On Ranking and Organizing Web Query Results

Okan Kolak* Wen-Syan Li

C&C Research Laboratories
NEC USA, Inc., 110 Rio Robles, M/S SJ100
San Jose, CA 95134, USA
Email: {okan,wen}@ccrl.sj.nec.com

Abstract

With the explosive growth of the WWW, most searches retrieve a large number of documents. The query results returned have no aggregate structure; thus, it is difficult to forage for relevant pages. We argue that organization of Web query results is essential to the user in terms of usability of query results. In addition, how the query results are organized has a great impact on the ranking scheme and query processing as well as search engine usability for the users. In this paper, we present various schemes on ranking and organizing Web query results in the scope of `NetTopix` *search engine project in NEC Research Laboratories in San Jose. We describe three associated tasks: Fulltext search and initial ranking, post query processing, and query result reorganization for presentation. We discuss our consideration and pros and cons of each design alternative.*

Keywords: Fulltext search, search engine, query result organization, WWW, ranking

1 Introduction

With the explosive growth of the WWW, most search queries retrieve a large number of documents. For a typical query, such as "NEC", most of the search engines return millions of documents. Without aggregate structure and organization, it is difficult for users to forage for relevant pages. From database technology point of view, all documents containing "NEC" are considered correct answers. However, from the points of view of computer human interaction and information retrieval, to further organize and rank all of these "correct" answers based on their relevance results in a significant impact on the utility of the underlying search engine.

One difficulty is that the users may have different purposes for searching the documents related to "NEC". For example, one user may want to find the *documents* which are relevant to NEC. Another user may be interested in finding the *Web sites* which have more information related to NEC. The other type of users are who, although they only type in "NEC" for queries, are actually interested some specific information related to NEC. For example, one user may want to search for information of NEC laptop computer while the other user may be interested in NEC's role in education. In this case, the users may prefer the system to *categorize* the documents based on their semantics. To serve these multiple purposes of search, the search engine requires high flexibility to rank and organize the query results according a wide variety of user requirements.

In this paper, we point out that organization of Web query results is essential to the user in terms of usability of query results. In addition, how the query results are organized has a great impact on query processing and the ranking schemes. We present several schemes for ranking and organizing Web query results in the scope of `NetTopix` search engine project in NEC C&C Research Laboratories in San Jose. In Figure 1, we show the main menu of `NetTopix` search engine. When the users pose a query, say "NEC", `NetTopix` provides various ranking and organization schemes for the results. Figure 2 shows that query results are organized by fulltext search result score. Figure 3 shows that alternatively query results can be organized by host name. Note that such organization is different from sorting by URL. Since the full text search engine only returns the scores per document basis, it requires additional post query processing to cluster the results by host name and rank the query result per host name basis. In Figure 4 we show another organiza-

*This work was performed when the author visited NEC, CCRL. The author is currently a Ph.D. student at Department of Computer Science, University of Maryland.

Figure 1: Query and result organization option specification menu in NetTopix

tion scheme by which the query results are organized by category. The figure illustrates several categories, such as "Education" and "Science". In the same way that category-based organization requires post query processing. The categories are ranked by their relevance tot he query terms. Similar to the ranking and organization by host name, it requires post query process to provide such functionalities.

The rest of paper is organized as follows. We first describe fulltext search schemes and initial ranking methods in Section 2. Then, we describe two post query processing schemes for result organization and re-ranking: by host/domain in Section 3 and by category in Section 4. We discuss our consideration as well as advantages and disadvantages of each design alternative.

2 Fulltext Search

`NetTopix` utilizes `JTOPIC`[4] as its full text search engine. We have configured `JTOPIC` to support the following functions: (1) search by both keyword and phrase; (2) force inclusion and force exclusion so that keyword/phrase must be/must not be in the document; (3) query term stemming (i.e. car and cars are treated as the same query term unless specifying otherwise); and (4) case sensitive option.[1]

In addition to these popular options provided by most search engines, `NetTopix` also allows users to specify the importance of each query term. For example, a query "NEC:3 Japan:1" specifies that the word "NEC" is three times more important than the word "Japan". This function allows the documents which contain only "NEC" be included and ranked higher than those documents containing only the word "Japan".

The major difference between Web documents and regular documents used in the IR field is that Web documents have a structure. That is, Web documents have many parts, including URL, title, body, link, anchor, highlighted word with special font, etc. Different parts of the document have different weights associated with them. In general, we can judge that a word appearing in the title is more important than the same word appearing in the body. We currently assign the weights to the position a word appears in a HTML documents as:[2]

Position	Weights
URL	100%
Title field	90%
Meta Keyword field	70%
Meta Description field	60%
Body	50%

The score of fulltext search depends on keyword occurrence frequency in a document. The higher the number of occurrences of keywords, the higher the document score. The score also depends on the relative size of the document. A document of 100 words with 1 keyword in it gets a higher score than a document of 1000 words with 1 keyword in it.

[1] If search terms are all lowercase or all upper case, search is case insensitive. If the terms are mixed case, search is case sensitive.

[2] These numbers are experimental and they perform better than using other numbers based on our preliminary study. A larger scale of formal study based on user feedback log is necessary.

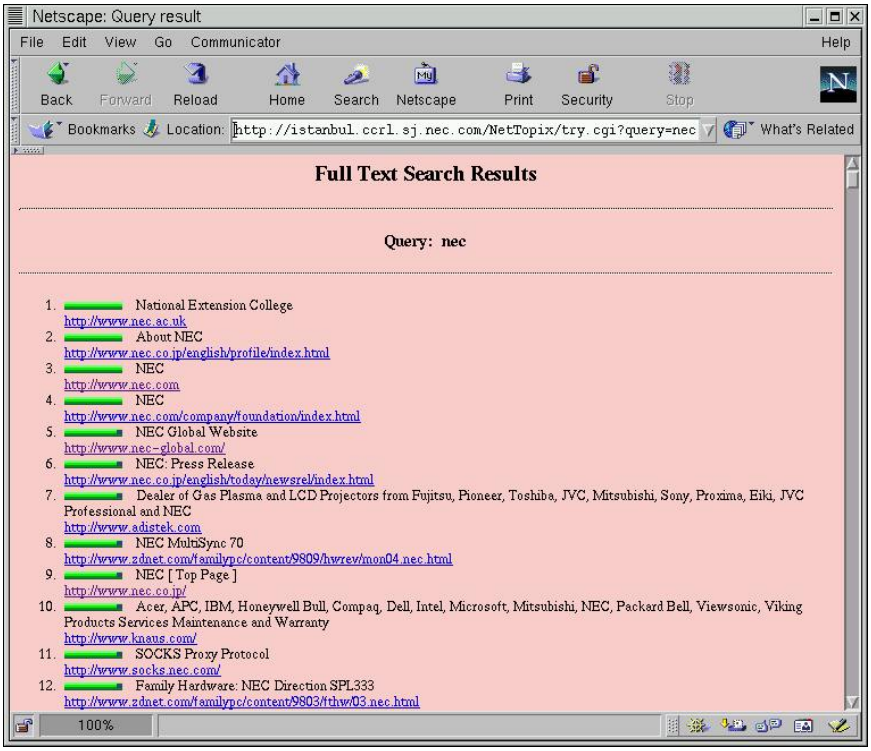

Figure 2: Fulltext search results

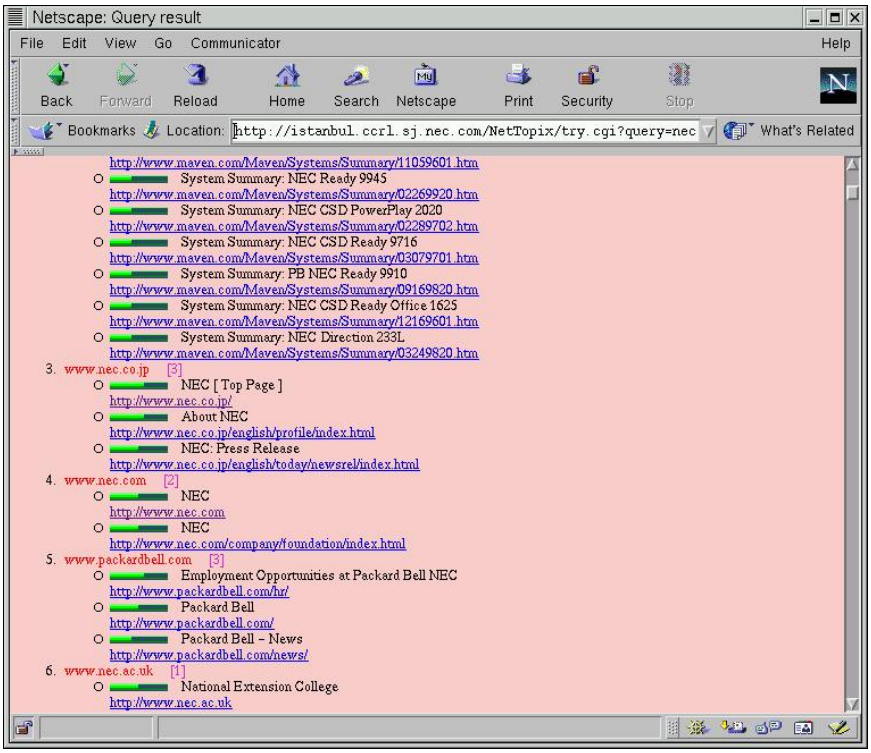

Figure 3: Query results organized by domain

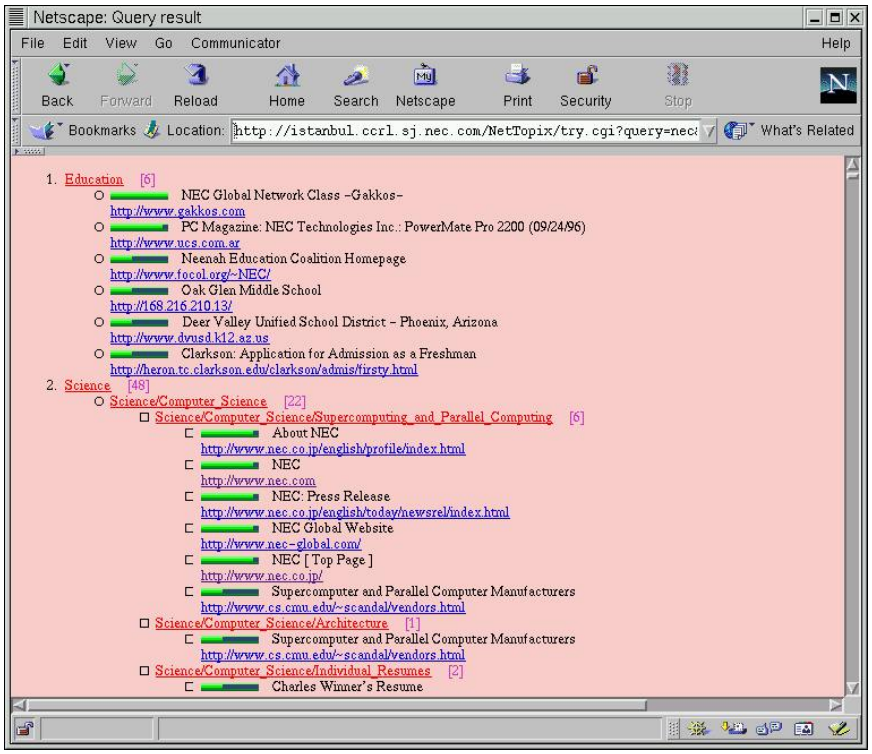

Figure 4: Query results organized by category

Based on the principles and configurations described above, `JTOPIC` returns the documents above a given threshold as a ranked list. In Figure 2, we show a window dump for query results which are organized in the order of fulltext search scores. Such scores are the basis for other post query processing for organization and ranking later if users specify. In the next section, we explain the post query process.

3 Organization and Ranking by Domain

In this type of result organization, results are grouped based on their Internet domains (i.e. host name such as www.nec.com) and sorted within groups in descending score order. Domain scores are calculated based on the scores of the matching documents in that domain, and then the query results within a domain are sorted in descending score order. In this section, we describe three tasks for organization and ranking by domain in detail as follows.

3.1 Clustering query results

We have explored the following three methods to cluster query results by their URLs.

3.1.1 Domain Based Clustering

All documents from a particular Internet domain are grouped into a cluster. Advantage of this method is its simplicity; however, a big drawback of this method is that big sites, such as Geocities[15], AOL[1], tend to get a lot of matches due to vast number of documents that they contain. This leads to a few unorganized clusters, which are hard for the users to distinguish relevant documents from others.

3.1.2 Logical Domain Based Clustering

As a matter of fact, many large Web sites, such as Geocities[15], AOL[1], and BigLobe[10] by NEC, are usually either Internet Access Provider sites or Web site hosting providers. Many sites with URL prefix of these Internet Access Provider site URLs are actually "logical domains" by themselves. Figure 5 shows a portion of link structure through two NEC Web site. If we search with query terms "Web" and "research", we may find thousands of pages in the two sites. Although the results are organized by their domains, the reality is that such a presentation is almost the same as the presentation in which results are simply sorted by score. To improve the organization of results by domain, we define *logical domains* within

physical Internet domains, such as `www.nec.com` and `www.ccrl.com`. Logical domains are identified based on directory structure and link structure between the pages. For instance, in web sites, user pages are usually located at a particular directory for each user, and users in general design an entry page (e.g. index.html) that has links to other pages of that user, and other pages tend to have a 'back' or 'home' link to this entry page. Using heuristics like these, it may be possible to identify logical domains automatically. In Figure 5, we may identify that the logical domains as `www.nec.com`, `www.ccrl.com`, `www.ccrl.com/hypermedia/`, and `www.ccrl.com/middleware/` and query results are clustered based on these logical domains. The detail of logical domain identification techniques are described in [8].

3.1.3 Improved Domain Based Clustering

Note that identifying logical domains is not an easy task. In many cases, whether or not a URL is a logical domain depends on the threshold set or judgment of the system designers. We explore another clustering scheme by imposing a limit on the maximum number of documents allowed in a cluster. Let $PathLen$ denote the number of directories within URL that will be effective in clustering. Clustering is performed in the following two steps: (1) With an initial value of 0 for $PathLen$, cluster all documents by comparing their domains and the first $PathLen$ directories after the domain; and (2) if any cluster contains more documents than allowed, divide the documents in that cluster into smaller clusters by using $PathLen + 1$ as new $PathLen$ value for that cluster.

To illustrate using an example, let's set the maximum number of documents in a cluster as 5, and assume that we have 3 results from `www.nec.com`, and 7 results from `www.ccrl.com`. At first pass, all documents in www.nec.com will form one cluster, and all documents in `www.ccrl.com` will form another, since $PathLen$ is 0, meaning we do not consider any directory, only the domain. After this pass, first cluster is formed and finalized since it contains 3 documents. But second cluster needs to be subdivided since it has 7 documents. Now we use the same method, with $PathLen = 1$. In this case we look at the first directory in URL too. Lets say we have 4 documents with URL `www.ccrl.com/hypermedia/*` and 3 documents with `www.ccrl.com/middleware/*`, which will form the new cluster in this case. And now all clusters have less than 5 documents, so clustering is achieved. This approach somewhat solves the problem with large Internet Access Provider sites, such as Geocities, which has a vast amount of documents, and a relatively shallow directory structure. However, time sites like `zdnet.com`, which has a deeper directory structure with many a modest number of document in each directory is problem. Algorithm manages to find a node deep in directory tree with some number of documents less than the maximum. However, when we look at the results for query "NEC", we see that the pages in `znet.com` talking about NEC products are ranked higher than NEC pages, which means results are still bad.

To summarize, we observe that path information is not sufficient for identifying proper scopes that are used to define "logical domains". We are investigating a new approach to integrate the above two schemes by examining both path information and link structures. propagating scores based on that locality definition. Using link information might be a better way to define clusters and locality.

3.2 Locality Analysis

The second step of post query processing is to perform locality analysis based on the directory structure to adjust the scores of documents. One of the motivations is that if there are more good documents colocated within a region in a domain, it is more likely that those documents are of good value to the users than those documents which are scattered among a domain. The system first groups query results to clusters by domain analysis as described above. The score of each document is propagated to neighbor documents. The score propagation scheme is as follows. Let C denote a cluster, $D(i,j)$ denote the distance between the documents i and j based on the directory structure, $orig\_score(i)$ denote the original score for the full text search of the document i, and α, with a value between 0 and 1, denote decaying factor for propagation scores. The adjusted score, $adj\_score(i)$, for the document i is

$$adj\_score(i) = \sum_{j \in C} orig\_score(j) \times \alpha^{D(i,j)}$$

After the scores of all documents in all clusters are calculated, a global normalization is performed, since some scores may go above 1 due to propagation of scores from neighbors.

By selecting 0.5 for α, this formula allows results mutually reinforce their scores within radius of 5 based on the directory structures.[3] Advantages of ranking

[3] The results which are 5 or more directory path away can increase less than 3% of the scores of other results.

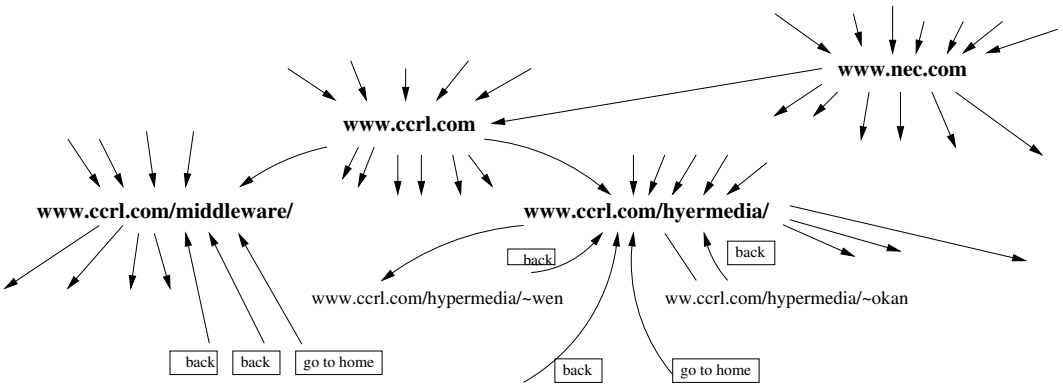

Figure 5: Identifying Logical Domains

by path (i.e. directory structure) is that the path information is always available from the URLs. For a given set of documents from a particular domain, the distance between them can be calculated easily. On the other hand, since the underlying directory structure is not visible to the viewer of the web pages, authors are not as careful and as consistent with directory structure, as they are with link structure within the documents. Consequently, the directory structure may not be very meaningful. One author may put all documents in a single directory, which leads to high locality, whereas another author may store some documents with very similar content in a deep directory hierarchy, which leads to lower locality. This difference in directory structure is not even visible to the end user, which leaves authors with almost complete freedom on arranging their directories.

To perform locality analysis, we may use link structures or directory structures. The score adjustment by the link-based locality analysis is similar to the topic distillation techniques discussed in [3, 6, 11]. The advantage of using links for locality analysis is that the link structures are in general more accurate measurements than paths. However, link structures may not be always available unless the documents in the specific Web site are crawled and indexed. furthermore, it is more expensive to perform topic distillation based on link as some performance statistics indicates in [2].

3.3 Ranking the Domains

After the query results are clustered by domain and their scores are adjusted accordingly, the system needs to rank clusters. Since each cluster contains multiple documents, we are currently using the average scores of the clusters for ranking. The final result presentation by domain for the query "NEC" is shown in Figure 3.

4 Organization and Ranking by Category

In this type of organization, the system groups the results into categories and presents the results to the users. Currently `NetTopix` utilizes external classifiers for document categorization. Many Internet search engines, such as InfoSeek[7] and HotBot[13], can be utilized as an external classifier. Some classifiers based on librarian categories, such as the well known Library of Congress Classification (LCC)[9], are also suitable for this purpose. The classification shown in this paper is based on Yahoo[14]. The details of our classification scheme, similar to Pharos[5] which is based on LCC, are described in [12].

For example, the query results for "nec" may be classified into 38 categories as shown in Figure 6 when such documents were collected into `NetTopix`. Such classification is carried out in advance during the indexing phase. Note that in Figure 6 the categories are formed as a seven level tree structure to support very fine classification. However, if we present the results organized by such fine categories, as shown in Figure 7, it may not be easy for users to browse through this long list of fine categories to find relevant ones.

We explore another clustering scheme by imposing a limit on the maximum number of documents allowed in a cluster. Let $CategoryPathLen$ denote the number of category paths of a classification category by which query results can be effective classified. The results shown in Figure 7 are reorganized following steps: (1) With an initial value of 0 for $CategoryPathLen$, group all documents in this category; (2) if the cluster contains more documents than allowed (e.g. 20), further classify the documents into multiple finer categories by using $CategoryPathLen+1$ as new $CategoryPathLen$ value for that cluster; and

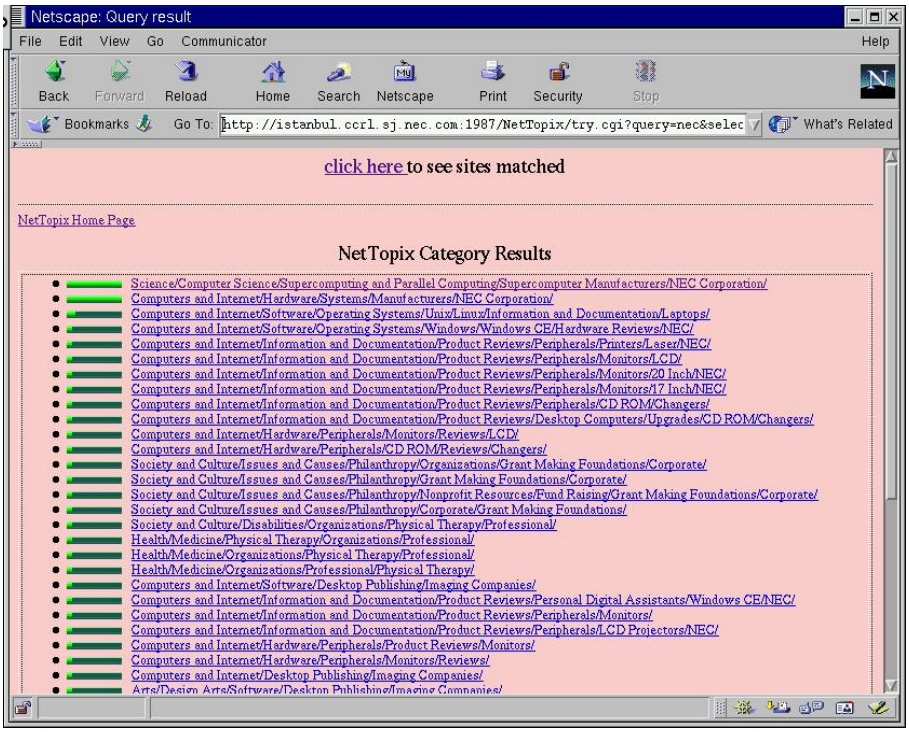

Figure 6: Categories related to "nec"

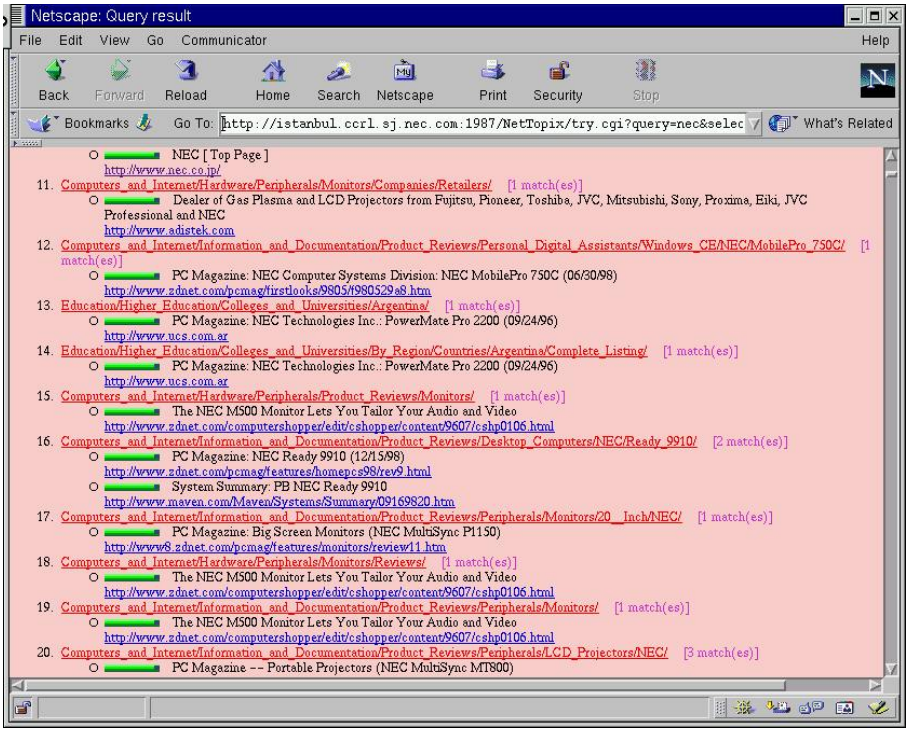

Figure 7: Categories for document containing "nec"

(3) repeat steps 1 and 2 until each category contains less than 20 documents or it can not be divided any further. In each level of category, the system further sorts the documents and/or categories by their average scores as described in Section 3.3. After the reorganization, the result presentation, as shown in Figure 4 is much easier for users to visualize and locate relevant categories and documents.

5 Concluding remarks

With the explosive growth of the WWW, most search engines retrieve a large number of documents. These search engines do not organize query results as aggregate structure; thus, it is difficult to distill relevant pages. With this motivation, we explore various organization and ranking schemes to provide higher usability of query results. Testing these schemes on `NetTopix`, which has a large collection of Web documents, we observe that path information alone may not be sufficient for identifying proper scopes that are used to define "logical domains". We also observe that the URL strings, especially the host name portion, are very effective in organizing results. One of the main reasons is that the host names are regulated and thus they tend to more meaningful. We are investigating a new approach to integrate the above two schemes by examining both path information and link structures. Our initial experience suggests that organizing query results by category provides a simple and useful way for users to locate target documents.

We would like to point out that there have a number of competing efforts on the Web. A comprehensive comparison is necessary to show the effectiveness of our approaches. It is however to be acknowledged that most of these efforts are completely unpublished due to commercial confidentiality.

References

[1] America Online, Inc. *Information available at http://www.aol.com/.*

[2] Krishna Bharat and Monika Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *Proceedings of the 21th Annual International ACM SIGIR Conference*, pages 104–111, Melbourne, Australia, August 1998.

[3] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text. In *Proceedings of the 7th World-Wide Web Conference*, pages 65–74, Brisbane, Queensland, Australia, April 1998.

[4] NEC Corporation. *JTOPIC Developer's Kit.* NEC Corporation, December 1997.

[5] R. Dolin, D. Agrawal, A. El Abbadi, and J. Pearlman. Using Automated Classification for Summarizating and Selecting Heterogeneous Information Sources. *D-Lib, Information available at http://www.dlib.org/dlib/january98/dolin/01dolin.html*, January 1998.

[6] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Inferring Web Communities from Link Topology. In *Proceedings of the 1998 ACM Hypertext Conference*, pages 225–234, Pittsburgh, PA, USA, June 1998.

[7] Infoseek Corporation. *Information available at http://www.infoseek.com/.*

[8] Wen-Syan Li, Okan Kolak, Quoc Vu, and Hajime Takano. Defining Logical Domains in a Web Site. In To appear in *Proceedings of the 2000 ACM Hypertext Conference*, San Antonio, Texas, USA, June 2000.

[9] Library of Congress. LC Classification Outline, fifth edition. 1986.

[10] NEC Corporation. *Information available at http://www.biglobe.ne.jp/.*

[11] Lawrence Page and Sergey Brin. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th World-Wide Web Conference*, Brisbane, Queensland, Australia, April 1998.

[12] Quoc Vu, Wen-Syan Li, and Edward Chang. On Constructing Personalized Navigation Trees for Web Documents. In *Proceedings of the 8th World-Wide Web Conference*, pages 94–95, Toronto, Canada, May 1999.

[13] Wired Digital Inc. *Information available at http://www.hotbot.com/.*

[14] Yahoo Communications Corporation. *Information available at http://www.yahoo.com/.*

[15] Yahoo Inc. *Information available at http://www.geocities.com/.*