Automatic Topic Identification Using Webpage Clustering

Xiaofeng He[a,b], Chris H.Q. Ding[b], Hongyuan Zha[a], Horst D. Simon[b]

[a] Department of Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802

[b] NERSC Division, Lawrence Berkeley National Laboratory
University of California, Berkeley, CA 94720, USA
{xhe,zha}cse.psu.edu, {chqding,hdsimon}lbl.gov

Abstract

Grouping webpages into distinct topics is one way to organize the large amount of retrieved information on the web. In this paper, we report that based on similarity metric which incorporates textual information, hyperlink structure and co-citation relations, an unsupervised clustering method can automatically and effectively identify relevant topics, as shown in experiments on several retrieved sets of webpages. The clustering method is a state-of-art spectral graph partitioning method based on normalized cut criterion first developed for image segmentation.

1 Introduction

With the exponential growth of information on the World Wide Web, there is great demand for developing efficient and effective methods for organizing and retrieving the information available. Document clustering plays an important role in information retrieval and taxonomy management for the Web and remains an interesting and challenging problem.

A particular issue is how to effectively organize large amount of returned webpages for a user query on a search engine, according to their relevance. Since typical queries have only 1-3 words, ranking those webpages according to textual similarity between query words and webpage text content does not perform well.

Ranking webpages using the information contained in the hyperlinks between webpages currently is an active research area. Two popular ranking methods are PageRank of Brin and Page[2], and the HITS algorithm of Kleinberg [11, 9, 1].

In this paper we consider grouping returned webpages of a query into different topics which are automatically generated by clustering methods exploring textual information, hyperlink structure and co-citation relations. For each topic or cluster, we rank the webpages according to HITS algorithm and list those authoritative webpages only. From the experimental results on several cases, this is an effective way to represent the large number of webpages for a user query.

Document clustering has been studied by many people (see [17] and references there). Clustering retrieval results have been examined by Hearst and Paderson [10] based on textual information only, with emphasis on summarization. More recently this approach is taken in Grouper web interface[18]. Exploring web link structures in the information retrieval context to identify topical themes is examined by Larson[13], Pirolli, et al. [14]. Most recently, Text information is used together with HITS algorithm in *topic distillation* in Chakrabarti et al. [4] and Bharat and Henzinger [1].

The new thrust in our approach is (1) to comprehensively incorporate information from text, link structure and co-citation, (2) combine them in a new way, i.e., as weight matrix for graph partitioning. In our webpage clustering, however, the link structure is the dominant factor, and the textual similarity is used to modulate the strength of each hyperlink. The clustering problem is, in essence, partitioning a graph into connected subgraphs. The nodes of the graph are webpages and the edges of the graph are the hyperlinks between webpages. To further enhance the link structure, co-citation is also incorporated.

In this work, we use a new clustering method based on spectral graph partition. This method has been successfully used in image segmentation by Shi and Malik[16] who use *normalized cut* as criterion to measure the goodness of the partitioning. This criterion lands itself immediately to the Rayleigh quotient which can be easily solved. The method recursively partition a subgraph into two sub-subgraphs until a stopping condition is met.

It is important to note that our goal is "topic identification and extraction from a webpage collection". Exactly how many clusters or topics is not important, as long as all main or significant topics are identified and extracted. The stopping condition is a measure of intra-cluster similarity vs inter-cluster dis-similarity, which essentially measures the significance of a cluster. Thus our recursive clustering method fits the goal well.

We gathered three retrieval datasets, each corresponding to a one-word query: *amazon, star, apple* submitted to a search engine. Using a similarity metric that incorporates textual information, link structure and co-citation, we clustered each dataset using the spectral partition with normalized cut. The results show that our method effectively distinguishes different topics mixed together in a dataset. This automatic topic clustering is rather difficult to achieve if we use text information only.

It is important to note that the goal of clustering webpages is to effectively *organize* the retrieval information. This is different from *topic distillation*[1] which attempts to find the most *relevant* webpages for a query. Given the large number of retrieved webpages and mixing of different topics therein, it is rather difficult to clearly define which webpage is most *relevant*. By grouping webpages into different topics and listing authoritative webpages in each topic, we let users determine which webpages are relevant. Very often, a user finds a webpage or a topic he/she had not anticipated, or may not be relevant in a strict sense; but the topic is related and interesting that the user will pursue further. This kind of *better-than-expected* webpages or topics could be missed in topic distillation process; however, they are enhanced in our approach as separate clusters, as the experimental results show.

2 Similarity Metric

Clustering objects into subgroups is usually based on a similarity metric between objects, with the goal that objects within a subgroup are very similar, and objects between different subgroups are not similar. In clustering of a graph, similarity between nodes is represented as weight of the edge.

In webpage clustering problem, we incorporate link structure, text information, and co-citation information into the similarity metric, which form the weight matrix W.

2.1 Hyperlink Structure

The link information is obtained directly from the link graph. Given a link graph $G = (V, E)$, which is directed, we define the adjacency matrix A of the graph to be:

$$A_{ij} = \begin{cases} 1 & \text{if} \quad e_{ij} \in E \quad \text{or} \quad e_{ji} \in E \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Link structure alone provides us with rich information on the topic. By exploring the link structure, we are able to extract useful information from the web [9, 5, 1, 8, 12]. One of the most popular algorithm to retrieve information from the link structure is Kleinberg's HITS algorithm[11], which will be briefly introduced later.

2.2 Textual information

The textual information is included to better cluster the webpages. Moreover, Unlike printed literature, web text references each other more randomly. This is another reason that the text information is incorporated. One approach that textual information has been incorporated is to measure the similarity between user query and the anchor text (text between ⟨A HREF=...⟩ and ⟨/A⟩) as in [5, 1]. We experimented with this approach and found it did not work as effective in our data sets.

Here we use a new approach that (a) utilizes the entire text of a webpage, not just the anchored words; (b) measures textual similarity S_{ij} between two webpages i, j, instead of between user query and the webpage. (c) use S_{ij} as the strength of the hyperlink between webpages i, j. The key observation here is that if two webpages has very little text similarity, it is unlikely that they belong to the same topic, even though they are connected by a hyperlink. Therefore S_{ij} properly gauges the extent or the importance of an individual hyperlink.

We call each webpage a document. The content of a document is obtained using a web crawler written in *Perl*. To accommodate the vast differences in webpage lengths, we only use the first 500 words of each document; the rest of the document is discarded if it has more than 500 words.

After text of all webpages are preprocessed, we represent each webpage by a vector in the vector space model of IR. For each element of the vector we use the standard tf.idf weighting: $tf(i,j) * idf(i)$. $tf(i,j)$ is the term frequency of word i in document j, representing the number of occurrence of word i in document j. $idf(i)$is the Inverse Document Frequency for word i, computed as

$$idf(i) = log(\frac{\text{no. of total docs}}{\text{no. of docs containing word } i})$$

We compute the *similarity* (or *relevance*) between two webpages using the standard cosine similarity measure. If x and y are vectors of two documents j_1 and j_2, the similarity between j_1 and j_2 is:

$$S(j_1, j_2) = \frac{\sum_i x(i) * y(i)}{\|x\|_2 * \|y\|_2}$$

Note that this simple textual similarity is the starting point of our approach. More sophisticated forms of textual similarity can be used. For example, one may weight more on the words in webpage titles.

2.3 Co-citation

Co-citation is another metric to measure the relevance of two webpages. If there are many pages pointing to both of them, then these two pages are likely to address the similar issue. The co-citation $C(i,j)$ of pages i and j is the number of webpages pointing to both i and j. The co-citation matrix C is easily obtained from the link graph.

The overall similarity between two webpages is the combination of above three factors. In matrix form, it can be written as

$$W = \alpha \frac{A \otimes S}{\|A \otimes S\|_2} + (1 - \alpha) \frac{C}{\|C\|_2} \quad (2)$$

where $(A \otimes S)_{ij} = A_{ij} S_{ij}$. W can also be viewed as the weight matrix of an *undirected* graph. Here α is a parameter and has value between 0 and 1. In all our experiments, we set α to be 0.5. Once W is computed, the problem of clustering webpages becomes the problem of graph partitioning using the weights given in W as the similarity between nodes. Note that diagonal elements of W are zero.

3 Clustering Algorithm

We solve the web graph partitioning problem by a spectral graph partitioning method which uses Fiedler vector of the affinity matrix to separate two clusters, measured by normalized cut criterion. We also used K-means clustering method and found the results of spectral partitioning is usually much better.

3.1 Spectral graph partition using Normalized cut

Normalized cut is an unbiased measure of partitioning the graph G into two subgraphs A, B. Shi and Malik [16] first use normalized cut in image segmentation, using the affinity matrix of an image as the weight matrix of an undirected graph. Using the similarity metric between webpages as the weight matrix on the web graph, the webpage clustering problem can use the same graph partitioning method as the image segmentation problem of Shi and Malik.

Define the objective function for the normalized cut as:

$$J(A,B) = \frac{cut(A,B)}{\sum_{i \in A} d_i} + \frac{cut(A,B)}{\sum_{j \in B} d_j} \quad (3)$$

where the weighted cut size is defined as

$$cut(A,B) = \sum_{i \in A, j \in B} W(i,j), \quad (4)$$

and d_i is the degree of node i: $d_i = \sum_k W(i,k)$. The clustering problem is to find the partition of G that minimizes $J(A,B)$.

Let y be an indicator vector that determines the partition: each element of y is an indicator variable for a node i,

$$y_i = \begin{cases} 1 & \text{if } i \in A \\ -b & \text{if } i \in B \end{cases} \quad (5)$$

Let $D = Diag(d_1, d_2, \cdots, d_n)$. be a diagonal matrix. The objective function becomes

$$J(y) = \frac{y^T (D - W) y}{y^T D y} \quad (6)$$

subject to the constraint

$$y^T D e = 0 \quad (7)$$

where $e = (1, 1, \cdots, 1)^T$. Although solving this optimization problem for discrete variables y_i is rather difficult, one can *relax* y_i to continuous variables within $[-1, 1]$, and find the optimal solution.

Expression (6) is the Rayleigh quotient. It can be minimized by solving the generalized eigensystem:

$$(D - W)y = \lambda D y \tag{8}$$

One easily see that $y = e$ is the eigenvector corresponding to smallest eigenvalue $\lambda_1 = 0$. Thus the eigenvector f corresponding to the second smallest eigenvalue is the solution to the minimization problem, since it automatically satisfies the constraint (7). f is usually called *Fiedler Vector*, in recognition of M. Fiedler's work[6, 7].

Here we show the result on a simple example. Figure 1(a) shows the adjacency matrix randomly generated to denote an undirected graph with two main subgraphs, where the links between the subgraphs are very sparse (low density) and links within each subgraph are dense (highly connected). Figure 1(b) shows the Fiedler vector for partitioning the graph. In this example, each element of f is close to one of two discrete values (0.02 and -0.09), corresponding to $c(1, -b)$ where c is normalization constant. The splitting point $f_i = 0$ clearly separate the graph into two distinct parts. An important property of Fiedler vector is that if the original graph is simply connected, each subgraph is guaranteed to be simply connected. Without this property, the splitting of nodes could lead to disconnected subgraphs.

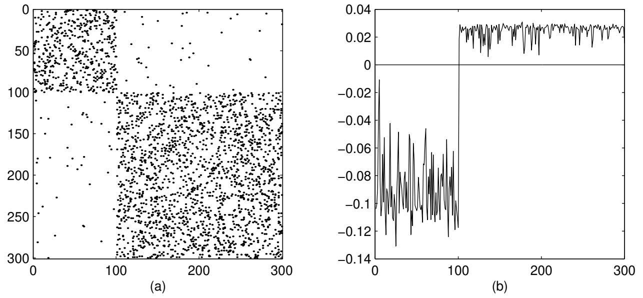

Figure 1: (a) The adjacency matrix of a random graph. The first and second diagonal blocks correspond to two subgraphs. They have high connectivity. Off-diagonal blocks represent sparse edges between the subgraphs. (b) Nodal values of the Fiedler vector.

For general problems, the node values f_i of the Fiedler vector are continuous and there is no clear gap between two clear discrete values. We sort the node values and then pickup a splitting point so that all nodes above it belong to one subgraph and all nodes below belong to the other subgraph. The cutoff point is chosen so that the objective function $J(y)$ is minimized. This can be done by checking N equally spaced splitting points, and then find the point with the smallest $J(f)$.

It is instructive to examine the objective function $J(A, B)$. If we minimize $cut(A, B)$ without restrictions on the sizes of two subgraphs, the optimal solution is often a very skew one, with one very small subgraph cut away from the original graph. If we minimize $cut(A, B)$ requiring that the two subgraphs have the same number of nodes: $|A| = |B|$, this reduces to graph partitioning using spectral bisection [15]. In this case, the indicator variables have discrete values $y_i = \{-1, 1\}$ and the objective function becomes $J(y) = y^T(D - W)y/y^Ty$. The solution is the eigenvector for the second smallest eigenvalue of $(D - W)y = \lambda y$. The general situation is more interesting when the two subgraghs have different number of nodes (like our clustering problems). The objective function of Eq.(3) allows the number of nodes to be determined as part of the solution, therefore is most appropriate for clustering.

3.2 Hierarchical Divisive Clustering

Once a webgraph is partitioned into two subgraphs, we can partition each of the subgraph using the same method. This recursive partitioning process is repeated until a stopping criteria is met. The value of the objective function in the normalized cut proves a natural stopping criteria. For a graph to be partitioned, we compute the Fiedler vector f, find the optimal cut, and obtained $min\ J(f)$ value. If $min\ J(f)$ value is above certain threshold value J_{stop}, it means the cutsize between two partitions is relatively large, and the resulting two subgraphs would have relatively high connectivity between them, hence it is better not to partition the graph further and the recursive process stops. The threshold value is set to $J_{stop} = 0.06$ in all our experiments.

Note that in this method, exactly how many clusters or topics will results is not known *a priori*. This feature fits our goal of "topic identification and extraction" very well. We are only interested in extract main or significant topics. In Web environment, documents (webpages) are created for many different reasons, and quality control is near zero. It is very likely that there are many webpages in a web collection that do not belong to any clear topics. Setting a K before hand and forcing all webpages into these K clusters will produce clusters of low quality.

The complete recursive clustering algorithm is:

1. $d = We$, where $e = (1, 1, \cdots, 1)$. $D = \text{Diag}(d)$.

2. Solve generalized eigensystem $(D - W)y = \lambda Dy$ for eigenvector f with the second smallest eigenvalue.

3. Check N equally spaced splitting points of f, find the cutting point with the smallest $J(A, B)$.

4. If the value of $J(A, B)$ is below the threshold, accept the partition and recursively partition the subgraphs. Otherwise, done.

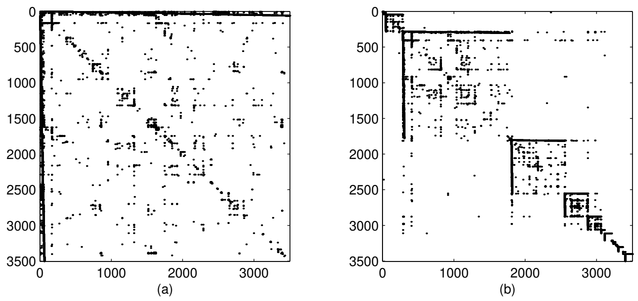

Figure 2: Clustering webpages related to user query *star*. (a) Weight matrix of the graph. (b) Clustering result. This corresponds to re-order the nodes such that nodes belonging to a cluster are grouped together.

Here we illustrate the clustering result on a dataset retrieved from a query that contains one word *star* (see Section 5.2 below). In Figure 2(a) we plot the weight matrix. All nonzero elements are shown as a dot. Figure 2(b) shows the re-ordered graph after clustering. Each diagonal block corresponds to a cluster. An off-diagonal block corresponds to a cut, i.e., all links between two clusters. One see that the clustering method effectively find the set of cut links that forms a very sparse off-diagonal blocks.

4 Experiments

4.1 Data sets

A dataset is obtained in the following way [3, 11]. We first submit a query to the text-based search engine *hotbot*. *hotbot* returns as query result a list of URLs with highest ranked webpages. We only take top 120 of returned URLs, which form the *root set*. Limiting the number of returned URLs to be 120, we can keep the overall data set with a reasonable size. Then we expand the root set by including all webpages that point to a webpage in the root set and those pointed to by a webpage in the root set. This expanded *neighborhood set* of webpages is the dataset associated with a particular query. The hyperlinks between these webpages form a directed link graph, from which the adjacency matrix is computed. We run web crawler to get the text information of these webpages and compute the webpage-webpage textual similarity as described before. The co-citation matrix is calculated from the link graph.

We apply the clustering algorithm on the data set. The threshold for stopping the recursive partitioning is set to .06. This leads to K clusters or topics. Since each cluster often has a large number of webpages, we choose only the most important webpages among a cluster. The most important or authoritative webpages are determined using Kleinberg's HITS algorithm.

4.2 HITS Algorithm

HITS algorithm make use of the directionality, defining the *authorities* as the most relevant webpages for the topic, and the *Hubs* as the webpages which link to many related authorities. Hubs implicitly represent an "endorsement" of the authorities they point to. The authority and hub information can be extracted based entirely on the link structure. Since good *authority* is pointed to by many good *hubs* and good *hub* points to many good authorities, such mutually reinforcing relationship can be represented as:

$$x_p = \sum_{q: e_{qp} \in E} y_q \qquad (9)$$

$$y_p = \sum_{q: e_{pq} \in E} x_q \qquad (10)$$

where x_p is the authority weight of webpage x and y_p is the hub weight. E is the set of links(edges). Final solutions are reached after iteratively updating the authority and hub weights of every webpage using Eqs.(9,10). Webpages with highest weights are the authorities and hubs of the topic.

5 Clustering results

We tested our method on three datasets corresponding to three one-word queries: *amazon, star*, and *apple*. The reason for using short query is to have multiple topics among the retrieved webpages, that can effectively test our clustering algorithm.

We first calculate the similarity matrix Eq.(2) for the web graph on the datasets, then run the spectral partitioning algorithm on the datasets to get clusters. Finally we apply HITS algorithm on each cluster obtained. The top authorities of each significant clusters are listed. By significant we mean the size of the cluster is not too small. The clusters with small size are not listed. Some small clusters exist because the nodes in them form highly connected components of the link graph. It shows that our algorithm is capable of finding these small but tightly connected clusters.

5.1 Dataset *amazon*

This dataset is retrieved for query *amazon*, with total 2294 webpages (URLs). *amazon* has at least three meanings. One is related to amazon.com, one of the largest on-line shopping websites. Another is the famous rain forest in South America. And the third is the name of ancient female warriors from Alecto, a female ruled monarchy.

Applying the clustering algorithm, we obtain following significant clusters. (In the following, the order of the clusters has no special meanings. It's merely the sequence by which we process the clusters.)

Cluster 1:
`www.amazon.com/`
`www.amazon.co.uk/`
`www.amazon.de`

These three are the homepages of amazon.com, the largest online shopping company. The first website is located in USA, the second in UK and third, Germany. It is quite re-assuring that our method clusters them together. We list only three authorities here because the rest listed in the top 10 all have very low authority weights compared with the first three. The reason is that those URLs have few in-coming links, and are not regarded as authorities, so we don't list them here.

Cluster 2:
`www.amazoncity.com/`
`www.amazoncityradio.com/`
`www.amazoncity.com/spiderwoman/`
`www.wired.com/news/news/culture/story/6751.html`

This cluster is about female issue.

Cluster 3:
`www.amazon.org/`
`www.amazonfembks.com/`
`igc.apc.org/women/bookstores/`
`www.teleport.com/ rocky/queer.shtml`
`www.advocate.com/html/gaylinks/resources.html`

The topic of this cluster is on female related issues: bi-sexuality and female books. That's a big surprise that we extract a cluster on bi-sexual topic. It's totally beyond our expectation.

Cluster 4:
`sothebys.amazon.com/exec/varzea/tg/special-sales/-/22822/-/8253/ref=gw_m_col_2/`
`sothebys.amazon.com/exec/varzea/subst/home/sothebys.html/ref=gw_m_col_au_2/`
`sothebys.amazon.com/exec/varzea/tg/special-sales/-/22822/-/8253/ref=gw_m_col_au_1/`
`s1.amazon.com/exec/varzea/subst/home/home.html/ref=gw_auc_1/`
`sothebys.amazon.com/exec/varzea/subst/home/sothebys.html/ref=gw_m_ln_br_so_2/`

All five authorities listed here are webpages of a large on-line auction company formed by Sothebys and amazon.com. It's not clustered into Cluster 1 because there are only a few links between them.

Cluster 5:
`www.swalliance.com/`
`timeline.echostation.com`
`www.echostation.com:8080/~1`
`downtime.echostation.com`
`rpg.echostation.com/`

The topic of this cluster is about the movie *Star War*, surprisingly. But there are total 68 pages on this topic, most created by Star War fans. Some are on-line shopping companies selling goods related to the movie *star war*.

There are two clusters, but for each of them, the webpages are from the same site. These two clusters are:

Cluster 6:
`misc.langenberg.com/`
`cooking.langenberg.com/`
`shipping.langenberg.com/`
`money.langenberg.com/`
`weather.langenberg.com/`
and
Cluster 7:

www.latingrocer2.com
www.latingrocer.com
www.latingrocer.com/Pages/customer.html
www.latingrocer.com/Pages/privacy.html
www.latingrocer.com/Pages/contact.html

Cluster 8:
http://www.internext.com.br/ariau

This cluster has only one authority. Other web documents in this cluster all point to it. It's a website of a hotel in Amazon River valley.

Clusters 5, 6, 7 are not really relevant to the query *amazon*. Their presence in the dataset is mainly due to the root set expansion, since they have links to the webpages in the root set. This signals a problem with root set expansion: some high in-degree but not relevant nodes are included in the neighborhood set during the expansion. Once included, these high in-degree nodes will rank high and cause distraction.

From the clusters we obtained, we found that no cluster has a focused topic on Amazon rain forest, unlike what we had expected. Checking the entire dataset, the 2294 URLs, we only got a couple of webpages that mention about the rain forest. They don't form a cluster with significant size. If we return to *hotbot* and enter the query *amazon*, the web sites presented are dominantly related to amazon.com. The pages about rain forest are not in the top 120 list returned by *hotbot*, and thus is not included in the root set. This probably indicated that on the web, the number of webpages on amazon rain forest is small.

As for the third meaning of amazon, although female warrior does not appear directly as a distinct topic in any cluster, some clusters focus on female issues, or even bi-sexual issue. By examining the content of these pages, we think that these issues are extension of the original meaning of amazon as female warriors. Especially, the cluster on the topic of bi-sexuality is beyond our expectation — our method can identify (from the web graph) topics even not aware of by us.

There are clusters with all authorities from the same websites, such as the clusters 6 and 7. They are explored as separate clusters because the webpages on the same site point to each other, raising the importance themselves. To avoid such situation to happen, before applying any clustering algorithm, we can coarsen the link graph first, so that the webpages from the same site collapse to one node in the graph.

5.2 Dataset *star*

This dataset is retrieved for query *star*. There are 3504 webpages (URLs) in this dataset. For *star*, we can think about a planet, a movie star, a famous athlete, or the movie *Star War*.

We run our clustering algorithm on the dataset to get clusters automatically. The authorities of each cluster are listed below:

Cluster 1:
www.starwars.com/
www.lucasarts.com/
www.sirstevesguide.com/
www.jediknight.net/
www.surfthe.net/swma/

This cluster is focused on *star war*.

Cluster 2:
www.kcstar.com/
www.dailystarnews.com/
www.kansascity.com/
www.starbulletin.com/
www.trib.com/

This cluster includes the webpages of some news media with the word *star* as part of their names.

Cluster 3:
www.weatherpoint.com/starnews
www.starnews.com/digest/sports.html
www.starnews.com/digest/citystate.html
www.indy.com
speednet.starnews.com/

The top authorities are all webpages of starnews.com in this cluster.

Cluster 4:
www.state.mn.us/mainmenu.html
www.mda.state.mn.us/
www.doli.state.mn.us/
www.legalethics.com/pa/states/state/mn.htm
www.exploreminnesota.com

This cluster's topic is the state of Minnesota. The reason that Minnesota is a topic of the cluster under query *star* is because the official State of Minnesota web site is called *North Star*, which is named second among all government web sites.

Cluster 5:
www.star-telegram.com/
www.dfw.com/

www.virtualtexan.com/
marketplace.dfw.com
www.star-telegram.com/advertise/vshops/

This is a cluster of star-telegram.com located in Texas. It is interesting to note that clusters 2, 3, 5 are all webpages of news media. They are partitioned to be different clusters since there is no link among the three clusters: they form separate communities.

Cluster 6:
www.starpages.net/

There is only one authority in this cluster. This website introduces stars in various fields, such as movie star, sport stars, model stars, etc. It is interesting to learn the existence of such a website.

Cluster 7:
www.aavso.org/
www.astro.wisc.edu/ dolan/constellations/
ourworld.compuserve.com/homepages/rawhide _home_page
adc.gsfc.nasa.gov/adc/adc_amateurs.html
heasarc.gsfc.nasa.gov/docs/www_info/webst ars.html

This cluster talks about space and astronomy, as we had expected.

Overall, it appears that all topics related to *star* are automatically identified and extracted into separate clusters.

5.3 Dataset *apple*

In this data set, there are 2757 URLs returned by the search engine. The authorities of each cluster found are listed below:

Cluster 1:
www.apple.com/
www.apple.com/support/
www.apple.com/education/
www.apple.com/quicktime/
www.apple.com/hotnews/

Here all top authorities are from the same web site: *www.apple.com*. This cluster is dominant in this query. Most URLs belong to it. After running the HITS algorithm with the site information considered, that is, the URLs from the same web site are collapsed to one node, we obtain different top authorities:

www.apple.com/powermac/server/
www.claris.com/
www.apple.ru/hardware/displays
www.cs.brandeis.edu/~xray/oldmac.html
www.next.com/

The second URL in this cluster is the website of a computer software company. It produces software used for the Macintosh. The other four URLs are all about the apple computer. This list of authorities provides more useful information than the previous one does.

Cluster 2:
www.yabloko.ru/
www.cityline.ru/politika/
www.russ.ru/
www.forum.msk.ru/
www.novayagazeta.ru

All web documents in this cluster are written in Russian.

Cluster 3:
www.michiganapples.com/

This cluster has only one authority. It's related to apple, the fruit.

The following cluster is formed in this query simply because its name happens to contain the query term *apple*:

Cluster 4:
www.ci.apple-valley.mn.us/

which is the website of the city of Apple Valley, MN.

Cluster 5:
www.valleyweb.com/

This is the website of Annapolis Valley in Canada where there is the Apple Blossom Festival in the spring celebrating the traditions and agricultural heritage.

There are not many clusters formed by the algorithm, nor do we find interesting topics which are beyond our original expectation.

6 Further discussion

By intuition, the links in the web graph can not be regarded as equally important, especially many web links are created randomly. Text similarity between two webpages provides us with a useful metric to address this issue. Without the text similarity as the measure of the link strength, that is, if the weight matrix is formed as:

$$W = \alpha \frac{A}{\|A\|_2} + (1-\alpha)\frac{C}{\|C\|_2},$$

this will unduly raise the strength of some links which connect two webpages with little in common. When applying Algorithm 1 to the dataset *amazon*, we group Clusters 1, 3 and 4 into one single cluster, using the same threshold as the stoping criterion, namely 0.06 in our previous experiments. Cluster 3 addresses the female issues and Clusters 1 and 4 are related to *amazon.com*. Apparently this clustering result is not a good one. This result justifies our choice of incorporating text information into weight metric.

7 Concluding Remarks

In this paper, we present a method to automatically group webpages into distinct topics and list the most authoritative/informative webpages in each topic. This method of organizing retrieved information on the web is an effective way to help user to explore the vast amount of information on the web. The experimental results on several retrieved datasets indicate the effectiveness of the method.

Two key parts in our method are (1) the similarity metric incorporates comprehensive information of text, hyperlink structure and co-citation; and (2) the unsupervised clustering method based on spectral graph partitioning using normalized cut.

References

[1] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. *21st ACM SIGIR on Research and Development in Information Retrieval*, 1998.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Proc. of 7th WWW Conferece*, 1998.

[3] S.J. Carriere and R. Kazman. Webquery: Searching and visualizing the web through connectivity. *Computer Networks and ISDN Systems*, 29(1):1257–1267, 1997.

[4] S. Chakrabarti, B. E. Dom, and J. M. Kleinberg. Mining the link structure of the world wide web. Feb 1999.

[5] S. Chakrabarti, B. E. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1-7):65–74, 1998.

[6] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Math J.*, 23:298–305, 1973.

[7] M. Fiedler. A property of eigenvectors of non-negative symmetric matrices and its application to graph theory. *Czechoslovak Math Journal*, 25:619–633, 1975.

[8] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. *Proc. of ACM SIGKDD, pp.150-160*, 2000.

[9] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia (HYPER-98)*, pages 225–234, New York, June 20–24 1998. ACM Press.

[10] M. A. Hearst and J. O. Paderson. Re-examining the cluster hypothesis: Scatter/gather on retrieval results. *Proc. SIGIR'96*, pages 246–255, 1996.

[11] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Proc. 9th ACM-SIAM Symposium on Discrete Algorithm*, pages 668–677, 25–27 January 1998.

[12] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the web. *Proc. of the 25th VLDB Conference*, 1999.

[13] R. R. Larson. Bibliometrics of the world wide web: an exploratory analysis of the intellectual structures of cyberspace. *Proc. SIGIR'96*, pages 71–78, 1996.

[14] P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures from the web. *Proc. SIGCHI'96*, pages 118–125, 1996.

[15] A. Pothen, H. D. Simon, and K. P. Liou. Partitioning sparse matrices with egenvectors of graph. *SIAM Journal of Matrix Anal. Appl.*, 11:430–452, July 1990.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, June 1977.

[17] P. Willett. Recent trends in hierarchical document clustering. *Information Processing and Management*, 24:577–597, 1988.

[18] Oren Zamir and Oren Etzioni. Grouper: A dynamic clustering interface to web search results. *WWW8*, 1999.