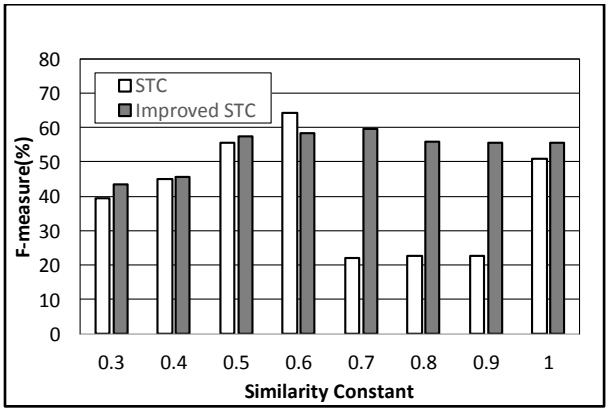

Figure 3. Clustering result of Salsa case

According to Fig.2 and Fig.3, researchers can see that in most cases, the improved algorithm outperforms the original STC algorithm. In Salsa case of experiment, when the similarity thresholds is 0.6, the improved algorithm's F-measure is slightly lower than the original algorithm. After detailed analysis of the data, researchers found that the precision of the improved algorithm was slightly decreased, but the recall was still increased. The improved STC algorithm got the max F-measure of 62.5% and 59.7% when similarity thresholds are set as 0.6 and 0.7 in Jaguar case and Salsa case.

The original STC algorithm is more sensitive with the similarity threshold selection, when the similarity threshold is increased, the performance of clustering declines obviously. This is because the original STC algorithm considers only the overlap of documents between two clusters when merging clusters. While the improved algorithm takes the correlation of clusters into account, leading to a more stable clustering performance. Experiment results show that the improved clustering algorithm not only has better clustering performance, but also displays better stability.

VI. CONCLUSIONS

This paper presents an improved STC algorithm, which combined vector space model with Pearson's correlation coefficient according to the characteristics suffix tree structure, considers the correlation between the basic clusters in the basic cluster merging process. Experimental results show that the improved algorithm performs better than the original algorithm and improves the quality of clustering. Future work is to further improve the clustering performance by bringing in better preprocessing and combining the algorithm with other clustering algorithms.

REFERENCES

[1] Weiner P. Linear pattern matching algorithms[C]//Switching and Automata Theory, 1973. SWAT'08. IEEE Conference Record of 14th Annual Symposium on. IEEE, 1973: 1-11.

[2] McCreight E M. A space-economical suffix tree construction algorithm[J]. Journal of the ACM (JACM), 1976, 23(2): 262-272.

[3] Ukkonen E. On-line construction of suffix trees[J]. Algorithmica, 1995, 14(3): 249-260.

[4] Gusfield D. Algorithms on strings, trees and sequences: computer science and computational biology[M]. Cambridge university press, 1997.

[5] Zamir O, Etzioni O, Madani O, et al. Fast and Intuitive Clustering of Web Documents[C]//KDD. 1997, 97: 287-290.

[6] Zamir O, Etzioni O. Web document clustering: A feasibility demonstration[C]//Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998: 46-54.

[7] Wang J, Mo Y, Huang B, et al. Web search results clustering based on a novel suffix tree structure[M]//Autonomic and Trusted Computing. Springer Berlin Heidelberg, 2008: 540-554.

[8] Wu J, Wang Z. Search results clustering in chinese context based on a new suffix tree[C]//Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on. IEEE, 2008: 110-115.

[9] Worawitphinyo P, Gao X, Jabeen S. Improving suffix tree clustering with new ranking and similarity measures[M]//Advanced Data Mining and Applications. Springer Berlin Heidelberg, 2011: 55-68.

[10] Moe R E. Improvements to Suffix Tree Clustering[M]//Advances in Information Retrieval. Springer International Publishing, 2014: 662-667.

[11] Janruang J, Guha S. Applying Semantic Suffix Net to suffix tree clustering[C]//Data Mining and Optimization (DMO), 2011 3rd Conference on. IEEE, 2011: 146-152.

[12] Rafi M, Maujood M, Munawar Fazal M, et al. A comparison of two suffix tree-based document clustering algorithms[C]//Information and Emerging Technologies (ICIET), 2010 International Conference on. IEEE, 2010: 1-5.

[13] Janruang J, Guha S. Semantic suffix tree clustering[C]//First IRAST International Conference on Data Engineering and Internet Technology, DEIT. 2011.

[14] Crabtree D, Gao X, Andreae P. Improving web clustering by cluster selection[C]//Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on. IEEE, 2005: 172-178.