Link Based Clustering of Web Search Results

Yitong Wang and Masaru Kitsuregawa

Institute of Industrial Science, The University of Tokyo
{ytwang, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract. With information proliferation on the Web, how to obtain high-quality information from the Web has been one of hot research topics in many fields like Database, IR as well as AI. Web search engine is the most commonly used tool for information retrieval; however, its current status is far from satisfaction. In this paper, we propose a new approach to cluster search results returned from Web search engine using link analysis. Unlike document clustering algorithms in IR that based on common words/phrases shared between documents, our approach is base on common links shared by pages using co-citation and coupling analysis. We also extend standard clustering algorithm K-means to make it more natural to handle noises and apply it to web search results. By filtering some irrelevant pages, our approach clusters high quality pages into groups to facilitate users' accessing and browsing. Preliminary experiments and evaluations are conducted to investigate its effectiveness. The experiment results show that clustering on web search results via link analysis is promising. **Keywords:** link analysis, co-citation, coupling, hub, authority

1. Introduction

Currently, how to obtain high-quality information from the Web efficiently and effectively according to user's query request has created big challenges for many disciplines like data engineering, IR as well as data mining because of features of the Web (huge volume, heterogeneous, dynamic, semi-structured etc.) Web Search engine is the most commonly used tool for information retrieval on the web; however, its current status is far from satisfaction for several possible reasons:

1. Information proliferate on the Web;
2. Different users have different requirements and expectations for search results;
3. Sometimes search request cannot be expressed clearly just in several keywords;
4. Synonym (different words have similar meaning) and homonym (same word has different meanings) make things more complicated;
5. Users may be just interested in "most qualified" information or small part of information returned while thousands of pages are returned from search engine;
6. Many returned pages are useless or irrelevant;
7. Many useful information/pages are not returned for some reasons

So many works [1][2][3] [15][18] try to explore link analysis to improve quality of web search results or mine useful knowledge on the web since links of one page could provide valuable information about "importance" or "relevance" of the page under consideration. [1] proposes that there are two kinds of pages in search results: “hub page” and “authority page” and they will reinforce each other. Its preliminary experiments indicated that HITS [1] could present some “high-quality” pages on the query topic.

While HITS may provide a choice for 6th item and 7th item of what we discussed above, further studies are needed for other items. We think that clustering of *web search results* would help a lot. While all pages in search results are already on the same general topic, by presenting search results in more narrow and detailed groups users could have an overview of the whole topic or just select interested groups to browse. In the rest of this paper, when we talk about *web search results/search results,* we mean web pages returned from web search engine on a specific query topic. We use URLs or pages interchangeably when referring to search results.

Although traditional document clustering algorithms that based on term frequency could be applied to web pages, we would like to reconsider clustering of web search results by taking account of some features of web page:

1. *Hyperlink* between web pages is the main difference between text documents and web pages. It may provide valuable information to group related pages.
2. Most web pages in search results are usually *top pages* of web sites, which mean that they probably just include some links and pictures instead of concrete contents. (This makes term-based clustering algorithms poorly worked)
3. Web pages are written in multiple languages. (Term-based clustering algorithms are difficult to be applied to web pages written in languages other than English.)

We also emphasize some requirements for clustering of web search results, which has been stated in [7]:

1. Relevance: *Not all web pages* but high-quality pages in search results need to be clustered. Clustering should separate related web pages from irrelevant ones.
2. Overlap: One web page could *belong to more than one cluster* since it could have more than one topic.
3. Incrementally: In order for speed, clustering procedure should start to process one page as soon as it arrives instead of waiting all information available.

In this paper, we study contributions of link analysis to clustering of web search results. Our idea is very simple: *pages that share common links each other are very likely to be tightly related*. Here, common links for two web pages *p* and *q* mean common *out-links* (*point from p* and *q*) as well as common in-links (*point to p* and *q).* Especially, we only consider *non-nepotistic links* (hyperlinks between pages from different websites) since we think that hyperlinks within the same website are more to reveal the inner-structure (like site-map) of the whole website than implying a semantic connection. Our approach combines link analysis and extension of cluster algorithm *K-means* so that it can overcome disadvantages of standard K-means and meet requirements for clustering of web search results.

The paper is organized as follows: next section is an assessment of previous related works on clustering in web domain. In section3, a detailed description of the proposed approach is given. Subsequently in section4, we report experiment results on different query topics as well as evaluations. The paper is concluded with summary and future work directions.

2. Background

Clustering analysis has a long history and serves for many different research fields, like data mining, pattern recognition as well as IR. Vector Space Model, also called *TFIDF* method is the most commonly used one for document representation in IR, which based on terms frequency. Various measurements of similarity between different documents could be applied and one popular way is *Cosine* measurement. *K-means* and *agglomerative hierarchical clustering* are two commonly used methods for document clustering in IR. K-means is based on the idea that a center point (*Centroid*) can represent a cluster. K-means cluster *N* data points into *K* flat (one-level) groups. The advantage of K-means is its speed and its disadvantage is that the quality and structure of final clusters will depend on the choice of k value and k initial centroids. In contrast to K-means, hierarchical clustering creates a nested sequence of partitions, with a single, all-inclusive cluster at the top and singleton clusters of individual points at the bottom. According to [5], hierarchical clustering produces "better" clusters with high time complexity. Detailed description and comparison of document clustering could be found in [5][10][13].

2.1 Prior Related Work on Clustering Search Results

Related work can be classified into following categories: clustering hypertext documents in a certain information space and clustering web search results. As for clustering web search results, some works are basing on the whole document and some works are focusing on clustering snippet attached with each URL in search results in order to achieve speed. Snippet is considered as a good summary to capture the main idea of the page under consideration.

[9] propose a hierarchical network search engine that clusters hypertext documents to structure a given information space for supporting various services like browsing and querying. All hypertext documents in a certain information space (e.g one website) were clustered into a hierarchical form based on contents as well as link structure of each hypertext document. By considering about links within the same website, related documents in the same website could be grouped into one cluster. However, our target is not general situation but search results classification, which clusters search results into more narrow and detailed groups.

[11] explores clustering hypertext documents by *co-citation analysis* (its explanation is in section2.2). First, co-citation pairs are formed with their co-citation frequency. Co-citation pairs whose co-citation frequencies are above pre-specified *threshold* will be kept for further processing. Final clusters are generated by iteratively merging co-citation pairs that share one document. [11] also indicated that its approach could be applied to WWW. However, if AB is a co-citation pair that co-cite document set *f1* and BC is another co-citation pair that co-cite document set *f2*, then document C is added to cluster AB regardless of whether *f1* and *f2* are disjoined or not will sometimes lead to arbitrary decision.

Scatter/Gather [11] is a document browsing system based on clustering, using a hybrid approach involving both k-means and agglomerative hierarchical clustering.

It proposes in [7] an algorithm called suffix Tree Clustering (STC) to group together snippets attached with web pages in search results. The algorithm use techniques that construct a STC tree within linear time of number of snippets. Each node in this tree captures a phrase and associates it with snippets that contain it. After obtaining base clusters in this way, final clusters are generated by merging two base clusters if they share majority (50%) members. Since snippets usually bring noises and outliers, [8] proposes an algorithm called fuzzy relational clustering (RFCMdd) based on the idea of identifying k-medoids. [8] compassionates [7] with the ability to process noises and outliers brought by snippets. However, snippets are not always available in search results and they are also not always a good representation of the whole documents for their subjectivity.

2.2 Link Analysis

[1][2][3][15][18] study contribution of link analysis to improve the quality of search results as well as mine communities on the Web. [1] proposes that there are two kinds of pages in search results: Hub and authority.

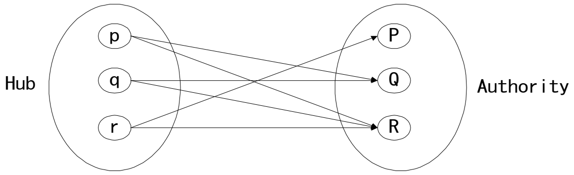

Fig. 1. Potential hub pages and authority pages in search results

Co-citation [21] and bibliographic *coupling* [20] are two more fundamental measures to be used to characterize the similarity between two documents. *Co-citation* measures the *number of citations* (*out-links*) *in common* between two documents and *coupling* measures the number of document (*in-links*) that cites both of two documents under consideration.

In the above Fig.1, *p* and *q co-cite Q* and *R* and their co-citation frequency is 2; *P* and *R are coupled* by *r* and their coupling frequency is 1.

This could also be proved from the computation in HITS [1] about "hub" value and "authority" value of each page:

$$Hub(p) = \sum_{p->p'} Authority(p') = \sum_{p->p'} \sum_{q->p'} Hub(q) \text{ (p, q share common out-links)}$$

$$Authority(p) = \sum_{p'->p} Hub(p') = \sum_{p'->p} \sum_{p'->q} Authority(q) \text{ (p, q share common in-links)}$$

If one page has many out-links and also has high co-citation frequency with other pages, it may be "good hub" and clustered with other pages into one group with high possibility. So do authority pages. Both *co-citation and coupling* are considered in our approach when measuring the similarity between one page and the correspondent cluster.

3. Clustering Based on Link Analysis

Just as indicated above, the underlying idea of our approach is that pages that *co-cite* (share common out-links) or are *coupled* (share common in-links) are with high probability to be clustered into one group. For each URL *P* in search results *R*, we extract its all out-links as well as top *n* in-links by services of AltaVista. We could get all distinct *N* out-links and *M* in-links for all URLs in *R*.

3.1 Definition

- Representation of each page *P* in *R*

Each page *P* in *R* is represented as 2 vectors: P_{Out} (*N*- dimension) and P_{In} (*M*-dimension). The *ith* item of vector P_{Out} is to indicate whether *P* has a out-link as the *ith* one in *N* out-links. If has, the *ith* item is 1, else 0. Identically, the *jth* item of P_{In} is to indicate whether *P* has an in-link as the *jth* one in *M* in-links. If has, the *jth* item is 1, else 0.

- Similarity measure

We adopt traditional *Cosine* measure to capture common links (in-link and out-link) shared by pages *P, Q* that under consideration:

Cosine (*P*, *Q*)= $(P \bullet Q)/(\|P\| \, \|Q\|) = ((P_{Out} \bullet Q_{Out}) + (P_{In} \bullet Q_{In}))/(\|P\| \, \|Q\|)$, *where*

$\|P\|^2 = (\sum_1^N P_{Out\,i}^{\;2} + \sum_1^M P_{In\,j}^{\;2})$ (Total number of out-links and in-links of page *P*),

$\|Q\|^2 = (\sum_1^N Q_{Out\,i}^{\;2} + \sum_1^M Q_{In\,j}^{\;2})$ (Total number of out-links and in-links of page Q),

$(P_{Out} \bullet Q_{Out})$ is dot product of vector P_{Out} and Q_{Out} to capture common out-links share by *P* and *Q* whereas $(P_{In} \bullet Q_{In})$ is to capture common in-links shared by *P* and *Q*. ||*P*|| is length of vector *P*.

- Center Point of Cluster

Centroid or center point *C* is used to represent the cluster *S* when calculating similarity of page *P* with cluster *S*. |S| is number of pages in cluster S. Since centroid is usually just a logical point, its item values could be smaller than 1. So, we have:

$C_{out} = 1/|S| \sum_{P_i \in S} P_{i\,Out}$, $C_{In} = 1/|S| \sum_{P_i \in S} P_{i\,In}$. *Similarity (P, S)=Cosine* (*P*, *C*)

= $(P \bullet C)/(\|P\| \, \|C\|) = ((P_{Out} \bullet C_{Out}) + (P_{In} \bullet C_{In}))/(\|P\| \, \|C\|)$

$\|C\|^2 = (\sum_i C_{Out\,i}^{\;2} + \sum_j C_{In\,j}^{\;2})$, $\|P\|^2 = (\sum_1^N P_{Out\,i}^{\;2} + \sum_1^M P_{In\,j}^{\;2})$

- Near-Common Link of Cluster

Near-common link of cluster means links shared by *majority members* of one cluster. If one link is shared by 50%members of the cluster, we call it "50% near-common link" of the cluster and the link shared by all members of the cluster is called *common link of cluster*.

3.2 Clustering Method

Here we introduce a new clustering method by extending standard K-means to meet requirements for clustering of web search results as well as to overcome disadvantages of K-means. In standard k-means, N data points are clustered into K groups. Value K and K initial centroids have to be pre-defined to start clustering procedure. Our clustering method is:

- Filter irrelevant pages

Not all web pages in search results but high quality pages (in our case, only pages whose sum of in-links and out-links are at least 2 are processed) join clustering procedure. By filtering some irrelevant pages, we could improve the *precision* of final results.

- Define similarity threshold

Similarity threshold is pre-defined to determine whether one page could be clustered into one cluster. Since similarity is meant to capture common links shared by different pages, similarity threshold could be easily defined and adjusted.

- Use *near-common link of cluster* to guarantee intra-cluster cohesiveness

By adjusting to different values, we found 30% near-common link is appropriate and we require that every cluster should have at least one 30% near-common link to guarantee its quality and intra-cluster cohesiveness.

- Assign each page to clusters

Each page *is assigned to existing clusters* if (a) similarity between the page and the correspondent cluster is above *similarity threshold* and (b) the page has a link in common with *near common links* of the correspondent cluster. If none of current existing clusters meet the demands, the page under consideration will become a new cluster itself. Centroid vector is used when calculating the similarity and it is *incrementally* recalculated when new members are introduced to the cluster. While one page could belong to more than one cluster, it is limited to *top 10* clusters based on similarity values. All pages that join clustering procedure are processed sequentially and the whole process is iteratively executed until it converges (centroids of all clusters are no longer changed). While the final result may be sensitive to the processing order, we would further examine it by changing processing order.

- Generate final clusters by merging *base clusters*

When the whole iteration process converges, *base clusters* are formed. Final clusters are generated by recursively merging two base clusters if they share majority members. *Merging threshold* is used to control merging procedure.

The algorithm described above has same *time complexity* (*O(nm)*) with standard K-means, where *n* is the number of pages that join clustering procedure and *m* is the number of iterations needed for clustering process to converge (The convergence is guaranteed by K-means algorithm). Since m <<n, the proposed approach is linear to the number of URLs/ pages that join clustering procedure.

In the proposed approach, not all URLs in search results will join clustering procedure and also not all URLs that join clustering procedure will be grouped with others. We think that this is more natural to handle noises and conforms to heterogeneous feature of the Web. There are three parameters in our approach that may affect quality of final results: *number of in-links*, *similarity threshold* and *merging threshold*, we have tried different values in experiments to investigate their effects.

4. Experiments and Evaluation

4.1 Experimental Environment

We carry out experiments on different query topics to check efficiency and effectiveness of the proposed approach. The whole process is divided into four steps:

- Data collection

In order to test effectiveness, efficiency and scalability of the proposed approach, we carry out experiments with datasets on different query topics, different search engines with different numbers of search results. Since the approach is based on link analysis, different numbers of in-links are also examined. Table 1 gives summary of datasets. We download all pages in search results and extract all out-links for each page as well as its in-links by AltaVista.

- Data cleaning

Since there are so many mirrors or duplicates on the Web, it will mislead clustering process if preserving these duplicates. We adopt a non-aggressive method to remove mirrors or duplicates in search results. Two pages *p* and *q* are said duplicate if (a) they each have at least 8 out-links and (b) they have at least 80% of their links in common. The page with higher common link percentage will be removed. As a result, its associated out-links and in-links are also deleted. Table2 shows how many pages in search results actually join clustering procedure after data cleaning and poor-quality pages (sum of in-links and out-link is less than 2) removal for different topics.

- Applying algorithm proposed in section3 to form base clusters

- Final clusters generation

By applying the proposed algorithm, we obtain base clusters. Final clusters are generated by merging two base clusters if they share majority (e.g.75%) members. The cluster that has higher common member percentage is merged into the other one. To merge cluster A into B, we union members of both A and B under the cluster name B. Table2 shows final clusters obtained for datasets with specified similarity threshold for different query topics.

4.2 Experiment Results

Some statistics about experiments as well as final results are shown in below tables. According to Table2, only 60%-70% pages in search results are preserved for clustering. When requiring more in-links, more information is used for clustering, so pages that join clustering and pages that are clustered into groups also increase.

Dataset	Topic	Number of Pages in Search Result	Search Engine	Number of in-Links required
1	Jaguar (1)	750	Google	100
2	Jaguar (2)	750	Google	20
3	Data mining	200	AltaVista	100
4	Java	400	Yahoo	100

Table 1. Information of testing dataset

Topic/ Similarity Threshold	Number of Pages that join clustering	Avg. Out-Links / Avg. In-links	Iterations when converge	Merging threshold	Number of final clusters
Jaguar (1) /0.1	449	10.1 / 11.0	8	0.75	50
Jaguar (2) /0.1	438	10.3 / 6.4	7	0.75	55
Data mining /0.1	120	13.9 / 20.6	3	0.75	15
Java /0.1	295	7.8 / 53.1	5	0.75	21

Table 2. Some statistics of experimenting after data cleaning

Topic/ Similarity Threshold	Total clusters	Size 2-3	Size 4-5	Size 6-10	Size 11-20	Size 21-40	Size 40-60	Size 60-80	Size above 80	Singleton clusters
Jaguar (1) /0.1	50	29	6	8	2	2	2	1	0	163
Jaguar (2) /0.1	55	31	8	7	4	3	2	0	0	160
Data mining /0.08	14	10	2	1	1	0	0	0	0	60
Java/ 0.08	23	15	2	2	2	1	0	0	1	105

Table 3. Distributions of clusters based on size with merging threshold 0.75

Table3 gives final cluster size distribution for different topics. As results reveal, one page could belong to more than one cluster or belong to singleton cluster, which means that it cannot be grouped with others. Since it is possible for “query topic” to have more than one meaning under different contexts, table3 indicates that the proposed approach could capture main semantic categories around query topic on the web as well as other small groups, which in most cases are pages from the same website.

Threshold of similarity	Topic	Number of final clusters	Singleton cluster	Maximum Cluster Size	Number of clusters with size >3
0.1	Data mining	15	66	13	5
0.08	Data mining	14	60	20	4
0.06	Data mining	12	53	20	8
0.1	Java	21	129	89	6
0.08	Java	23	105	107	8

Table 4. Results of final clusters with different thresholds of similarity

No.	Main topic	URLs in the cluster with this topic
1	*Jaguar Car*	http://www.jagweb.com/ http://www.jaguarcars.com http://www.classicjaguar.com/…
2	*Jaguar Club*	http://www.jag-lovers.org/ http://seattlejagclub.org/ http://www.jagclub.com/…
3	*Magazine on Jaguar car/club*	http://www.kreiha.de/jaguar-magazin-online http://www.jagweb.com/jagworld/ http://www.jcna.com/….
4	*Jaguar Game*	http://atarihq.com/interactive/ http://www.millcomm.com/forhan/jaguar.html http://www.lpl.arizona.edu/~breid/videogames/jaguar.html…
5	*Mammal: Big Cat*	http://www.bluelion.org/jaguar.htm http://lynx.uio.no/catfolk/onca-01.htm http://www.gf.state.az.us/frames/fishwild/jaguar.htm#1…
6	*Touring: Jaguar Reef Lodge*	http://www.belizenet.com/jagreef.html http://www.divejaguarreef.com/ http://www.jaguarreef.com/jagreef/qtvr.html…
7	*Jaguar Racing Car*	http://www.jaguarracing.cz/ http://www.dmoz.org/Sports/Motorsports/Auto_Racing/Formula_One/Teams/Jaguar http://www.jaguar-racing.com/uk/html/…

Table 5. Examples of some main clusters for Topic “Jaguar”

Table4 compares final clusters with different similarity thresholds. By decreasing similarity threshold, we could see that more pages are clustered, maximum cluster size increases, which means more pages belong to main group. Moreover, some medium-size clusters emerge and distinctions between clusters are also not so clear. Table5 presents examples of some main clusters of dataset1. From table5, we could see that pages in the same cluster do share similar topic and contents under the general query topic.

4.3 Entropy-based Evaluation

Validating clustering algorithm and evaluating its quality is complex because it is difficult to find an objective measure of quality of clusters. We decide to use *Entropy* to measure the quality of clusters. Entropy provides a measure of “goodness” for un-nested clusters by comparing the groups produced by the clustering technique to known classes. In our initiative evaluation, we manually check each page that joins the clustering procedure and then give our judgment. Each page is given two estimates: relevance (to the query topic), main topic and then create *classes* manually. Although it is time-consuming and it could lead to bias in our evaluation, we plan to carry out user experiment to counteract potential bias. We adopt the computing of Entropy introduced in [10]: Let CS is a cluster solution and for each cluster j, its entropy is E (j)= $-\sum_{i} p_{ij} \log(p_{ij})$ · p_{ij} is used to compute the “probability” that a member of cluster j belongs to the given class i. The average entropy for a set of clusters is calculated as the sum of entropies of each cluster weighted by its size: $E_{CS} = \sum_{j=1}^{m} \frac{n_j * E(j)}{n}$, where n_j is the size of cluster j, m is the number of clusters and n is the total number of data points. Our evaluations focus on topic “Jaguar” with different merging thresholds.

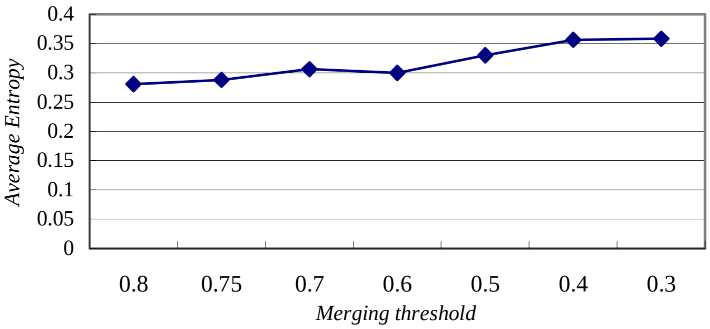

Fig. 2. Effects of merging threshold on Entropy for Topic “Jaguar”

Fig.2 shows overview of the effects of merging threshold on the quality of clusters. It indicates that higher merging threshold gives better results and the proposed approach is insensitive to the changes of merging threshold when merging thresholds are bigger than 0.6. In order to get in-depth understanding, we examine entropy distribution for every cluster with different merging thresholds and the results is shown in Fig.3. Cluster1, 2, …, 24 are in descendent order based on their sizes. For cluster22, since its entropy values are 0 for all merging thresholds, there is no correspondent bar. As merging threshold decreases from 0.8, we could see that some clusters will be merged into other clusters and thus correspondent bars will disappear. While cluster1 has the biggest cluster size for merging threshold 0.8, it is no longer the biggest one with the merging threshold less than 0.75. Instead, cluster3 is the biggest cluster when merging threshold is less than 0.75. This is also demonstrated by its high entropy values. Fig.3 suggests that small-size clusters are very stable and have high qualities since they have low entropy values. According to hand-check results, cluster12, 16 and 21are not semantically interpretable and this is in accordance with their high entropy values revealed in Fig.3. We consider introducing some heuristics to trim them out and adapt the approach to various situations. As for big-size clusters, they have relatively high entropy values and are tempt to change into higher values when merging threshold decreases. While we would like to continually check the result of merging threshold 0.9, 0.75 and 0.8 are our recommendations to generate reasonable clusters.

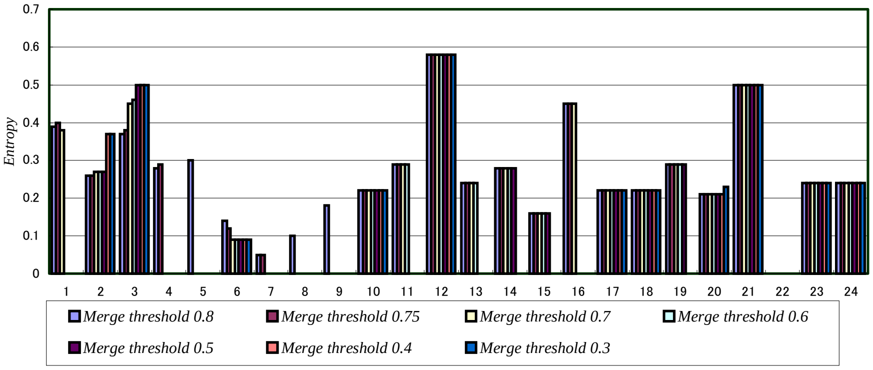

Fig. 3. Entropy of each cluster ordered by cluster size for Topic "Jaguar"

5. Conclusion

In this paper, we propose a new approach to cluster web search results based on link analysis. Our motivations are that currently users are more and more plagued by the inefficiency of web search engine and we would like to study how link analysis could contribute for clustering on search results. Since it is very important for scalability in web domain because of huge volume of information, time complexity of the proposed approach is linear to the number of pages join the clustering procedure. Our approach explores link analysis to filter irrelevant (with query topic) pages and cluster "high-quality" pages into groups to facilitate users' accessing and browsing, which only few works focus on this aspect. In order to get in-depth understanding about effectiveness of the proposed approach to cluster search results, we carry out experiments on different query topic: Jaguar, data mining, Java and evaluate experiment results using entropy metric. We also extend standard K-means algorithm to overcome its disadvantages like compulsory choice of K value and k initial centroids to make it more natural to handle noises. Experiment results indicate that the proposed approach could generate reasonable clusters when merging threshold 0.75 or 0.8 is given.

Since technique of using link analysis to cluster search results is still primitive, further experimenting as well as detailed analyses and interpretation of experiment results are our next step works. Comprehensive comparing with other related research is also needed.

References

1. **Kleinberg 98** *Authoritative sources in a hyperlinked environment*. In proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms (SODA), January 1998.
2. **Ravi Kumar *et. al.* 99** *Trawling the Web for emerging cyber-communities*. In Proceedings of 8th WWW conference, 1999, Toronto, Canada.
3. **Brin and Page 98** *The anatomy of a large-scale hypertextual web search engine*. In Proceedings of WWW7, Brisbane, Australia, April 1998.
4. **Oren Zamir and Oren Etzioni *99*** *Grouper: A Dynamic Clustering Interface to Web Search Results*. In Proceedings of 8th WWW Conference, Toronto Canada.
5. **Richard C. Dubes and Anil K.Jain,** *Algorithms for Clustering Data***,** Prentice Hall, 1988
6. **Oren Zamir and Oren Etzioni 97** *Fast and Intuitive clustering of Web documents,* KDD'97, pp287-290
7. **Oren Zamir and Oren Etzioni 98** *Web document clustering: A feasibility demonstration*. In Proceedings of SIGIR' 98 Melbourne, Australia.
8. **Zhihua Jiang *et. al.*** *Retriever: Improving Web Search Engine Results Using Clustering.* http://citeseer.nj.nec.com/275012.html.
9. **Ron Weiss *et. al.* 96** *Hypursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering*. ACM Conference on Hypertext, Washington USA,1996
10. **Michael Steinbach *et. al.*** *A Comparison of Document Clustering techniques*. KDD'2000. Technical report of University of Minnesota**.**
11. **James Pitkow and Peter Pirolli 97** *Life, Death and lawfulness on the Electronic Frontier*. In proceedings of ACM SIGCHI Conference on Human Factors in computing, 1997
12. **Cutting, D.R. *et. al.92*** *Scatter/gather: A Cluster-based approach to browsing large document collections*. In Proceedings of the 15th ACM SIGIR, pp 318-329; 1992
13. **A.V. Leouski and W.B. Croft. 96** *An evaluation of techniques for clustering search results.* Technical Report IR-76 Department of Computer Science, University of Massachusetts, Amherst, 1996
14. **Broder *et. al.* 97** *Syntactic clustering of the Web*. In proceedings of the Sixth International World Wide Web Conference, April 1997, pages 391-404.
15. **Bharat and Henzinger 98** *Improved algorithms for topic distillation in hyperlinked environments.* In Proceedings of the 21st SIGIR conference, Melbourne, Australia, 1998.
16. **Chakrabarti *et. al.* 98** *Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text.* Proceedings of the 7th WorldWide Web conference, 1998.
17. **Florescu, Levy and Mendelzon 98** *Database Techniques for the World-Wide Web: A Survey*. SIGMOD Record 27(3): 59-74 (1998).
18. **Gibson, Kleinberg and Raghavan 98** *Inferring Web communities from link topology*. Proc. 9th ACM Conference on Hypertext and Hypermedia, 1998.
19. **Agrawal and Srikant 94** *Fast Algorithms for mining Association rules,* In Proceedings of VLDB, Sept 1994, Santiago, Chile.
20. **M.M. Kessler,** *Bibliographic coupling between scientific papers*, American Documentation, 14(1963), pp 10-25
21. **H. Small,** *Co-citation in the scientific literature: A new measure of the relationship between two documents*, J. American Soc. Info. Sci., 24(1973), pp 265-269