AUTOMATIC TOPIC DISCOVERY FROM HYPERLINKED TEXT

Kuo Jui Wu
Institute of Information Science
Academia Sinica, Taiwan
wugray@iis.sinica.edu.tw

Meng Chang Chen
Institute of Information Science
Academia Sinica, Taiwan
mcc@iis.sinica.edu.tw

ABSTRACT

Topic discovery is an important means for marketing, e-Business, social science study and many other applications for various purposes, such as identifying a group with certain properties, observing the raise and diminishment of a certain group. The explosively growing of Internet makes automatic topic discovery a must for the task. In this paper, first we propose the TGM method to rank the eigenvectors of the Web hyperlinked matrix and their associated topics. Then we propose the ATD method, which combines a clustering algorithm with a conventional principal eigenvector computation method, to identify the topics relevant to a given query without manual examination. Our extensive experiments show the ATD method performs very well, and beats TGM in terms of computation time and topic discovery quality.

Keyword: topic discovery, hyperlink analysis, authority and hub

1 INTRODUCTION

With the explosive growth of the World Wide Web, information discovery from the Internet becomes difficult. The emergence of large portals and search engines are designed to help users to find their information needs. Directory services, such as Yahoo [16], are like “Yellow page” of Internet document (or called *Web pages* that both terms are interchangeable in this paper) collected and categorized by human. Users browse through the directories and click the items they are interested in the Internet resource lists. The documents collected in the directories are only part of the whole Internet documents, and most of their collected pages are already well known and popular. Thus the information discovery from the Internet using Directory services is limited to their collection.

Another Internet document collectors are search engines like AltaVista [15]. Search engines use robots or spiders to crawl the Internet and index the contents of documents in their collections. When receiving queries from users, search engines lookup their databases and present users a list of matched documents ordered by similarity scores between page contents and queries. Users submit queries composed of keywords to search engine and usually receive a long list of URLS of matched documents. After several clicks on the top few URLs, users may be discouraged because of retrieving many irrelevant or non-interesting results.

The problem of search engines is that it often returns too many results, and the ranking of the documents may not meet user’s expectation. In addition, many search engines only return with a set of individual documents. The Internet documents returned from search engine may contain several topics about the given query. For example, the results of query “jordan” may contain topics such as “Michael Jordan”, “The F1 Jordan Grand Prix Team”, “Jordan Middle School”, and “Country: Jordan”, etc. It is obvious that the documents from different topics represent different interests. For marketing, it is critical to separate the documents of different topics and identify their associated topics. Furthermore, it is sometime important to rank the documents according to their importance to the specific topic. In some cases, a topic may represent a cyber community, such as “Michael Jordan fans club”, that is of great interest to many e-Business applications. Periodical scanning the Internet to discover new cyber communities has become a common practice for many e-Business related applications.

Internet documents are associated via hyperlinks. When Internet document authors prepare the documents, hyperlinks are added for various purposes, such as reference, related page, etc. It is pointed out that Internet documents link to relevant pages more often than non-relevant pages [5]. For example, if Web page h_1 is about NBA basketball games and h_1 links to Web page h_2, h_3, and h_4, then probably h_2, h_3, and h_4 are also NBA related pages. While the links may have different intensions, cross-references among documents as a whole provide useful information of their underlying associations. Topic discovery is an emerging technology that can be applied to resolve the problems. For instance, Kleinberg proposes an algorithm named HITS (Hyperlink-Induced Topic Search) [8] that the major concept is the authority endorsement and conferral. That is, links between Web pages contain latent human judgment that people write Web pages and create links intentionally. One application of HITS algorithm proposed by Kleinberg is to re-rank the results returned from search engine based on link information among Web pages.

In order to find out other topics, Kleinberg suggests calculating all the eigenvectors of the hyperlink matrix that other topics may be represented by the non-principal eigenvectors. But which non-principal eigenvector contains an interesting topic cannot be determined automatically. To alleviate the problem, we propose an algorithm partitions the web pages into clusters, and for each cluster we run a conventional principal eigenvector computation algorithm to find out the representing vector. In this way, we can automatically discover the interesting topics for a given query and rank each topic according to their authority.

The remainder of this paper is organized as follows. First, we describe a preliminary survey of current techniques for clustering and classification for Web pages. Next, we describe the heart of the paper, the ATD algorithm. Following is the experiment setup, results, and discussions. We give some conclusions and future work in the final part.

2 DOCUMENT CLUSTERING AND CLASSIFICATION

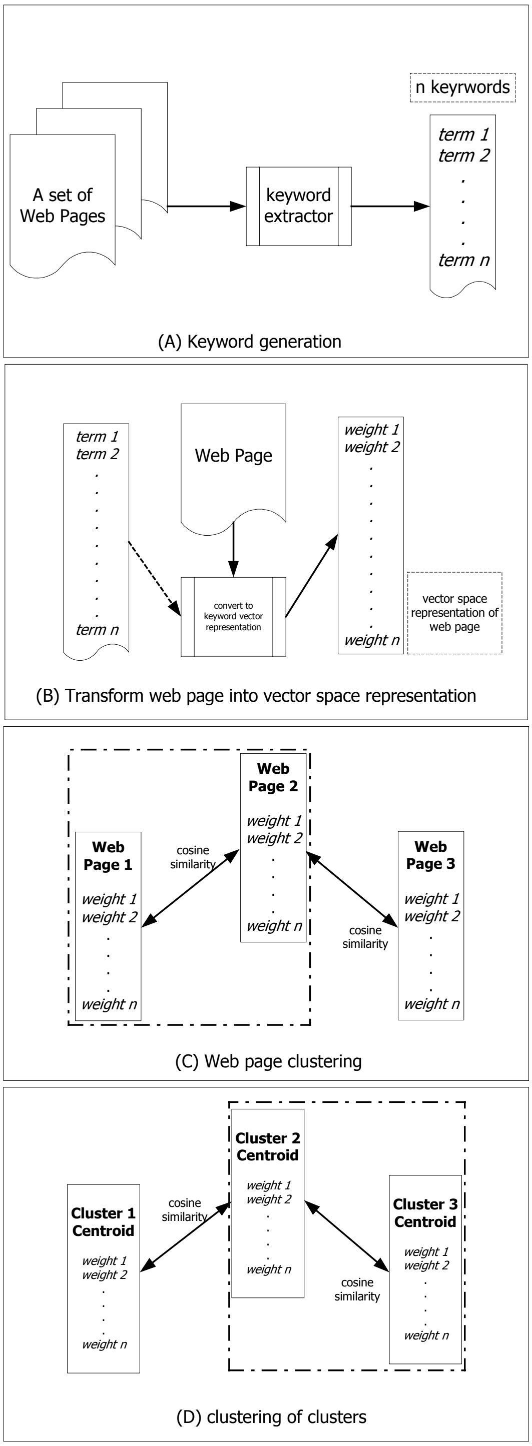
There are two popular categories of approaches for clustering and ranking of Web pages. One is content-based method that is based on the content similarity of Web pages. The other category is hyperlink-based method that uses the hyperlinks, and the associated information such as link types, counts, and topology, among Web pages to cluster and rank Web Pages.

Content-based Method

A simple content-based clustering works as follows. Web pages are represented by **vector space model** [13]. Let S be the set of all Web pages. As shown in figure 1(a), assume that there are n terms (t_1 to t_n) after keyword extraction of S. Then for each Web page $h_i \in S$, the vector space model representation v_i is a vector with n element. The j_{th} element of the v_i is the weight of term t_j in h_i. The weight can simply be 1 when term t_j appears in h_i, and 0 when the term is absent from h_i. Moreover, in a sophisticated way, the weight can be the term frequency of term t_j in h_i, etc. The v_i is represented as $v_i=\{w_1, w_2, \ldots, w_n\}$ as shown in figure 1(b). After transforming $\forall h_i \in S$ into v_i, the cosine similarities between Web pages in S, $\forall h_x, h_y \in S, sim(h_x, h_y) = \left|v_x \cdot v_y\right| / \left|v_x\right|\left|v_y\right|$ are computed. A cluster can also be represented as a centroid vector in the vector space model. The j_{th} element of a centroid vector is the average of the j_{th} element of vector space model representation for all pages in the same cluster. Similarity score of two clusters is the cosine similarity of the centroids. For each run of clustering, each pair of pages remaining in S with the highest similarity score is grouped together. The two steps are shown in figure 1(c) and 1(d). The clustering process will repeat until the required number of clusters or other criterion is reached. The content-based ranking is first transformed the query into vector representation. Then the cosine similarities between query and Web pages are measured. According to the similarity scores, ranking of the Web pages are determined.

Hyperlink-based Method

Hyperlink-based method exploits the hidden knowledge embedded in the hyperlinks among Web pages. First, the hyperlinks in the Web pages are transformed into Web graph that Web pages are regarded as nodes and the links are edges in the Web graph. Clustering Web graph based on link information is a graph-partitioning problem. There exist many kinds of structures in the Web graph, such as cores [7], Web rings and directed trees. Clustering of Web graph is to find these structures and isolate them by removing minimal edges that link to and from other clusters. That means nodes in a cluster are tightly connected, while clusters are loosely connected. It is likely that each cluster forms a topic because links in the cluster as a whole may represent some implicit intensions of the authors. In each cluster, Web pages are ranked by their link counts, types, or authority values [8]. Previous works of [1], [2], [3], [4], [6], [7], [9] and [10] propose several methods that cluster and rank the Web pages for ranking Web pages, discovering cyber communities, and finding related Web pages.



(A) Keyword generation

(B) Transform web page into vector space representation

(C) Web page clustering

(D) clustering of clusters

Figure 1: Content-based clustering and classification

3 TOPIC DISCOVERY

Kleinberg proposes that every Web page has two properties: authority and hub. Authority attribute of a Web page is the representative and authoritative of the topic corresponding to its content and to the pages it links. Hub attribute is that the page itself contains links to other pages, which are representative and authoritative of the same topic. Thus, a good authority page is pointed by good hub pages, and a good hub page points to good authority pages, as shown in figure 2. This is so called the mutually reinforcing relationship. In addition, a topic should be composed of Web pages with good authority and hub values. For the Web page ranking in a topic, the work of Amento et al. [1] has shown that hyperlink can be exploited to rank Web pages in terms of authority and representative. In their experiments, they find that hyperlink based algorithms, such as HITS, in-degree, etc. are capable of identifying high quality Web pages. In the topic discovery, page ranking is a useful feature that can be applied to many applications.

Topic Induction

To exploit the above theory to discover topics, first we have to prepare a collection of Web pages, called *base set* (or vicinity graph). There are several criteria for a base set. First, base set should be broad enough to cover the desired topics. Second, the size of base set should be not too large; otherwise, computation cost is the penalty. It is a good start to construct a base set from collecting the Web pages containing the keywords in the query. Content-based search engines can help to provide such a collection of Web pages for a given query. The HITS (Hyperlink-Induced Topic Search) algorithm proposed in [8] exploits an existing content-based search engine to retrieve the search results of a given broad topic query. Then it fetches these Web pages and employs only the hyperlinks in the pages. HITS has a pre-processing step to construct a base set. First the top 200 Web pages of the search results are chosen to form a root set. The “200” is a tunable parameter. Next HITS expands the root set into a base set according to the link information by the following steps. (i) Pages that are pointed by any pages in root set are added into base set. (ii) Web pages which link to any pages in the root set are added into base set. If there are more than 50 pages that link to a single page in the root set, only randomly chosen 50 pages are added. The “50” is also a tunable parameter. (iii) The root set is added into the base set. Figure 3 shows the base set construction process.

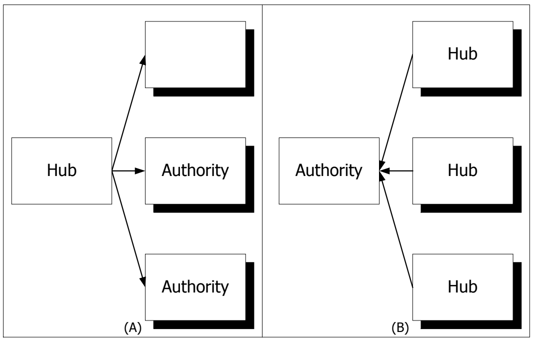


Figure 2: Mutually reinforcing relationship

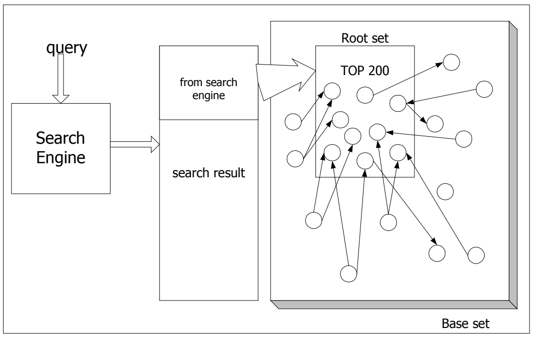


Figure 3: Base set construction

After the base set is constructed, the link adjacency matrix A is built. The link adjacency matrix A is an $n*n$ matrix. The Web pages in the base set are numerated from 1 to n to build the matrix where n is the size of base set. If Web page i has a link to Web page j, then (i, j) element of the matrix A is set to 1, otherwise, (i,j) is set to 0. For example, suppose there are 3 Web pages, namely h_1, h_2, and h_3, in the base set. Web page h_1 links to h_2 and h_3, h_2 links to h_3, and h_3 links to h_1, then the adjacency matrix A is built as

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Following we describe the iterative version of HITS algorithm. It tells how to calculate authority and hub score of each Web page.

1. Initial value of authority and hub score for each page is set to 1.

2. For each page, the new authority score is the sum of the hub scores of the Web pages that link to it (as shown in figure 2.B).

$$Auth_i = \sum_{j:(j,i)\in E} Hub_j \quad (1)$$

(j,i) represents a hyperlink which initiated from page j and links to page i, and E is the set of all hyperlinks in the base set.

3. For each Web page, the new hub score is the sum of the authority scores of the Web pages that it links (as shown in figure 2.A).

$$Hub_i = \sum_{j:(i,j)\in E} Auth_j \quad (2)$$

4. Normalize the authority and hub scores of the Web pages.

5. Repeat step 1 to 4 until the scores are converged, or a pre-defined number of repeats is reached.

Kleinberg et al. also propose a linear algebra version of HITS algorithm to calculate the authority and hub scores. The step 2 and step 3 of HITS algorithm can be written as $Auth = A^T \cdot Hub$ and $Hub = A \cdot Auth$, the authority scores of each Web page (i.e. $Auth$) are correspond to the principal eigenvector associated with largest eigenvalue of matrix A^TA, and the hub scores (i.e. Hub) are correspondent to the principal eigenvector of matrix AA^T, where A is the link adjacency matrix. After calculating authority and hub scores of Web pages in the base set, HITS ranks the pages according to the authority scores and presents the top N results to user. So the ranking is ordered by the authoritative of the topic for the given query.

HITS discovers and ranks the topic with the most strongly connected pages. Other topics may exist in the base set that each is associated with a non-principal eigenvector of matrix A^TA or AA^T. While all the eigenvectors can be obtained by using any eigenvector computing algorithms, it is not clearly which non-principal eigenvector is associated with a meaningful topic. Recent work of Davison et al. [4] uses a fast eigenvector solver to calculate the eigenvectors. They demonstrate that by looking at the top 1/4 eigenvalues and the associated eigenvectors, they can find clusters of Web pages that are more interesting than the one extracted by the principal eigenvector. But they are still unable to determine which eigenvector is associated with the most interesting topic or locate most of the interesting topics.

Here we define a ***Topic Goodness Metric*** (***TGM***) to measure the quality of a discovered topic. With each eigenvector *x*, let *TGM(x)* be defined as

$$TGM(x) = \left|\underset{i\in T_{ax}}{Auth_x(i)}\right| + \left|\underset{i\in T_{hx}}{Hub_x(i)}\right| \quad (3)$$

where web pages $i \in T_{ax}$ is the top *k* authorities of eigenvector x, and $Auth_x(i)$ is the associated authority value (similarly for $i \in T_{hx}$ and $Hub_x(i)$). The reason behind the definition of TGM is that the authorities and hubs associated with eigenvectors of larger absolute value will typically be densely linked in the vicinity graph, and will probably have more concrete meaning. By setting threshold of minimal TGM value, we can find several interesting topics associated with non-principal eigenvectors and discard the non-prominent ones. We will use TGM function to discover topics to demonstrate this point of view. We call the method of using TGM value to rank the topics associated with eigenvector as TGM method for simplicity reason.

Automatic Topic Discovery (ATD) algorithm

We propose an automatic topic discovery algorithm called ***ATD*** algorithm, which is composed of a content-base method, a clustering algorithm and a principal eigenvector computation algorithm The idea behind our algorithm is to isolate each strongly inter-connected cluster in the base set before calculating the principal eigenvector of each cluster. First, ATD consults a content-based search engine for a given query, and then construct the base set from the query return. In ATD, the base set construction is the same with the procedure in Figure 3. Then we use the proposed **A-H-A** clustering algorithm, described below, to extract clusters from the base set.

1. Initially, let *k*=1. Repeat step 2, 3, 4 and 5 until no more cluster is found.

2. Step **A**: Let $Cluster_k$=NULL. For the node *O* with maximum out-degree in the base set, find the node *C* with maximum in-degree linked by *O*. Node *C* is added into $Cluster_k$ and removed from the base set.

3. Step **H**: Add nodes that link to *C* into $Cluster_k$ and remove from base set.

4. Step **A**: Add nodes linked by nodes added in step **H** into $Cluster_k$ and remove from base set.

5. Let *k*=*k*+1.

Figure 4 shows one iteration of A-H-A algorithm. A-H-A finds clusters by following the underlying link topology of Web pages in base set. To some extent, the Web page with large out-degree can be regarded as a hub candidate. Similarly, Web page with large in-degree can be regarded as a authority candidate. So the first step of A-H-A clustering algorithm (Step **A**) finds the node *C*, which is considered as a centroid authority of the topic. This centroid can trawl in the hubs of the topic in the second step (Step **H**) because good hub links to good authority. In order to form a complete topic, A-H-A clustering algorithm trawls in other authorities in the third step (Step **A**). So this clustering algorithm conforms the relationship between authority and hub. A cluster is discarded if number of nodes in this cluster is less than a predefined threshold value. After the base set is partitioned into several clusters, we find the principal eigenvector of each cluster and present these topics, their members Web pages and ranks to user.

In order to remove noise in the Web pages, the A-H-A algorithm will discard the hyperlinks with the following properties during clustering process.

- The source and destination are in the same domain: Many of this kind of hyperlinks serve only for navigational purpose, such as “back” and “click here to go to the main menu”. These kinds of hyperlinks have no contribution to the topic. In addition, a good topic should be formed by the Web pages from many different authors. If there exist hyperlinks from same domain, pseudo topics may be discovered and the ranking of pages in the topics will be influenced.

- The destination of a hyperlink is a large portal site, i.e. Yahoo!, or popular, general-purpose sites. The destination pages are irrelevant to almost any topic, but many Web pages contain these links.

Moreover, URL with destination such as *host.domain/**copyright.html*** or *host.domain/**privacy.html***, etc., should be removed too. We use an URL stop-list to remove these hyperlinks.

The complete workflow of automatic topic discovery is illustrated in figure 5. First a broad topic query is sent to a content-based search engine. The content-based search engine lookups its index against the query and return a list of matched documents. Based on the search results, the page contents are examined to obtain link information. Then an expanded focused vicinity Web graph composed of relevant Web pages and hyperlinks is built. By following the underlying hyperlink topology and relationship between authority and hub, the Web graph is partitioned into several inter-connected clusters relevant to the broad topic query. These clusters are ranked by any an eigenvector computing algorithms and become the topics discovered by our algorithm.

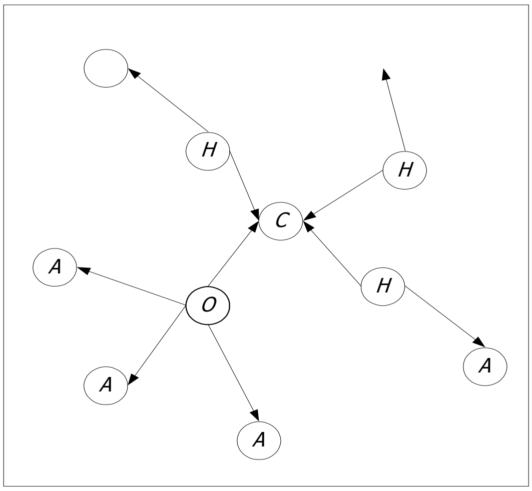


Figure 4: A-H-A clustering

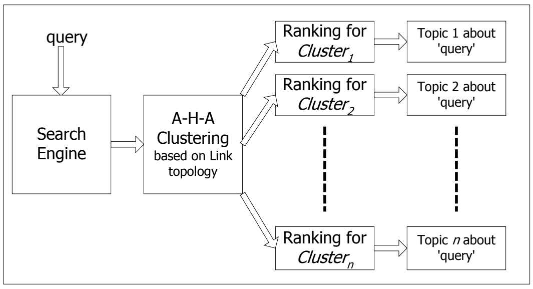


Figure 5: Automatic topic discovery

4 EXPERIMENTS AND DISCUSSION

In the experiments we use the AltaVista search engine as the content-based search engine to submit queries. The base set is built by the following parameters. Top 200 of the search result are placed into root set. When expanding the root set into base set, we let each page in the root set trawls in at most 50 Web pages. While running A-H-A clustering algorithm of ATD, the minimal cluster size is set to 20 Web pages. As for TGM, we include the top 10 eigenvectors (non-principal eigenvectors may contain two potential topics, laid in positive and negative ends of the eigenvector), and only measure top 20 authorities and hubs of each eigenvector (potential topic). In this section we show two experiments using ***TGM*** and ***ATD*** for the same base set of the query “jaguar”.

Experiment TGM

In this experiment, we use an Intel Pentium III 550Mhz PC running FreeBSD and the gsl-0.7 library to calculate the eigenvectors of the matrix. Before running TGM method, the hyperlinks and Web pages in the base set are checked. Web pages of the URL stop-list and hyperlinks which come from and link to the same domain are removed. The computation time in this experiment is over 20 minutes. Table 1 and table 2 shows the two topics, jaguar car and atari jaguar games, and their TGM value.

Table 1: Jaguar cars – from 2nd non-principal eigenvector, positive end

TGM value	6.03921
TOP 1 HUB	http://www.webfocus.co.nz/jaguar/
TOP 1 AUTHORITY	http://www.jaguarmagazine.com/
TOP 2 AUTHORITY	http://www.collection.co.uk/
TOP 3 AUTHORITY	http://www.jec.org.uk/

Table 2: Atari jaguar games – from principal eigenvector

TGM value	5.13644
TOP 1 HUB	http://atarihq.com/interactive/
TOP 1 AUTHORITY	http://jaguar.holyoak.com/
TOP 2 AUTHORITY	http://songbird.atari.net/
TOP 3 AUTHORITY	http://members.aol.com/atarijag/

We examined the discovered topics ranked by TGM value and found that topics with smaller

TGM value may be a subset of these the topics with higher TGM values. Note that the maximal topic size is set to be 40 (at most 20 authorities and 20 hubs) in this experiment and the pages with authority (and hub) value of 0 will be excluded. Thus there are only two qualified topics.

Experiment: ATD algorithm

In this experiment we use an Intel Pentium III 550Mhz PC with Windows 2000 to run the algorithm implemented by Microsoft Visual C++. The computation time is less than 10 seconds. There are three topics, jaguar cars, NFL jaguar team, and atari jaguar games, in table 3, 4, and 5. Their size is 65, 29, and 24 Web pages respectively. We can see that ATD algorithm discover 1 more topic than the TGM method.

Table 3: Jaguar cars

TOP 1 HUB	http://www.xks.com/
TOP 1 AUTHORITY	http://www.jaguarcars.com/
TOP 2 AUTHORITY	http://www.jagweb.com/
TOP 3 AUTHORITY	http://www.classicjaguar.com/

Table 4: NFL jaguar team

TOP 1 HUB	http://www.macjag.com/
TOP 1 AUTHORITY	http://www.footballfanatics.com/football.taf?partner_id=9
TOP 2 AUTHORITY	http://www.nflfans.net/afccentral/jaguars/
TOP 3 AUTHORITY	http://jaguars.jacksonville.com/

Table 5: Atari jaguar game

TOP 1 HUB	http://atarihq.com/
TOP 1 AUTHORITY	http://jaguar.holyoak.com/
TOP 2 AUTHORITY	http://www.telegames.com/
TOP 3 AUTHORITY	http://songbird.atari.net/

Comparison

Following we do an evaluation of topic relevance for different topic discovery method, ATD and TGM. We recruited a group of ten subjects to rate the topics. The rating has 3 choices, where "+" stands for relevance, "-" stands for non-relevance, and "*" for relevance and the same with formerly rated topics. (Recall that some topics with smaller TGM value are subset of topics with larger TGM value.) Because there are many different aspects or meanings of broad-topic queries, the subject is told to rate the topics relevance according to his own definition of relevance between topic and query in mind. We choose the minimal threshold of TGM to 4.0 to obtain topics for evaluation. The evaluation contains two tests, queries of 'jaguar' and 'japan'. Table 6 to table 9 shows the results.

Table 6: Topic relevance – TGM, "jaguar"

Topic	1	2	3	4
TGM value	6.04	5.14	4.78	4.63
Relevance	+	+	*	-

Table 7: Topic relevance – ATD, "jaguar"

Topic	1	2	3
Topic size	65	29	24
Relevance	+	+	+

Table 8: Topic relevance – TGM, "japan"

Topic	1	2	3	4	5
TGM value	5.17	4.85	4.80	4.39	4.06
Relevance	+	-	+	+	*

Table 9: Topic relevance – ATD, "japan"

Topic	1	2	3	4
Topic size	51	25	27	25
Relevance	+	+	+	+

The TGM method can be used to discover topics, which are densely linked sub-graph in the base set. But some of these densely linked sub-graphs are irrelevant to query or a subset of a larger sub-graph. By using A-H-A clustering algorithm and minimal topic size threshold, these topics will not be enumerated, or will be discarded in ATD algorithm. Unlike TGM method, the topics discovered by ATD algorithm are distinct topics. The computation of ATD algorithm is much faster than TGM method because ATD operate on the small- sized topic rather than the whole base set.

Discussion

In the experiments, the query contains only one keyword for demonstration purpose. In real world applications, a query can contain more keywords in order to accurately describe the user interest, and consequently the system will return with less discovered topics. There are some problems observed from our experiments.

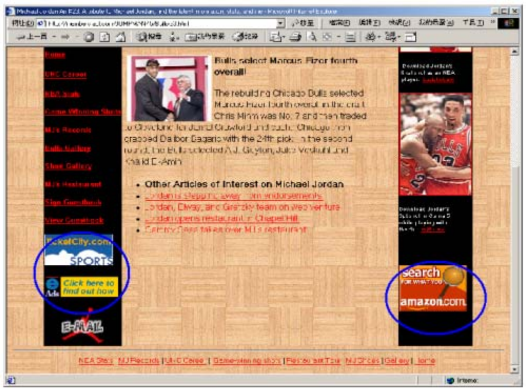
- Commercialization of the Internet: In our experiments we found that the advertisement links, as shown in figure 6, had a negative effect to the topics discovery and ranking accuracy. These commercial pages were added into the base set because they were linked by the top 200 results returned from search engine. A solution to the problem is using an URL stop-list to remove all of those commercial pages.

- TKC (Tightly-Knit Community) effect: The other issue is that some companies build many Web pages that link to each other. Those pages may have different URI and form a strongly inter-connected cluster. Once one of these pages is returned in the search results (e.g. Web page A1), it will trawl in other linked pages (A2, A3, and A4) into the base set as shown in figure 7. As a result, those Web pages will be discovered as an important topic, but almost irrelevant to the query. This is another kind of Internet commercialization, and is known as the **TKC effect** [11]. Since those Web pages have different URIs, it is not likely to detect this phenomenon by using URI checking problem. One solution to this problem is to calculate the domain name distribution of Web pages in the topics. If a topic is composed of Web pages whose domain names are belonging to only few different hosts, very probably it is a tightly-knit topic.

- Mirrored pages: Some Web pages are produced from copying other pages, not just the contents, but also the commercial links. This causes a problem that the authority scores of the commercial pages will be boost. This is a common behavior on Internet that cannot be ignored. We solve this problem by comparing the hyperlinks of each pair of web pages. If over 80% if the links in two Web pages are the same, then they are regarded as mirrored pages. We keep only one of them and remove all the others.

- Annotation of topics: While the algorithm can discover topics from Internet documents, it is difficult to annotate these topics automatically when presented them to user. We have observed that titles of the top hub pages can be used as topic annotation as shown in the experiments.

- Minimal size of a topic: If we set the threshold of minimal topic size to a smaller value, we can discover more topics (as shown in figure 8 for query term 'jordan'), but there will be two shortcomings. One is that some of the discovered topics are trivial topics. The other is that some topics are unstructured. Both of these topics are not valuable to users. If we choose a larger threshold, it is likely to omit some important topics. From repeated experiments, this threshold is set as 20.



Bulls select Marcus Fizer fourth overall!

The rebuilding Chicago Bulls selected Marcus Fizer fourth overall in the draft. Chris Mihm was No. 7 and then traded to Cleveland for Jamal Crawford and cash. Chicago then grabbed Dalibor Bagaric with the 24th pick. In the second round, the Bulls selected A.J. Guyton, Jake Voskuhl and Khalid El-Amin.

- Other Articles of Interest on Michael Jordan
- Jordan is stepping away from endorsements
- Jordan, Elway, and Gretzky team on web venture
- Jordan opens restaurant in Chapel Hill
- [illegible] takes over MJ's restaurant

Figure 6: Commercialization of Internet: non-relevant advertisement links

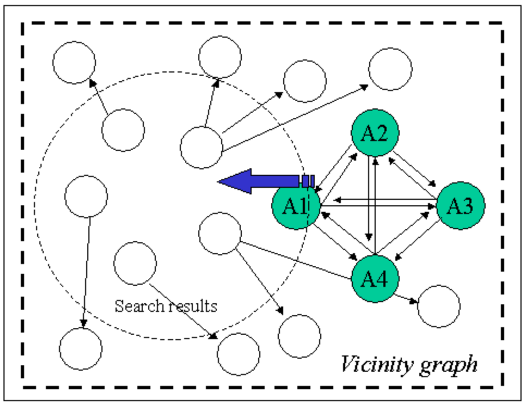


Figure 7: TKC Effect

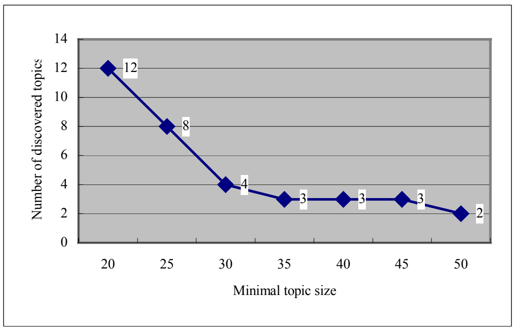


Figure 8: Number of discovered topics vs. minimal topic size

5 CONCLUSIONS

In this work we propose an automatic topic discovery algorithm to discover multiple topics included in the search results for the user given query. Unlike conventional algorithms, our method can automatically enumerate these topics without human intervention. The clustering algorithm presented in this work is based on the underlying link topology and conform to the mutually reinforcing relationship of authority and hub. We use a heuristic to employ the title of top 1 hub page as notation to describe each topic, and this works quite well. This heuristic also meets the definition of a good hub.

The commercialization of Internet causes most Web pages containing hyperlinks to commercial sites and advertisements. Together with the precision of top 200 search results from search engines is not high enough that make the base set containing many irrelevant pages and sometimes affect the results of topic discovery and ranking. Therefore the way to reduce the number of irrelevant Web pages in the base set is the key concern of constructing a base set and requires further study.

It is worthwhile to observe the creation, growth and fall of cyber communities via automatic topic discovery technique that cyber community may be considered as epitome or a special part of the society. The precise interpretation of the automatic discovery needs further study and great help from domain experts.

6 REFERENCES

1. Brian Amento, Loren Terveen, and Will Hill. Does “Authority” Mean Quality? Predicting Expert Quality Ratings of Web Documents. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 2000.

2. Krishna Bharat and Monika R. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In Proceeding of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 1998.

3. Huan Chang, David Cohn, and Andrew McCallum. Creating Customized Authority Lists. In Proceedings of the 7th International Conference on Machine Learning, June 2000.

4. Brian D. Davison, Apostolos Gerasoulis, Konstantinos Kleisouris, Yingfang Lu, Hyun-ju Seo, Wei Wang, and Baohua Wu. DiscoWeb: Applying Link Analysis to Web Search. In Poster proceedings of the 8th International World Wide Web Conference, pages 149-149, May 1999.

5. Brian D. Davison. Topical Locality in the Web. In Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval, July 2000.

6. J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. In Proceedings of the 8th International World Wide Web Conference, pages 389--401, May 1999.

7. David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Inferring Web Communities from Link Topology. In Proceedings of the 9th ACM Conference on

Hypertext and Hypermedia, page 225-234, June 1998.

8. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 668-677, January 1998.

9. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. In Proceedings of the 8th International World Wide Web Conference, page 403-415, May 1999a.

10. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins. Extracting large-scale knowledge bases from the web. In Proceedings of IEEE International Conference on Very Large Databases (VLDB), 1999b.

11. Ronny Lempel and Shlomo Moran. The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC effect. In Proceedings of 9th International World Wide Web conference, May 2000.

12. Shian-Hua Lin, Meng Chang Chen, Jan-Ming Ho and Yueh-Ming Huang, ACIRD: Intelligent Internet Documents Organization and Retrieval, Technical Report of Institute of Information Science, Academia Sinica, 2000 Also to appear on IEEE Transactions on Knowledge and Data Engineering.

13. Gerard Salton. Automatic Text Processing. Addison Wesley, 1989.

14. GSL -- The GNU Scientific Library http://sources.redhat.com/gsl/

15. AltaVista. http://www.altavista.com/

16. Yahoo!. http://www.yahoo.com/