LET:Towards More Precise Clustering of Search Results*

Yi Zhang, Lidong Bing,Yexin Wang, Yan Zhang †
State Key Laboratory on Machine Perception
Peking University,100871 Beijing, China
{zhangyi, bingld,wangyx,zhy}@cis.pku.edu.cn

Abstract

Web users are always distracted by a large number of results returned from search engines. Clustering can efficiently facilitate users' browsing pages of certain topic. However, most traditional clustering methods are based on either content analysis or link analysis alone, which appears unilateral. In this paper, we propose an expanding clustering idea with the reasonable combination of content and link analysis. Experimental results on Google's three query sets show that our LET algorithm outperforms traditional methods such as K-means.

1 Introduction

Clustering can efficiently alleviate the difficulty of locating the target information. Most of the previous clustering studies are based on term frequency and they aim at clustering text documents instead of web pages. Even though these methods could be applied to web pages, web pages have their own special features which are excessively valuable for clustering. First, there are no hyperlinks between text documents, while they are widely used in web pages. Secondly, for search results, the ranking method of the clusters should not be neglected. Thus, a quick clustering algorithm with an appropriate display is of great importance.

Considering the differences between text documents and web pages mentioned above, we propose a novel expanding idea called LET. We treat a page's link structure as an important and effective complement to its content. Through careful analysis and experiments, we find that merging the content of a page with the information of its child pages in some degree can obviously emphasize the main subject of the parent page and help clustering. Nevertheless, merging the child pages is not simply to include everything without consideration. Though the child pages and their parent page probably will share some topics, they surely will not be exactly the same. To emphasize this point, we bring out the concept-expanding factor which we regard as the acceptance weight for parent page to decide how much it needs to extend the content from its child page. Our approach can be applied to all content-based algorithms. In this paper, we choose the K-means clustering algorithm as the basic of our experiments.

The rest of the paper is organized as follows. The related work is presented in Section 2. Section 3 describes LET algorithm in detail. Section 4 evaluates LET based on experimental results. Finally, we conclude this paper with a discussion and future work in Section 5.

2 Related Work

A variety of clustering algorithms have already been applied to cluster web documents such as hierarchical clustering method [4], K-means [6] and so on. Many methods are based on common words or phrases shared among the pages and topically cluster them into coherent groups according to the similarity measure, while others study the contributions of link analysis to improve the clustering quality of search results.

In 1999, Zamir and Etzioni presented a Suffix Tree Clustering [12, 13]. The method divides documents into different subsets according to their common phrases, and then creates clusters based on the subsets.

In 1998, Kleinberg first proposed that a link between web pages is a valuable implication of their related relations and put forward the well known HITS algorithm [7]. In 2002, Wang and Kitsuregawa developed a link-based clustering algorithm [10]. Their approach is based on common links shared by pages and to cluster the search results by exploring both co-citation and coupling. Afterwards, they presented improved clustering method by considering more factors such as common terms, in-links and out-links shared among pages [11].

Besides these ideas, some work also tried to utilize other techniques such as machine learning [14, 8].

* Supported by NSFC under Grant No.60673129 and 60573166, Beijing Natural Science Foundation under Grant No.4073034

† Corresponding author

3 The LET Algorithm

3.1 Assumptions and Definitions

Our idea is based on the assumption that properly adding the necessary contents of child pages to their parent page will effectively improve the performance of clustering. However, we cannot declare that two connected pages are exactly on the same topic. We define expanding factor to measure how much a page could obtain the content contribution from its child pages. The value of expanding factor can also be deduced by the surfing behaviors of web users. When a user finishes viewing one page, he or she randomly chooses one link of the current page and jumps to it with the probability ϵ, which is also called damping factor in calculating PageRank [3] or jumps to another page uniformly selected from the collection of the pages with the probability 1-ϵ. The reason that the user chooses whether to follow or leave is strongly influenced by the anchor text of the link which could indicate the main idea of the child page. Hence the jumping probability ϵ can express the tightness between the page and its child pages.

Expanding Factor

$$\alpha = \frac{\epsilon}{n} \tag{1}$$

where ϵ is the damping factor and n is the number of contributive links one page owns that are not for advertisement or entertainment.

3.2 Data Preprocessing

Since Chinese words are more challenging due to their homonymy and uncertainty, we use Chinese in our experiments. Given the whole pages set, first, we extract all unique terms from the entire pages set according to the Chinese words library. Then we eliminate "stopwords" such as"and", and "a" from terms of each page. After that, for each web document, we get a bag of words about the main topic of the page. Afterwards, we calculate Phrase Frequency/Inverted Documents Frequency which is usually called TFIDF of each term in accordance with the definition of Equation 2 to decide which words should be chosen as the better candidates of salient phrases for each page.

Phrase Frequency / Inverted Document Frequency

$$TFIDF = f(w) \cdot log \frac{N}{|D(w)|} \tag{2}$$

where for a word w in one document,f represents term frequency calculation, N is the number of the documents in the dataset and D means document frequency.

In addition, words appear in different parts of the web document should be given different weights. Therefore, we enlarge phrase frequency of the title word several times to emphasize its weight in the whole word set. Finally, we treat the first M terms with highest TFIDF values of each web document as the salient terms and each web document can be presented as a base vector, of which the TFIDF value of each salient term is considered as one feature.

3.3 Content Expanding

Words that come from the page itself sometimes can not sufficiently deliver the main points of the page sometimes. The contents of its child pages can help to identify its topic. We extend the dimensions of each page's base vector by considering the base vector of its child pages.

However, not every base vector of the child pages needs to be taken into account. Always many links are not related to the content of their parent pages. To solve this problem, we find a pellucid method to decide which child pages will be included during the expanding process. First, for each page, we can easily get all its child pages. We check up each title of them with the base M-dimension vector of their parent page. If the title of a child page contains one or more salient terms occurring in its parent page' base vector, we suppose the page is a contributive child page and can safely conclude that it likely relates to its parent's content to some extent.

Expanding factor offers us a measure to determinate how much the parent page's content should be extended. On holding the two base vectors, of which one is the parent's and the other is its contributive child's, we can compute the expanded vector of each parent page, which is introduced as follows.

Expanded Vector

Let V_e be the base vector of a parent page V. We suppose Q_i is the ith base vector of its contributive child pages. After content expanding, we finally obtain the expanded vectors V_e'.

$$V_e' = V_e + \alpha \cdot \sum_{i=1}^{n} Q_i \tag{3}$$

where n is the number of all V's content related outlinks, α is the Expanding Factor mentioned before.

3.4 Implemented with K-means Algorithm

The simple expanded vector of each document is calculated and each parent page is represented as a vector with M dimensions. We use the well known text clustering K-means algorithm to cluster the web documents.

When a query is submitted, a search engine first finds the relevant pages in the page repository. Then it implements

K-means algorithm on their pre-calculated expanded vectors. The key idea of K-means algorithm can be described in three steps. First, we select K vectors of web documents as initial cluster centers from the dataset. Secondly, each page is assigned to the nearest center in according to the similarity between the page and the correspondent cluster center. This process iterates until the criterion function converges. Finally, clusters are generated and can be shown to users.

Although it is a query dependent algorithm, the online computation is efficient and bearable. K-means itself guarantees that the time complexity is only O(mn), where n is the number of pages that need to be clustered and m is the number of iterations to guarantee that centroids of all clusters no longer change. Because of its linear time referring to the whole pages set, results can be shown in time so that this query specific clustering algorithm can be practically operated in search engines.

3.5 Ranking the Results

Most search engines prefer to give bigger clusters a larger priority to rank ahead such as Vivisimo [1]. However, this method is not reasonable because a cluster containing a large amount of results does not mean the more importance it obtains. So we rank different clusters in according to the highest PageRank value it holds. If one cluster contains the page with a higher PageRank, it will be listed preferentially. In a cluster, we rank the pages based on their PageRank values.

4 Experiments and Evaluations

In this section we conduct several experiments to validate the effectiveness of LET algorithm. We evaluate our clustering method with comparison to traditional K-means algorithm.

4.1 Data

We carry out our experiments on three different query results returned from Google. Referring to the query selection in [14], three queries are chosen from three main types of queries, which are ambiguous queries, entity names and general terms. They are Jordan, WangGang and Investigation in Chinese.

First, we submit the three queries respectively to the query box of Google. Then we download all the result pages in the returned list using our crawler. Finally, we have three collections of dataset. The number of pages in each collection is shown in Table 1.

Table 1. Statistics on the Collections

Query Type	Query Words	Number of Pages
Ambiguous query	Jordan	802
Entity names	WangGang	914
General terms	Investigation	828

It is not possible for us to obtain the whole dataset as Google owns. To simulate the expanding process, we acquire all their child pages by following the links of these pages in the three collections in advance.

4.2 Evaluation Metrics

There is no agreement for people to measure the quality of clusters. Some evaluation methods use entropy [2] to measure the goodness or purity of the clustering results in comparison with known classes. Some methods introduce distinct distance functions [9] to measure the difference between clustering results and labeled results. In our experiments, we use two classical metrics precision and recall to estimate the quality of clusters. We manually check all the search results for each of three topics and mark each document to its relevant cluster. Then we compare clustering results generated by our algorithm with the results marked by human. For each cluster, precision and recall are defined as follows:

Precision

Let P be the clustering precision.

$$P = \frac{|C \cap T|}{|T|} \tag{4}$$

where C is the set of manually tagged correct web documents to the cluster, T is the set of all the pages grouped to the cluster by our algorithm.

Recall

Let R be the clustering recall.

$$R = \frac{|C \cap T|}{|C|} \tag{5}$$

The meanings of C and T are same as in Equation 4.

4.3 Parameter Selection

During the expanding process, we set ϵ=0.8 which is usually adopted by most ranking algorithms. We assume that a web user browses randomly by following the links of the current page with the probability 0.8 because he or she is attracted by the anchor texts of its links. In our experiment, we will prove the reasonability and effectivity of the assumption. In addition, we set M=100 and use a 100-dimension vector to represent the main idea of each document. Besides, it has been observed that the majority of Web searchers ,approximately 80%, view no more than

Table 2. Main Subtopics for Each of Three Collections

Sub topics	Jordan	WangGang	Investigation
1	Basketball player	Government officer	On marriage
2	Woman superstar	Actor	On educaiton /scientific research
3	Shoes brand	Teleplay	On car
4	On-line sale	Professor	Geological survey
5	Player Jordan's partner Scotti pippen	Manager	On network development
6	...	...	...

three results pages of search engines [5]. Because our ranking method makes every 10 clusters display in one resluts page, we set K=30 to generate three results pages and each collection will be grouped into 30 clusters.

4.4 Experimental Results

4.4.1 Clustering Results

We implement our algorithm and traditional K-means algorithm individually on the three collections. Table 2 lists the main meaningful produced clusters.

As we can see in table 2, our proposed approach successfully discerns some medium and semantic groups from the documents of the three different topics. Taking the query Jordan for example, besides our usual expectation, our method can distinguish some smaller but valuable subtopics such as a known shoes brand in China and several on-line sale sites.

4.4.2 Performance Comparison

From Figure 1, we can see the clustering precision at the top 10, top 20, top 50, top 100 and top 200 results returned by Google. The number following LET and K-means stands for the corresponding query collection. Number 1 is for query Jordan, 2 is for query WangGang and 3 is for query Investigation. It is obviously seen that our LET algorithm shows a noticeable clustering precision improvement for the top ranking pages of all the three collections. Besides, the clustering performance of our algorithm is particularly better in the first several search results, which could make page with a higher PageRank value easily and precisely found.

For each topic, by using our clusters ranking method, we compare each precision and recall of first N clusters generated by our algorithm LET with those of the original K-means algorithm. We take the top5, top 10, top 15 and top 20 ranking clusters to compare the performance between LET and K-means. Experimental results indicate that the clustering precision of three collections averagely improves 30% and the recall increases 25%. Figure 2 shows the curves for the collection of query WangGang.

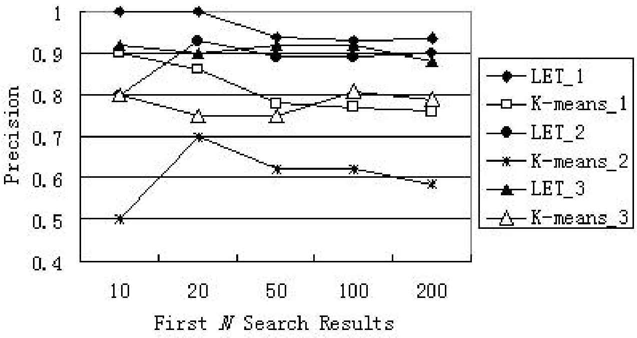

Figure 1. Precision of Clustering Top N Pages in the Three Collections

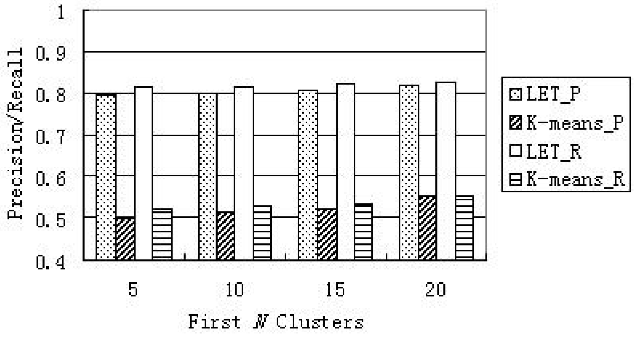

Figure 2. Precision and Recall of First N Clusters of Query WangGang

It is obviously seen that our LET algorithm gives a higher performance than traditional K-means algorithm on the precision and recall of clustering results. Based on our clusters ranking method, cluster that owns the page with a higher pagerank value will be shown ahead. As shown in the experiments, the first several clusters returned to users provide good clustering results, which will satisfy users' retrieval requirements at their first glance.

4.4.3 Influence of the Expanding Factor

To validate our estimate on expanding factor, we implement the experiment by setting different α parameter on acquired collections of Google. For each collection, it achieves the best performance when we set expanding factor α=0.8. Take the query collection of WangGang for example, seen from Figure 3, it could partially verify some rationality of our assumption on the choosing of expanding factor. The reason might be that we pay a careful consideration on the behaviors of web users. In addition, from the experiment, we can see that the performance depends on the search results returned by the web search engine to a certain extent. On holding a collection, we are not sure whether the clustering quality will be better or not when we set expanding factor α=0.85. However, we can affirmatively say that both over emphasizing and completely ignoring the contents of its child pages are not advisable.

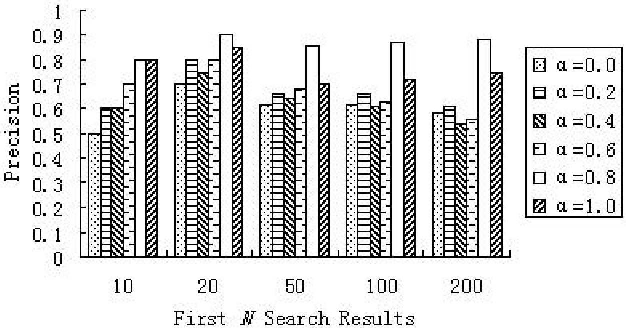

Figure 3. Influence of α on Clustering Performance of Query WangGang

5 Conclusion and Future Work

In this paper, we felicitously combine content analysis with link structure of page on the concern of the unique features of web documents and propose an expanding idea to cluster web documents. Our experiments on the query collections returned by Google yield good results and show that our algorithm LET really improves the clustering quality and make it easy for users to locate target information. Moreover, we also conduct experiments to evaluate the influence of choosing different expanding factors on the dataset. Because of the simplicity and linear complexity of the LET algorithm, it is feasible to operate our method on a real commercial search engine.

Currently we use K-means method in our implementation and we will encounter the same problems as K-means. Nevertheless, due to the facility of our idea, it can be easily combined with other text clustering methods to avoid these troubles. As future work, we will apply this idea to other algorithms and compare their clustering performance.

References

[1] Vivisimo clustering engine. www.vivisimo.com.

[2] F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *Proc.of the 8th ACM SIGKDD*, pages 436–442, New York:ACM Press, 2002.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc.of 13th International World Wide Conference*, May 1998.

[4] E.M.Voorhees. Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Information Processing and Management*, 22:465–475, 1986.

[5] B. J. Jansen and A. Spink. An analysis of web documents retrieved and viewed. In *International Conference on Internet Computing*, Las Vegas, America, June 23-26 2003.

[6] J.J.Rocchio. Document retrieval systems-optimization and evaluation. *Ph.D.Thesis*, 1966.

[7] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, New York,USA, January.

[8] J. Lee and D. Lee. An improved cluster labeling method for support vector clustering. *IEEE transactions on pattern analysis and machine intelligence*, 27, 2005.

[9] P. Pantel and D. Lin. Document clustering with committees. In *Proceedings of SIGIR-02*, Tampere, Finland, August 11-15 2002.

[10] Y. Wang and M. Kitsuregawa. Use link-based clustering to improved web search results. In *Proceedings of WISE 2001*, pages 119–128, New Orleans,Louisiana,USA, 2001.

[11] Y. Wang and M. Kitsuregawa. On combining link and contents informaiton for web page clustering. In *Proceedings of DEXA-02*, Aix-en-Provence, France, September 2-6 2002.

[12] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of SIGIR-98*, Melbourne, AU, 1998.

[13] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to web search results. In *WWW8*, Toronto, May 1999.

[14] H. Zeng, Q. He, Z. Chen, and W. Ma. Learning to cluster web search results. In *Proceedings of SIGIR-04*, Sheffield, South Yorkshire, UK, July 25-29 2004.