Empirical Analysis of Grouping Web Pages Using Vector Space Model for Link Structures

Yuichi Sasaki
Research Group of Complex Systems Engineering
Graduate School of Information Science and Technology,
Hokkaido University
Kita14 Nishi9,Kita,Sapporo,JAPAN
Email: yusasaki@complex.eng.hokudai.ac.jp

Masahito Kurihara
Research Group of Complex Systems Engineering
Graduate School of Information Science and Technology,
Hokkaido University
Kita14 Nishi9,Kita,Sapporo,JAPAN
Email: kurihara@ist.hokudai.ac.jp

***Abstract*—Several kinds of vector space models for analyzing document similarity for grouping web pages have been developed. However, they are not used for analyzing link structures, partly because they are complex and links do not necessarily satisfy the similarity relation. If we can devise vector space models for link structures, we can combine them with those models for document similarity in order to develop the unified basis for grouping web pages.**

In this paper, we present a vector space model for link structures, based on the notion of link vectors, the specifically designed characteristic vectors for link structures. We also discuss the extension of this model to the model called content-link vector space model, which can treat document information and link information of web pages in a unified way. The preliminary experiments show that the models show good performance even when document information is ignored.

I. Introduction

A. Background

Recently, the number of web pages on the Internet has been rapidly increasing and the pages have become more diverse. According to Gulli and Signorini [1], there exist more than 11.5 billions of web pages. As the result, the output of Internet searches often contains a lot of unexpected, useless web pages. We must make an extra effort to find the web pages we really want. This problem has led to many studies on creating new methods to reduce the search effort needed and to help us get more useful results.

One major approch involves techniques to discover appropriate *groups* of interrelated web pages. This approach exploits not only the given search strings, but also the link structures and the contents of the web pages in order to provide users with useful information on the interrelationships of the retrieved pages. However, studies for analyzing the contents and those for analyzing the link structures have been conducted almost independently of each other. In this paper, we present a technique that can effectively serves as a unified basis for combining those two kinds of analyses.

B. Vector space model using link structures

Several kinds of vector space models[2] that compute document similarity for grouping web pages have been developed and reported. They are based on *document vectors* (or *content vectors*) computed from statistic information on the contents, such as the *term frequency-inverse document frequency* (tf-idf) scores for the words contained in the documents. In a hypertext environment like the World Wide Web, however, web pages contain hyperlinks in addition to their normal contents. Since those links are useful for identifying the semantic relations between the web pages to be grouped, we believe that vector space models should be extended so that they can handle, in a unified way, link structures in addition to content information.

As the first step to this extension, we present in this paper a design of vector representation for link structures, called the *link vectors*. To quantify link structures as numerical vector elements is not straightforward because the links do not necessarily involve similarity relations. In spite of this difficulty, Kumar et al. [3] considered directed graph structures, defined by a set of links, and showed that complete bipartite graphs can be regarded as communities, where a community means a group of individuals who share a common interest, or whose web pages have similar contents. In reality, however, communities are not always identified with complete bipartite graphs. By using the link vectors, on the other hand, we can handle complete bipartite graphs flexibly by relaxing their rigid graphical definition, while retaining the essential ideas behind the original definition. In this paper, we report the results of our implementation and experiments, and empirically show that the link vectors show almost the same performance as the graphical approach of Kumar et al., even when the link vectors are not combined with content vectors.

To create more useful groups of web pages, we also present a new representation for characterizing web pages. This representation, called the content-link vectors, is a weighted concatenation of content vectors and link vectors. Given the content-link vectors for two web pages, their similarity can be defined in the same, unified way as in the standard vector space models, i.e., as the normalized dot product of the vectors. Although comprehensive experiments are still to be done, we believe that this representation can improve the performance of web page grouping systems significantly.

The paper is structured as follows. In Section 2, we briefly describe the vector space models for the documents and review some related works on web page grouping. We then define the

978-1-4244-3782-5/08/$25.00 ©2008 IEEE

link vectors and content-link vectors in Section 3. In Section 4, we report and analyze the results of our experiments. In Section 5, we present our conclusion and discuss some future works.

II. Related work

A. Vector space model

Vector space models[2] can be used for computing the similarity between documents. The elements of the feature vectors for the models are the word statistic values of the documents, such as word frequencies or tf-idf scores. Let $\overrightarrow{D_i}$ be a *feature vector* for document d_i,

$$\overrightarrow{D_i} = [ws(d_i, w_1), ws(d_i, w_2), \ldots, ws(d_i, w_n)] \quad (1)$$

where n is the number of words and $ws(d_i, w_j)$ is the score of the word w_i in d_i. The similarity $ds(d_i, d_j)$ between two documents d_i and d_j is the normalized dot product of the feature vectors, given by

$$ds(d_i, d_j) = \frac{\overrightarrow{D_i} \cdot \overrightarrow{D_j}}{|\overrightarrow{D_i}||\overrightarrow{D_j}|}. \quad (2)$$

If $ds(d_i, d_j)$ is close to 1, d_i and d_j are considered to be similar.

B. Trawling the web

Kumar et al.[3] considered that *fans*, which are web pages that contain links to the *center* pages, make communities defined by complete bipartite graph structures. They proposed a trawling algorithm that quickly finds (i, j)-sized complete bipartite subgraphs on the Web, where i and j are the number of the fans and centers. They then applied the algorithm to a huge number of crawled web pages. The manual inspection of randomly sampled web pages in the resultant communities showed that pages in the same communities were strongly related.

C. Grouping web pages

a) Shortest-path betweenness: Girvan and Newman have proposed an iterative division method based on progressively finding and removing the community edges with the largest *betweenness*, until the network breaks up into components[4]. The steps in the Girvan-Newman algorithm are as follows.

1) Compute the largest betweenness scores.
2) Remove the edge with the largest betweenness.
3) Repeat steps 1 and 2 until all the edges are removed and the system breaks up into n non-connected nodes.

The motivation of betweenness is that the traffic of flow through the network will have to travel over at least one of the edges between the communities if it wishes to pass from one community to another. The betweenness of an edge is defined as the number of shortest paths between vertex pairs.

b) Newman clustering: Newman[4] proposed *modularity* that measures the quality of a particular division of a network. It also can be used for dividing a network. We define e_{ij} as the fraction of edges in the original network that connect vertices in group i to those in group j. Then the modularity is defined as

$$Q = \sum_i \Big(e_{ii} - (\sum_j e_{ij})^2 \Big). \quad (3)$$

Physically, Q is the fraction of all edges that lie within communities minus the expected value in which the vertices have the same degrees but where the edges are placed at random. The more edges within the same communities and the less edges between different communities, the larger the modularity grows. In practice, some value above 0.3 is a good indicator that a network has a significant communitiy structure.

In dividing a network, it is hard to find the grobal maximum modularity over all possible divisions. Therefore, a greedy optimization is used to find a good result. Such an optimization algorithm[5], [6] starts with communities that consist of only one vertex, then the two communities whose amalgamation produces the largest increase in Q are repeatedly joined together. When community i and j are joined together, the increment modularity is

$$\Delta Q_{ij} = 2(e_{ij} - \sum_i e_{ij} \sum_j e_{ij}). \quad (4)$$

For a network of n vertices, the algorithm stops when $n - 1$ joins result in a single community left. Note that this equation implies that Q has a single peak in the course of the algorithm, since after the largest ΔQ becomes negative, all ΔQ can only decrease. The communities that gives the largest Q are the best result of the network division.

c) Content-link hypertext clustering: Weiss[7] has proposed HyPursuit, which is a clustering search engine that exploits both document content and link structures in web pages. HyPursuit computes the content and link structure similarity between web pages and uses a hybrid similarity function to combine them. The content similarity between two documents is the normalized dot product of the feature vectors. The elements of the feature vectors are the fraction of term frequencies weighted with the document size and term attribute, such as header, title or plain text. The link structure similarity between two documents captures the following three important notions about hyperlink structures implying semantic relations.

1) The shortest path length between the two documents.
2) The sum of the scores weakened by the shortest path length between each document and the common ancestor documents that both documents refer to.
3) The sum of scores are weakened by the shortest path length between each document and the common descendant documents that both documents are refered to.

III. Proposed Vector Space Models

A. Link vector

Vector space models can be used not only for measuring the similarity between two documents, but also for computing it. This is done with link structures that are generally difficult to deal with. In this section, we design link vectors for link structures. The link vectors handle link structures based on the idea that fans and centers make communities [3]. Use of the link vectors relaxes the restrictions to complete bipartite graphs and leads to more flexible web grouping rather than strongly related grouping like communities defined by Kumar.

In a complete bipartite graph consisting of fans and centers, its link structure is created by the fans having links to the centers. This means that the centers contain link structure information provied by the fans. Let us define the values of the link vector elements of the centers as 1, because we can go to the centers by following *one* link from the fans. Let p_i be a center web page that has common N fans. Then link vectors $\overrightarrow{L_i}$ and $\overrightarrow{L_j}$ have N with the common value 1, because we can reach p_i and p_j by following one link from the fans. Then, in the same way as the document contents, the similarity $Sim_l(p_i, p_j)$ between p_i and p_j determined from the link structure is the normalized dot product of $\overrightarrow{L_i}$ and $\overrightarrow{L_j}$,

$$Sim_l(p_i, p_j) = \frac{\overrightarrow{L_i} \cdot \overrightarrow{L_j}}{|\overrightarrow{L_i}||\overrightarrow{L_j}|} = \frac{1 \cdot 1 + 1 \cdot 1 + \cdots + 1 \cdot 1}{\sqrt{N} \cdot \sqrt{N}} = 1. \tag{5}$$

The result indicates that the centers that have common fans are quite similar and can be grouped together. Consequently, this link vector leads to finding a group of centers with links from the fans to the centers.

The link vectors also flexibly measure the difference between web pages that have a complete bipartite graph structure and those that do not, as we demonstrated in Fig.1.

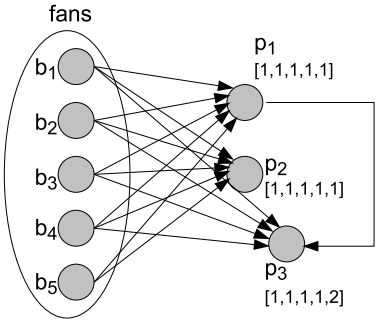

Fig. 1. Examples of link vectors

TABLE I
Link vectors and their similarities

web page p_i	sim p_1	sim p_2	sim p_3	link vector $[b_1, b_2, b_3, b_4, b_5]$
p_1	—	1.000	0.949	[1,1,1,1,1]
p_2	1.000	—	0.949	[1,1,1,1,1]
p_3	0.949	0.949	—	[1,1,1,1,2]

Let L_i be a link vector for page p_i in Fig.1. All elements of link vectors L_1 and L_2 are 1 because of the complete bipartite graph structure. On the other hand, the fifth element of L_3 is 2 because p_3 requires at least two links to reach from b_5. That is why the link vectors are $\overrightarrow{L_1} = \lfloor 1, 1, 1, 1, 1 \rfloor$, $\overrightarrow{L_2} = \lfloor 1, 1, 1, 1, 1 \rfloor$ and $\overrightarrow{L_3} = \lfloor 1, 1, 1, 1, 2 \rfloor$. Then the normalized dot products of the pairs of these link vectors are $Sim_l(p_1, p_2) = 1.0$ and $Sim_l(p_1, p_3) = 0.949$. The result shows that p_1 and p_3 are less similar than p_1 and p_2. With link vectors, the elements of which are the path lengths from the fans to the centers, web pages that do not have a complete bipartite graph structure show a difference in similarity.

Furthermore, we extend the definition of link vectors for complete bipartite link structures to every possible connected link structures. For this extension, we have two things to do. First, in order to be able to determine which pages are fans in the general link structures, we introduce the notion of *base pages*, which work just like fans, while all other pages work just like centers. The base pages are choosen by users or web grouping systems and are considered as a resource for tracing web pages. Second, there are a lot of paths from the base pages to other pages, so we must select the path for the link vector elements. For these paths, we take the shortest path because of the low computing cost and the straightforwardness of the analysis. In spite of these two changes, complete bipartite subgraphs in the entire link structure lead to high similarity because almost all of the centers have the same vector elements if there exists at least one fan on the shortest path from the base pages to the centers. In practice, the fans that have a lot of links to other pages are more likely to be included in the shortest path.

We have described the link vectors for the link structures regarded as directed web graphs. On the other hand, the link structures are also considered as undirected graphs. In undirected graphs, we can follow a link back to the fans from the centers, so that the fans also have link vectors like the centers. As a result, fans make groups with each other.

As mentioned above, we design a link vector for a web page p_i. Let b_j be a base web page and $spl(p_i, b_j)$ be the shorttest path length from b_j to p_i. A link vector $\overrightarrow{L_i}$ for the web page p_i is defined as

$$\overrightarrow{L_i} = \lfloor spl(p_i, b_1), spl(p_i, b_2), \ldots, spl(p_i, b_k) \rfloor \tag{6}$$

where k is the number of web pages. We also define the link vectors for base pages as $spl(b_i, b_i) = 0$ so that base pages could be contained in the groups of web pages. The similarity between web pages p_i and p_j is the normalized dot product excluding the compution of unknown link vector elements, which are the shortest path lengths between a base page and the pages unreachable from the base page in disconnected networks. The unknown link vector elements are not counted in the computation of the normalized dot product between two link vectors. A link vector finds a similarity between two web pages on the basis of their location relative to the base pages in the network.

B. Customized link vectors

Creating link vectors based on our vector space model is so simple that we can design different link vectors in a relatively easy way. In addition to the proposed link vectors, we introduce two other vectors : *word-of-mouth* link vectors and *degree weighted* link vectors.

The *word-of-mouth* link vector model is based on the so-called word-of-mouth influence. In this model, the web pages of senders spread information to adjacent web pages, which then do the same action itelatively. In addition, information from senders may get lost as it is transmited. Let r be the rate at which the information is lost. We define the word-of-mouth link vector elements $wom(p_i, b_j)$ as the degree of influence from sender b_j to web page p_i, namely $wom(p_i, b_j) = r^{spl(p_i,b_j)}$. Similar to Eq.6, the word-of-mouth link vector is defined by

$$\overrightarrow{M_i} = [wom(p_i, b_1), wom(p_i, b_2), \ldots, wom(p_i, b_k)]. \quad (7)$$

If there exists no path between p_i and b_j because of the disconnected link structure, the word-of-mouth link vector is defined as $wom(p_i, b_j) = 0$. It is represents situations where infomation from senders does not come across the disconnected network. In contrast to the simple link vectors, the word-of-mouth link vectors can handle unknown link vector elements. The similarity between two word-of-mouth link vectors is defined by the normalized dot product just like Eq.2. The link vectors are suitable for extracting the groups of web pages that have common information, such as rumor and news, from senders.

Our motivation for introducing another vector, the *degree weighted* link vectors, is that simple link vectors do not consider the number of links that web pages actually have. We combine the degrees, which are the number of links on the page, with the cost to follow a link, and define w_{ij} as the cost to follow a link from page i to page j. Instead of using the breadth-first search algorithm to find the shortest path, we use Dijkstra's algorithm to find the paths with the lowest cost and assign the cost to the elements of the degree weigted link vectors.

The degree weighted link vector is designed for extracting groups consisting of web pages of a hub and spoke structure. Hub web pages have a lot of links, and spoke web pages are those connected to the hub. Let the number of links of page i be k_i and the number of links on page j be k_j. Then, w_{ij} is given by

$$w_{ij} = min(k_i, k_j)/max(k_i, k_j). \quad (8)$$

To avoid grouping hub pages together, Eq.8 emphasizes to the difference in the number of links. The link vector elements for web pages that have a hub and spoke structure have similar values because of low cost w_{ij}. Therefore, the groups expected to be created by this model will consist of some hub pages and the web pages around them.

C. Content-link vector

We propose *content-link* vector model. This model combines document contents and link structures in a unified in the framework of the vector space models. In this model, the lack of link structures in the document vectors is covered by the link vectors. It is expected to be able to extract groups of web pages that have never been found by using only links or contents. Such pages include the web pages that have a common interest represented by links, in addition, have a similar content covering the common interest. This vector uses the information on web pages more effectively than the document or link vectors alone do.

Let $\overrightarrow{D_i}$ be the feature vector for the document content in a web page p_i and $\overrightarrow{L_i}$ be the feature vector of the link structure around p_i. The content-link vector $\overrightarrow{P_i}$ for the document content and the link structure is then defined by

$$\overrightarrow{P_i} = \left\lfloor \alpha\overrightarrow{D_i}, (1-\alpha)\overrightarrow{L_i} \right\rfloor \quad (9)$$

where α, ranging from 0 to 1, is the weight factor for mixing two features. The range of α are from 0 to 1. When $\alpha = 1$, we only use the document contents, while when $\alpha = 0$ we only use the link structure. Then, the similarity $Sim_p(p_i, p_j)$ between web pages p_i and p_j is defined by

$$Sim_p(p_i, p_j) = \frac{\overrightarrow{P_i} \cdot \overrightarrow{P_j}}{|\overrightarrow{P_i}||\overrightarrow{P_j}|}. \quad (10)$$

IV. Experiment

To analyse the behavior of the link vector model, we have applied it to the analysis of the network data of the US political blogoshere. Although this network data consist of directed links, we have considered them as undirected graphs. We have used content-link vectors with $\alpha = 0$, since the document content has not been considered in this experiment. To compute the similarity between web pages, three link vectors have been used: the standard link vector, the degree weighted link vector, and the word-of-mouth link vector with $r = 0.5$. To measure the link vector elements, all pages from the network data have been selected as the base pages, because they have a significant effect on web page grouping. After computing the similarity between web pages, we have clustered them by the complete linkage clustering.

The network data of the US political blogosphere was gathered in a single day from several online blog directories, such as eTalkingHead, BlogCatalog, CampaignLine, and Blogarama. The data have two political leanings, liberal and conservative comming from the corresponding blog directories. Then some blogs are added by retrieving and labeled manually on the basis of their posts and blogrolls. Note that neither directory labels, which often rely on self-reported or automated categorizations, nor manual labels are 100% accurate. The total number of gathered blogs was 1494 with 739 liberal and 735 conservative. Of the links originating from one of those communities, 91% stayed within that community. In our experiments, since we have grouped web pages with link structure, web pages with no links did not join other pages. Therefore, only 1222 blogs which had at least one link have been grouped.

In the rest of this section we first show a typical result of web page grouping with the standard link vectors. We then discuss the effectiveness from the results using our three link vectors. Finally, we compare the link vector model with other grouping methods: the Girvan and Newman method, Newman clustering, and HyPursuit.

A. Results with link vectors

The standard link vector model extracts groups consisting mostly of liberal or conservative blogs. As a typical result, we show the component graph (See Fig.2) and snapshot image (See Fig.3) of the resultant group in the situation with max Hubert and Arabie's adjusted Rand index[9].

Fig.2 shows the group sizes and propotions of political orientations. The group is visualized in Fig.3, in which the liberals are depicted as squares and the conservatives are depicted as circles. The vertices with the same darkness forms the same cluster.

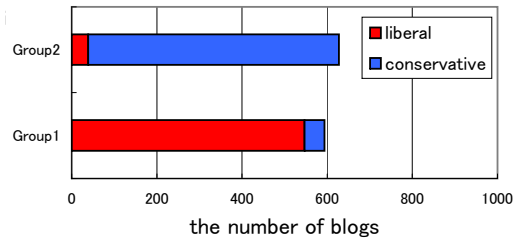

Fig. 2.

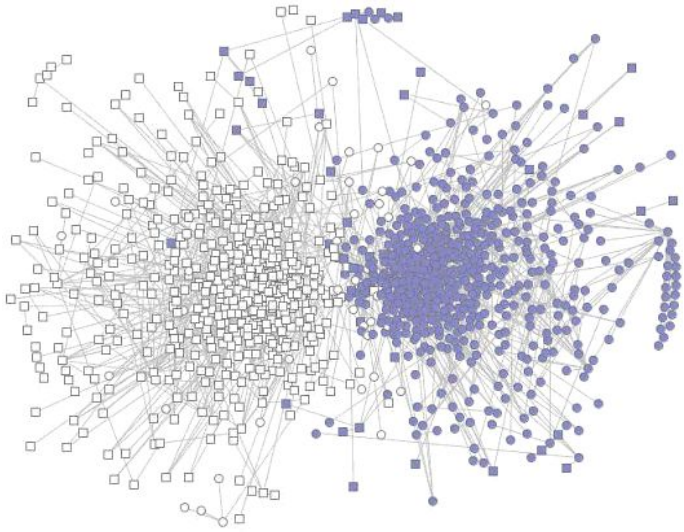

Fig. 3. The result with the simple link vectors grouped by the complete linkage clustering at 1221 steps of grouping

With the link vectors, we have got two groups as shown in Fig.2 : more than 90% of the pages in the Group 1 (Group 2) are liberals (conservatives). In the begining of the process of clustering, the groups were built from web pages around some hubs, because the link vector elements of grouped pages, especially those having just one link, were given almost the same values from the hubs. In the middle of the process, we found some fans in the complete bipartite subgraphs. In addition, the groups in relaxed complete bipartite graph strucutes were extracted. We show an example of such a group in Fig.4. Such a group is not created from complete bipartite graphs, but some web pages have common citations to other web pages. Even though the hub page gives the same values to the link vectors of its surrounding web pages, some web with common citations have made them grouped together. Toward the end of the process, a lot of liberal or conservative web pages that were grouped in the middle steps have been unified together. Because the complete linkage algorithm tend to remain unmerged hub pages and groups arround them were not unified until the final steps. At the end of the process, they are grouped together as shown in Fig.3. In the entire process, the link vector model has shown effective results in the sense that almost all pages in each group are either liberals or conservatives. This means that we have successfully distinguished the two groups from each other.

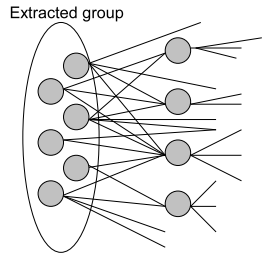

Fig. 4. A group created from a relaxed complete bipartite graph

While link vectors led to effective web grouping, the similarities take almost the same values close to 1. This is due to the small world phenomenon on the Web that the diameters of link structures are usually at most 10. Namely, the pages that are less than 10 links apart from each other have almost the same similarity. Because of this matter, when we consider to combine link vectors with document contents, its result might have little effect for grouping. In this case, we can set a greater value to α in the content-link vectors in order to have the content information to play a more important role.

In addition to standard link vectors, the word-of-mouth and degree weighted link vectors could also extract groups.

The word-of-mouth link vectors have found the groups of the hub page and the web pages arround it. In contrast to the standard link vectors, the hub page is included in these groups. These results show that the information from hubs are transmitted more easily. In other words, the common information is more likely to be delivered from the hub senders. Especially, the web pages that have only a few links are more frequently influenced from the senders and they are grouped at the begining of the clustering process.

With the degree weighted link vectors, the web pages that have the hub and spoke structures have been successfully grouped together. In the clustering process, hub pages containing larger number of links and web pages around them

were grouped together. Then, some web pages that have links to more than two hubs were extracted independently of their hubs. Groups extracted at the begining are marged with other smaller groups or pages. Furthermore, the link vector model has found different groups, in which the web pages link to both liberal and conservative hub pages.

B. Comparison with other grouping methods

We discuss the results for comparing the standard link vector model with other web grouping methods.

In the Girvan and Newman method, we got two groups of higher degree web pages belonging to either liberals or conservatives. When we go to the web page having a few links from other pages, we have to pass these links so that they give higher betweenness. While the link vector model extracted the web pages around hub pages, this method led to groups of hub pages.

The Newman clustering based on the modularity showed an effective performance in extracting liberal and conservative groups. In this clustering, the clustered web pages were almost well connected with each other. The liberal group was extracted at the beginning, and the conservative group was extracted after that. The groups were extracted from the web pages that had moderate number of links and they were getting larger toward the end of the process. While the Newman clustering and the Girvan and Newman method extract only political leanings, our link vector model also extracts local groups on the basis of link structures such as relaxed complete bipartite subgraphs, groups that have a common reputation and hub and spoke structures.

HyPursuit clustering only consider undirected link structures. It gives more priority to web pages at the middle position of the graph diameter than hub pages. As a result of this clustering, the web pages that link to both side of political blogs are grouped together in the begining, then the hub pages and web pages that have certain number of links around them are extracted. This clustering has a possiblity of finding the hub pages and web pages of the middle position, but it does not always lead to better performance for combining with document contents. Because our link vectors are based on complete bipartite graphs, they cover the performance of document vectors with the proper manner.

V. CONCLUSION

In this paper, we have proposed some feature vectors based on the vector space model: link vectors for considering link structures effectively and content-link vectors for characterizing both document contents and link structures in a unified way. Then we have applied the link vectors to the network data of the US political blogoshere and empirically showed that the link vectors led to the appropriate groups. In the comparison with the Girvan and Newman method and the Newman clustering, the link vectors led to the grouping of both political orientations and local link structures, while the other methods extracted only two groups of liberals and conservatives. The link vectors also show the difference from HyPursuit in using the link structure for combining with the document content.

As our future work, we will improve the link vectors so that they do not introduce almost the same similarity. Meanwhile, we will implement the document vectors and complete our content-link vectors and apply this to some other network data on the Web to confirm better performance and find various groups that have never found by using link structures or the document contents alone.

REFERENCES

[1] A. Gulli, A. Signorini. (2005, Jan.) The Indexable Web is more than 11.5 billion pages. [Online]. Available: http://www.cs.uiowa.edu/˜asignori/web-size/

[2] V. V. Raghavan and S. K. M. Wong, "A critical analysis of vector space model for information retrieval," Journal of the American Society for Information Science, Vol.37 (5), 1986, pp.279-287.

[3] R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, "Trawling the web for emerging cyber-communities", Proceedings of the 8th WWW conference, 1999.

[4] M. E. J. Newman, M. Girvan, "Finding and evaluating community structure in networks," Physical Review E, Vol.69, No.026113, 2004.

[5] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," Physical Review E, Vol.69, No.066133, 2004.

[6] A. Clauset, M. E. J. Newman, C. Moore., "Finding community structure in very large networks," Physical Review E, Vol.70, No.066111, 2004.

[7] R. Weiss, B. Vóelez, M. A. Sheldon, C. Namprempre, P. Szilagyi, A. Duda, D. K. Gifford, "HyPursuit: A Hierachical Network Search Engine that Exploits Content-Link Hypertext Clustering," Proceedings of the 7th ACM Conference on Hypertext, 1996, pp.180-193.

[8] L. A. Adamic, N. Glance, "The political blogosphere and the 2004 US Election," Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem, 2005.

[9] F. Boutin, M. Hascoët, "Cluster Validity Indices for Graph Partitioning," Proceedings of the Conference on Information Visualization IV', 2004.