Use Link-based Clustering to Improve 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

While web search engine could retrieve information on the Web for a specific topic, users have to step a long ordered list in order to locate the needed information, which is often tedious and less efficient. In this paper, we propose a new link-based clustering approach to cluster search results returned from Web search engine by exploring both co-citation and coupling. Unlike document clustering algorithms in IR that are based on common words/phrases shared among documents, our approach is based on common links shared by pages. 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 in web search results into semantically meaningful groups to facilitate users' accessing and browsing. Preliminary experiments and evaluations are conducted to investigate its effectiveness. The experimental results show that link-based clustering of web search results is promising and beneficial.

Keywords: link analysis, co-citation, coupling, hub, authority

1. Introduction

With information proliferation on the web as well as popularity of Internet, the Web is the biggest data source for various applications. However, how to obtain high-quality information from the Web efficiently and effectively according to user's query request is still a major research problem and all features of Web (huge volume, heterogeneous, dynamic and semi-structure etc.) has created big challenges for many disciplines like data engineering, IR as well as data mining.

While web search engine could retrieve information on the Web for a specific topic, users have to step a long ordered list in order to locate the needed information, which is often tedious and less efficient due to various reasons like information proliferation on the Web; users differ with requirements and expectations for search results; sometimes a search request cannot be expressed clearly with few keywords; synonym (different words have similar meaning) and homonym (same word has different meanings) make things more complicated; sometimes users may be just interested in "most qualified" information or small part of information returned. In short, the accessing (recall and precision) and interpretation of search results of current search engine are far from satisfying.

Kleinberg argued in [1] that links between web pages could provide valuable information to determine related pages (with query topic). So, many works [2,3,15,18] try to explore link analysis to improve quality of web search process or mine useful knowledge on the web. It is proposed in [1] that there are two kinds of pages in search results: "hub page" and "authority page" and they reinforce each other. As its preliminary experiments indicated in [1], HITS algorithm could produce "high-quality" pages on the query topic.

HITS might provide a solution for some problems and further investigation is still in high demand. The goal of our work is to cluster high-quality pages in web search results into more detailed, semantically meaningful groups to facilitate user's searching and browsing. By doing so, users could have an overview of the whole topic or just select interested groups to browse. 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 are based on term frequency could be applied to web pages, we would like to stress some special features of our research target, that is, clustering web search results in a more narrow and detailed groups:

(1) ***Hyperlink*** between web pages is the main difference between text documents and web pages and it may provide valuable information to group related pages.

(2) Many web pages in search results are the *top pages* of web sites, which mean that they probably just include some links and pictures instead of concrete contents. (This makes the traditional document clustering algorithms work poorly)

(3) Web search results is different from a corpus of web pages on that web search results usually focus on a general query topic while a corpus of web pages may cover various topics.

We also emphasize some requirements for clustering of web search results that has been stated in [7]: relevance and overlap. Relevance means that clustering should separate related web pages from irrelevant ones, that is to say, *not all web pages* but high-quality pages in search results need to be clustered and overlap means that one web page could *belong to more than one cluster* since it could have more than one topic.

In this paper, we study contributions of link-based clustering to improve 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).* We only consider hyperlinks between pages from different websites (*non-nepotistic links*) since we think that hyperlinks within the same website are more to reveal the inner-structure (like site-map) of the website than implying a semantic confer. Our approach combines link analysis and extension of cluster algorithm *K-means* so that it can overcome disadvantages of standard K-means and meet the requirements for clustering of web search results.

The paper is organized as follows. Section 2 is an assessment of previous related works of clustering in web domain. In Section 3, we describe link-based clustering algorithm. Subsequently in Section 4, we report experimental results on different query topics, evaluations and comparisons with other clustering algorithm (STC) that is based on the snippet attached with each URL in search results. The paper is concluded with summarizing and future work.

2. Background

Cluster 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, which is based on terms frequency is the commonly used one for document representation in IR. *K-means* and *agglomerative hierarchical clustering* are two fundamental clustering methods in IR. The advantage of K-means is its speed and the disadvantage is that the quality and structure of final clusters will depend on the choice of k value and k initial centroids when clustering n data points into k groups. According to [5], hierarchical clustering produces “better” clusters but with high cost.

2.1 Prior Related Work on Clustering Search Results

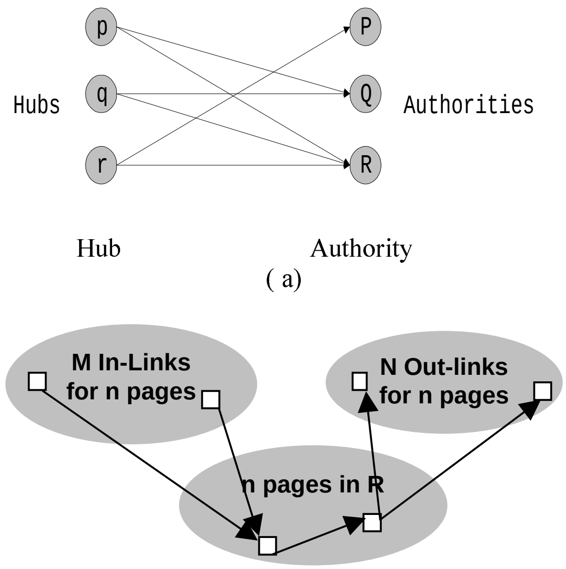
Related works can be classified into following categories: clustering hypertext documents in a certain information space and clustering web search results. As for the latter one, some works are basing on the whole document and some works are focusing on clustering snippet attached with each URL in search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

It is in [9] that a hierarchical network search engine is proposed to cluster 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) can be clustered into a hierarchical form based on the contents as well as the 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 a general situation, but search results classification, which clusters search results into more narrow and detailed groups. In [11] clustering hypertext documents by *co-citation analysis* (its explanation is in Section 2.2) is explored. First, co-citation pairs are formed with their co-citation frequency. Co-citation pairs whose co-citation frequencies are above pre-specified *threshold* are kept for further processing. Final clusters are generated by iteratively merging co-citation pairs that share one document. It is also indicated in [11] that the proposed approach could be applied to WWW. However, if AB is a co-citation pair that co-cites document set *f1* and BC is another co-citation pair that co-cites document set *f2*, then document C is added to cluster AB regardless of whether *f1* and *f2* are disjoint or not. This will sometimes lead to arbitrary decision. Scatter/Gather [11] proposed a document browsing system based on clustering, using a hybrid approach involving both k-means and agglomerative hierarchical clustering.

In [7], an algorithm called Suffix Tree Clustering (STC) is proposed to group together snippets attached with web pages in search results. The algorithm use techniques that construct a STC tree within a linear time of number of snippets. Each node in this tree captures a phrase and associates it with snippets that contain it. If one node in the tree associates more than one snippet, the associated snippets form a base cluster. After obtaining base clusters in this way, final clusters are generated by iteratively merging two base clusters if they share majority (50%) members. Since snippets usually bring noises and outliers, an algorithm called fuzzy relational clustering (RFCMdd), which is based on the idea of identifying k-medoids, is proposed in [8] to compensate the work in [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. Moreover, the fact that in STC algorithm, two snippets is clustered into same group even if they only share on word will lead to very high overlap and in turn generate a big cluster.

2.2 Link Analysis

In [1,2,3,15,18], authors study contributions of link analysis to improve the quality of search results as well as mine communities on the Web.



(a)

(b)

Figure 1. Potential Hub pages and Authority pages in Web 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 Figure 1(a), *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 the "hub" value and "authority" value of each page:

$$Hub(p) = \sum_{p->p'} Authority(p') = \sum_{p->p'} \sum_{q->p'} Hub(q)$$

(*p, q co-cite and share common out-links*)

$$Authority(p) = \sum_{p'->p} Hub(p') = \sum_{p'->p} \sum_{p'->q} Authority(q)$$

(*p* and *q are coupled and 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" (as shown in Figure 1(a)) and clustered with other pages into one group with high probability. 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 since both hubs and authorities appear in web search results.

3. Link-based clustering

By co-citation and coupling analysis, our approach clusters search results based on common links (in-links and out-links) they shared. In the rest of our discussion in the paper, we have several notations: n, m, M, N are positive integers, *R* is the set of specified number of search results. We use n to denote specified number of search results used for clustering, m to denote specified number of in-links extracted for each URL/page in *R*. M and N denote total number of distinct in-links and out-links extracted for all *n* pages in *R* respectively. This is depicted in Figure 1 (b).

1) Representation of each page *P in R*
   Each web 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} indicates whether *P* has the correspondent out-link as the *ith* one in N out-links. If yes, the *ith* item is 1, else 0. Identically, the *jth* item of P_{In} indicates whether *P* has the correspondent in-link as the *jth* one in M in-links. If yes, *jth* item is 1, else 0.
2) Similarity measure
   We adopt the traditional *Cosine* measure to capture common links (in-link and out-link) shared by pages *P, Q* that is under consideration:
   Cosine (*P*, *Q*)= (*P* • *Q)*/(||*P*|| || *Q*||)
   =((P_{Out} • Q_{Out})+ (P_{In} • 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} • Q_{Out}) is the dot product of vectors P_{Out} and Q_{Out}. It captures common out-links share by *P* and *Q* whereas (P_{In} • Q_{In}) is to capture common in-links shared by *P* and *Q*. ||*P*|| is length of vector *P*. Centroid or center point *C* is used to represent the cluster *S* when calculating similarity of page *P* with cluster *S*. Centroid is usually just a logical point.

$C_{out} = {}^{1}\!/_{|S|} \sum_{P_i \in S} P_{i\,Out}$, $C_{In} = {}^{1}\!/_{|S|} \sum_{P_i \in S} P_{i\,In}$. |S| is number of pages in cluster S, C is the centroid of cluster S, item value of vector C could be smaller than 1.

Similarity (P, S)=Cosine (*P*, *C*)= (*P* ● *C*)/||*P*|| || *C*|| $= ((P_{Out} \bullet C_{Out}) + (P_{In} \bullet C_{In}))/\|P\|\,\|C\|$

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

3) Clustering method

We extend standard K-means to meet requirements for clustering of web search results as well as to overcome disadvantages of K-means. Our clustering method is as follows:

1) 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*.
2) Define *similarity threshold*
   Similarity threshold is pre-defined to control the process of assigning one page to a cluster. Since similarity is meant to capture the common links shared by different pages, similarity threshold could be easily defined and adjusted.
3) Assign each page to clusters
   Each page *is assigned to existing clusters* when the similarity between the page and the correspondent cluster is above the *similarity threshold.* If none of current existing clusters meet the demand, the page under consideration becomes 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). Preliminary experimental results show that final results are insensitive to the processing order; however, further investigation and proof are needed, which is not discussed here.
4) 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. ***Merge threshold*** is used.

The convergence of the approach is guaranteed by K-means itself since our extension does not affect this aspect. The algorithm described above also has same *time complexity* (*O(nm)*) as 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. Since *m* <<*n*, the proposed approach is in a linear time to the number of URLs/ pages that join clustering procedure. There are mainly two parameters in our approach that may affect quality of final results: *similarity threshold* and *merge threshold.* We have conducted experiments to investigate their effects on the final results.

4. Experiments and Evaluations

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

1) Data collection
   Just as depicted in Figure 1(b), for each topic we not only download specified number of search results but also extract all N out-links and M in-links by AltaVista for all pages in search results.
2) Data cleaning
   In this step, we remove mirrors or duplicates pages in search results since it will mislead clustering process if preserving them. We adopt a non-aggressive method to remove them. Two pages *p* and *q* are said duplicate if (a) they each have at least 8 out-links and (b) they share at least 80% of their out-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.
3) Applying the proposed algorithm to form base clusters
4) Final clusters generation
   Final clusters are generated by recursively 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 unite distinct members of both A and B under the cluster name B.

4.1 Experimental Results

In our first experiment, we investigate the effects of parameters introduced in the proposed approach: similarity threshold and merge threshold. We use the following six query topics in this experiment: Jaguar, Data mining, Java, Jordan, Israel and Salsa. The words for each topic is appeared as keywords for Web Search

Engine Yahoo!. While for topic Jaguar, we have tried different number of pages (200, 400 and 600 pages) as search results. As for other five topics, we use 200 pages as search results for testing. We download the specified number of pages, so totally we have eight collections of dataset. We also extract its out-links and 100 in-links by AltaVista for each pages in the dataset. Here we only consider non-nepotistic links that connect web pages from different websites. We vary similarity thresholds (with four values of 0.2, 0.15, 0.1 and 0.06) and merge thresholds (with six values of 0.8, 0.75, 0.7, 0.6, 0.5, 0.4) to investigate their effects on final clustering results. After data cleaning, about 85% of the web pages is preserved in each dataset for further clustering process.

As final clustering 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. In the rest of discussion, "pages/URLs clustered" means pages or URLs that appear in final clusters whose size is bigger than 3. The size of a cluster is the number of pages in the cluster. We ignore singleton clusters.

Figure 2 is the result of average URLs clustered for 8 datasets for different similarity thresholds. "Jaguar", "Jaguar400" and "Jaguar600" are datasets on topic "Jaguar" with 200, 400 and 600 pages respectively. According to Figure 2, the overall trend for all topics is that the percentage of URLs clustered decreases as similarity threshold increases. For different query topics, this kind of change is sometimes gradual and sometimes sharp. It could be also observed from Figure 2 that for the same topic, just increasing number of pages used for clustering ("Jaguar", "Jagaur400" and "Jaguar600") only slightly affect final percentage of URLs clustered. Table 1 gives the detailed information about final clustering results for six query topics by varying similarity thresholds and merge thresholds. We find from Table 1 that similarity threshold 0.2 is too strict for most topics to get reasonable results although it is totally insensitive to the change of merging threshold. Similarity threshold 0.15 or 0.1 is relatively a good choice since the correspondent percentage of URLs clustered is reasonable and almost insensitive to the variation of merge thresholds. As for similarity threshold 0.06, the increased number of URLs can be clustered but it is sensitive to the change of merge threshold. This could be interpreted in more detail in Figure 3.

Figure 3 (a) is to check the overlap of final clusters for topic "Jaguar" by varying similarity and merge thresholds. According to final clustering results, when fixing the similarity threshold, the number of distinct URLs clustered is almost same but with different overlap for different merge thresholds. That is to say, with smaller merge threshold, pages that originally in different clusters or same pages appeared in more than one cluster are more likely to be merged into one cluster. Just as we could estimate, merge threshold 0.8 is of highest overlap while 0.4 is of the lowest one. As for vertical comparing, similarity threshold 0.2 is of the lowest overlap and 0.06 is of the highest one. The low overlap obtained by decreasing merge threshold will decrease the "purity" of final cluster, which is proved in Figure 3(b). Figure 3(b) is the average entropy for "Jaguar" of similarity 0.1 for different merge thresholds. It shows that merge thresholds bigger than 0.7 are good choices. So in conclusion, similarity threshold is more influential than merge threshold on the quality of final clustering results. For most topics, similarity threshold 0.1 or 0.15 and merge threshold 0.7 or 0.75 is our recommendation. This is also hold for "Jaguar400" and "Jaguar600" that are shown in Figure 2.

Table 2 shows the semantic impression of final clustering results for the six topics, eight datasets. We list main meaningful groups whose size is bigger than 3 for six topics (with 200 pages) and whose size is bigger than 5 for "Jaguar400" and "Jaguar600". According to Table 2, the proposed approach does discern some medium but semantically meaningful groups around the main idea about the query topic, which is very useful and helpful for end user to identify the idea of the topic on the Web. E.g. for topic "Jaguar", in addition to discern the main idea like "car", "cat", the clustering results also identify sub-topics like "club", "magazine", "Game" and "touring place" etc.

One phenomenon observed is that some small clusters produced by the proposed approach are semantically very similar and should be in one group from a more general point of view. We think this could be solved by introducing hierarchical clustering to make the final clustering results more natural and easy to interpret.

4.2 Evaluations

Validating clustering algorithm and evaluating its quality is complex because it is difficult to find an objective measure of quality of clusters. We would like to use three metrics *precisions*, *recall* and *entropy* to evaluate quality of final clusters. We manually check 200 web pages for each of six topics and mark each one page with "relevant" or "irrelevant" to indicate whether it is relevant to the corresponding query topic. Precision and recall are defined as follows:

Precision=number of URLs that are both clustered and 'relevant' marked / number of URLs clustered (1)

Recall=number of URLs that are both clustered and 'relevant' marked /number of 'relevant' marked URLs (2)

Entropy provides a measure of "goodness" or "purity" for un-nested clusters by comparing the groups produced by the clustering technique to known classes. Small entropy value of the cluster indicates its high intra-cohesiveness while big entropy value means that its members are not tightly related and focus on different sub-topics under the same general topic. 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: relevant (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 [9]: Let CS be a cluster solution and E (j)= $-\sum_i p_{ij}\log(p_{ij})$ is the entropy for each cluster j. 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 entropy 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. Basing on the three metrics defined above, we evaluate the proposed link-based clustering with different similarity thresholds. The results are depicted in Figure 4 (a) to (c).

4.3 Link-based Clustering verse Snippet-based Clustering

In order to compare the proposed approach with other clustering algorithms, we implemented STC algorithm proposed in [7] (also see explanation in Section 2.1) that is based on snippet attached with each URL in search results.

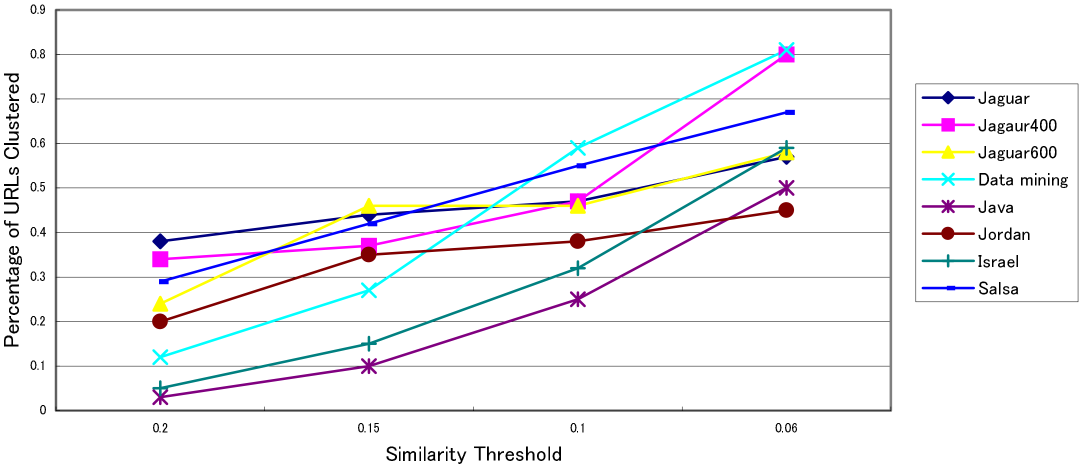


Figure 2. Effects of similarity thresholds on URLS clustered (with cluster size>3)

	0.2	0.15	0.1	0.06
0.8	7/78	8/91	8/98	9/121
0.75	7/78	8/91	7/95	9/121
0.7	7/78	7/85	7/95	9/121
0.6	7/78	7/85	6/93	9/121
0.5	7/78	7/87	6/93	8/116
0.4	7/78	7/87	6/93	7/108

(a) Jaguar

	0.2	0.15	0.1	0.06
0.8	5/25	9/56	17/140	22/225
0.75	5/25	9/56	15/135	18/208
0.7	5/25	9/56	15/135	18/206
0.6	5/25	9/54	12/126	16/192
0.5	5/24	9/54	9/118	13/168
0.4	5/24	9/54	9/118	12/162

(b) Data mining

	0.2	0.15	0.1	0.06
0.8	1/6	2/10	8/50	9/100
0.75	1/6	2/10	8/50	9/100
0.7	1/6	2/10	8/50	9/100
0.6	1/6	2/10	8/50	9/100
0.5	1/6	2/10	8/50	9/102
0.4	1/6	2/10	8/50	9/102

	0.2	0.15	0.1	0.06
0.8	7/41	7/80	5/82	7/98
0.75	7/41	6/75	5/82	7/98
0.7	7/41	6/75	5/82	7/98
0.6	7/41	6/75	4/78	6/95
0.5	7/41	5/70	3/76	5/89
0.4	7/41	5/70	3/76	5/89

(c) Java

(d) Jordan

	0.2	0.15	0.1	0.06
0.8	1/6	3/20	5/60	11/138
0.75	1/6	3/20	5/60	11/138
0.7	1/6	3/20	5/60	10/133
0.6	1/6	3/20	5/60	9/131
0.5	1/6	3/20	5/60	9/131
0.4	1/6	3/20	5/60	8/118

(e) Israel

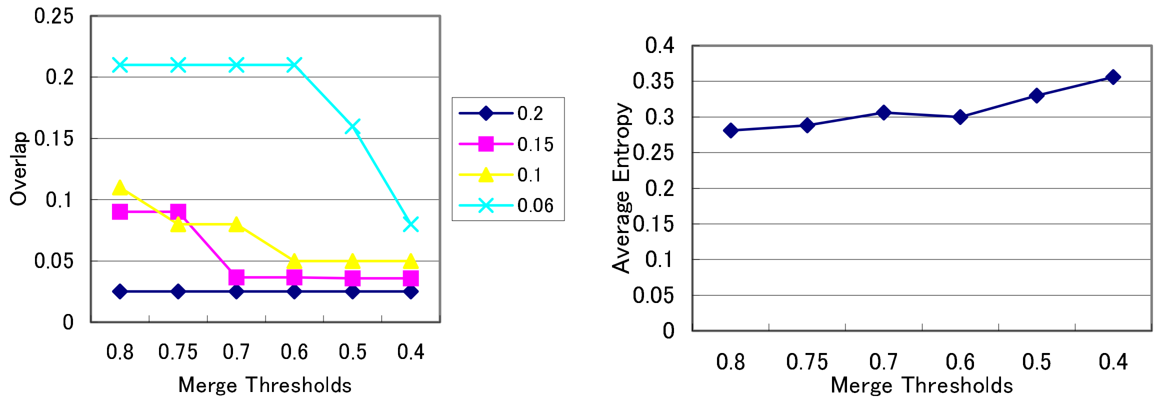
	0.2	0.15	0.1	0.06
0.8	6/62	7/83	12/153	13/165
0.75	6/62	7/83	11/150	11/156
0.7	6/62	7/83	11/150	11/156
0.6	6/62	7/83	9/135	10/153
0.5	5/59	7/86	7/110	8/140
0.4	5/59	7/86	7/110	6/134

(f) Salsa

Table 1. Entry in each table is Number of clusters /total number of web pages in these clusters for different similarity thresholds and merge thresholds for six topics of testing.

Subtopics (Cluster Size>3)	Jaguar	Jaguar400 (size >5)	Jaguar600 (size>5)	Data mining	Java	Jordan	Israel	Salsa
1	Car	Car	Car	Technical support	Programming support	Hashemite kingdom	Government Info. for Israel	Music /Dance
2	Club	Club	Club	IBM Research	Coffee	Player: Michael Jordan	Embassy info. Of Israel	Hot sauce
3	Game	Game	Game	Research in Uni.	Game	Writer: Robert Jordan	Tourism info of Israel	Club
4	Magazine	Magazine	Magazine	Magazine /Publication	Research of Java Security in Uni.	Tourism info.	News/ Magazine	Salsa in Germany
5	Big Cat	Big Cat	Big Cat	Software /product support	Magazines	Links	Arts info.	Links to salsa recipes
6	Atari Emulation	Atari Emulation	Atari Emulation	KDD	IBM /Java			Cookbook/ Recipe
7	Links	On-line Sale	On-line Sale	Data mining group	Sun /Java			Salsa events
8		Racing Car	Racing Car	Conference/ Workshop	Links			
9		Links	Touring Place	Links				
10			Research project					
11			Links					

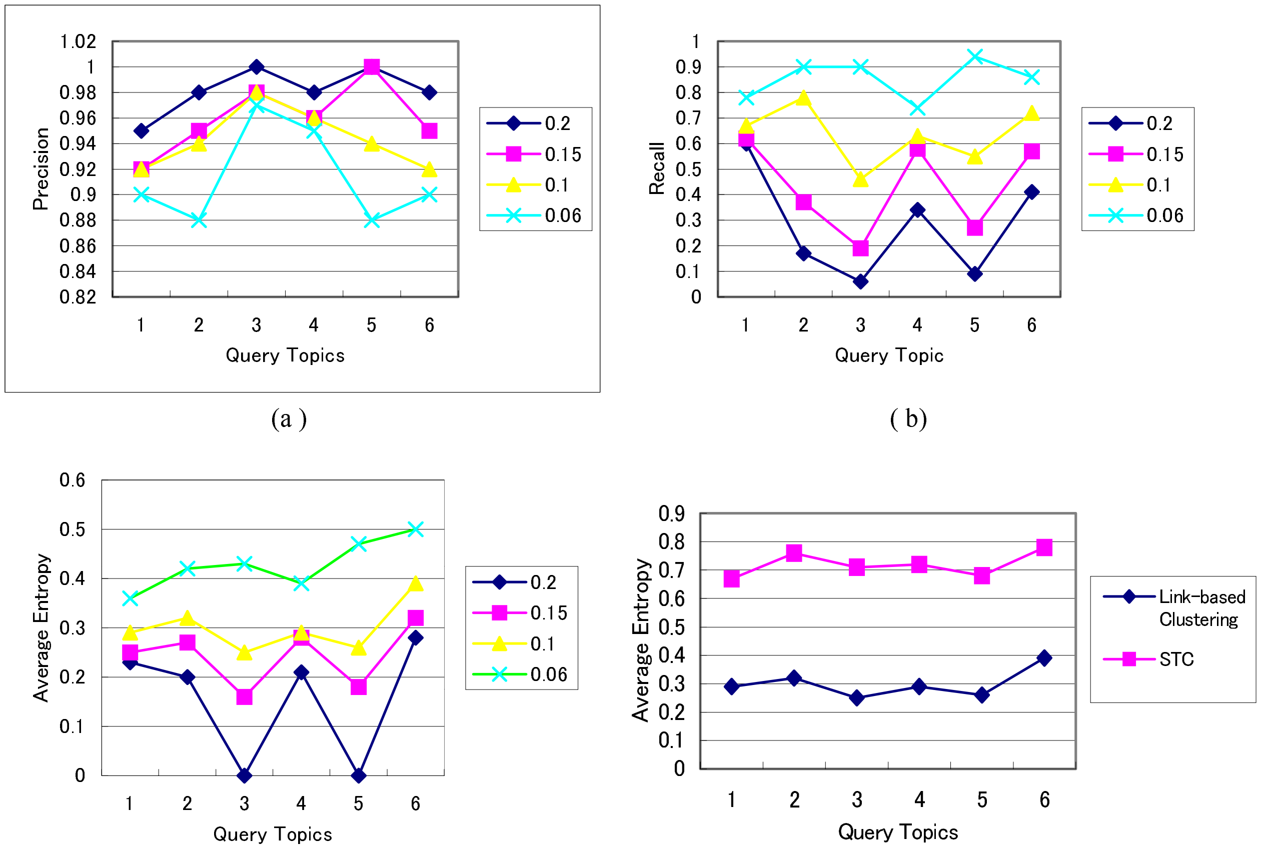
Table 2. Main subtopics for each of six query topics of testing



(a)

(b)

Figure 3. (a) The overlap of final clusters for topic "Jaguar" with different similarity and merge thresholds, (b) the average entropy of final clusters for Topic "Jaguar" with similarity threshold 0.1 with different merge thresholds.

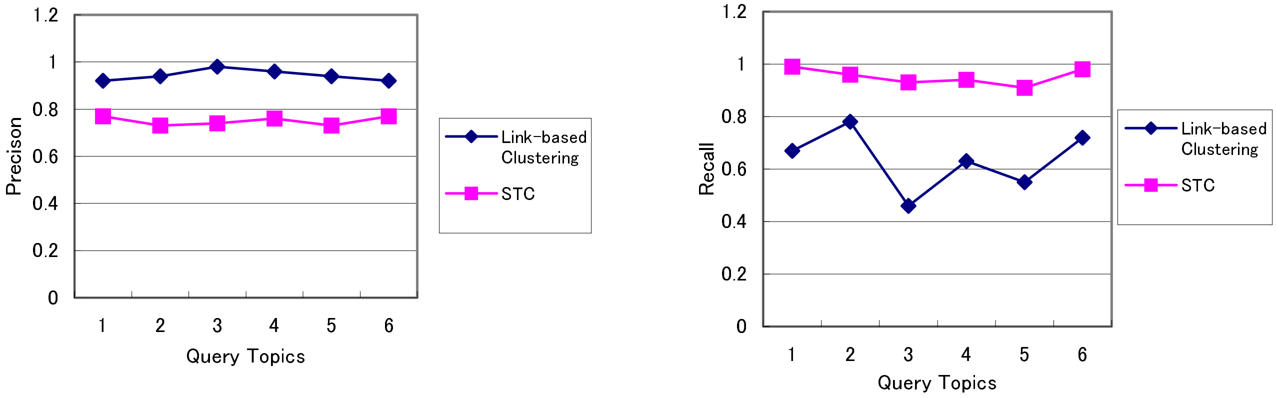


(a)

(b)

(c)

(d)



(e) (f)

Figure 4. Comparison based on the metrics of precision, recall and entropy for six topics. Figure (a),(b) and (c) are comparisons of the link-based clustering with different similarity thresholds (merge threshold is 0.75). Figure (d),(e) and (f) are comparisons of link-based clustering (similarity threshold 0.1 and merge threshold 0.75) with STC algorithm (merge threshold 0.5).

We conducted experimenting on STC algorithm for the six topics. We extract six collections of 200 snippets and apply STC algorithm to them. Just as stated in [7] that we use 0.5 as merge threshold to merge base clusters. We did not try different merge thresholds since it is stated that STC algorithm is insensitive to the variation of merge thresholds. The detailed comparisons are depicted in Figure 4 (d) to (f).

The query topics numbered from 1 to 6 as horizon axle in Figure 4 is in the same order with query topics mentioned in the beginning of Section 4. This order is also hold for all discussions in the paper. By fixing merge threshold with 0.75, we investigate the quality of final clustering results for six topics with different similarity thresholds. According to Figure 4(a), similarity threshold 0.2 gives the highest precision and with similarity threshold decreases, the precision decreases accordingly. The general precision value for all topics is very high and around 0.9. Just as depicted in Figure 4(b), reverse to precision, correspondent recall increases as similarity threshold decreases. However, the general recall value is relatively low and changes greatly between 0 and 1 as we vary similarity thresholds. Entropy comparison in Figure 4(c) shows that 0.2 produces "pure" clusters since it produces a few clusters with medium size and members of each cluster are tightly related. Entropy value is also monotonic increasing with similarity threshold decreases for all six topics. What we summarized from these comparisons is consistent with our recommendation mentioned in section 4.1 that similarity threshold between 0.15 and 0.1 might be a good choice.

We compared the final clustering results produced by link-based approach with snippet-based STC algorithm on the three metrics. We choose 0.1 and 0.75 as similarity and merge threshold respectively in links-based clustering. The results are depicted in Figure 4(d) to (f). According to the final clustering result produced by STC algorithm, it just discerns the main group, which usually includes most pages in dataset as well as several very small groups that include just 3 or 4 pages. It fails to identify some medium, but meaningful groups around the main topic. This result lead to very high entropy, as depicted in Figure 4(d), which means the main group is not so "pure" and some pages in it should be separated and grouped into more cohesive clusters. Since the six topics for testing are quite general, the marked "relevant" pages according to manually check cover around 60% to 70% of total 200 pages. This could explain the high precision value for both link-based approach and snippet-based approach shown in Figure 4(e). As for recall, clustering results produced by STC are of high recall value since most URLs in search results are clustered and most of URLs clustered are in one cluster. The comparison of recall is depicted in Figure 4(f). We think that one possible reason that STC algorithm works poor under our experimenting environment might be that the query topics for testing are quite general (one word or two words) while the advantage of STC is to capture the relationship and order of the words appeared as keywords.

5. Conclusion

In this paper, we propose a new link-based clustering approach to cluster web search results by exploring both co-citation and coupling analysis. Our goal is to cluster high quality pages (by filtering some irrelevant pages) in search results returned from web search engine for a specific query topic into semantically meaningful groups to facilitate users' accessing and browsing. We also extend standard K-means algorithm to overcome its disadvantages to make it more natural to handle noises. In order to get in-depth understanding about effectiveness of the proposed approach, we carry out experiments on six different query topics: Jaguar, Data mining, Java, Israel and Salsa by varying different values similarity threshold and merge threshold. We also tried different number of search results. We implemented STC algorithm proposed in [7] that is based on snippets attached with each URL in search results and applied it to six collections of 200 snippets. Evaluations and comparison are based on three metrics: precision, recall and entropy. Experimental results suggested that similarity threshold around 0.1 or 0.15 and merge threshold around 0.75 or 0.7 might be good choices for link-based clustering to generate reasonable clusters. The experimentation and evaluation indicate that on the average, link-based clustering works better than snippet-based clustering (STC).

While recall of final clusters produced by the proposed approach is relatively low, how to improve recall without sacrificing precision and entropy is our next-step work. We would like to extend our work by combining word processing and link analysis and introduce some heuristic rules to remove noise links to improve final clusters quality.

Reference:

1. **Kleinberg 98** Jon Kleinberg. *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** Sergey Brin, and Larry Page. *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*
9. **Ron Weiss *et. al.* 96** *Hypursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering* Hypertext'96 Washington USA
10. **Michael Steinbach, George karypis and Vipin Kumar** *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 Conference on Research and Development in Information Retrieval. 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** Krishna Bharat, and Monika Henzinger. *Improved algorithms for topic distillation in hyperlinked environments*. In Proceedings of the 21st SIGIR conference, Melbourne, Australia, 1998.
16. **Chakrabarti *et. al* 98** Soumen Chakrabarti, Byron Dom, David Gibson, Jon Kleinberg, Prabhakar Raghavan, and Sridhar Rajagopalan. *Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text*. Proceedings of the 7th World-Wide Web conference, 1998.
17. **Florescu, Levy and Mendelzon 98** Daniela Florescu, Alon Levy, Alberto Mendelzon. *Database Techniques for the World-Wide Web: A Survey.* SIGMOD Record 27(3): 59-74 (1998).
18. **Gibson, Kleinberg and Raghavan 98** David Gibson, Jon Kleinberg, Prabhakar Raghavan. *Inferring Web communities from link topology.* Proc. 9th ACM Conference on Hypertext and Hypermedia, 1998.
19. **Agrawal and Srikant 94** Rakesh Agrawal and Ramakrishnan Srikanth. *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