Yonsei/ETRI at TREC-10: Utilizing Web Document Properties

Dong-Yul Ra, Eui-Kyu Park and Joong-Sik Jang
Computer Science Dept., Yonsei University
{dyra, ekpark, lunch}@dragon.yonsei.ac.kr

Myung-Gil Jang and Chung Hee Lee
Electronics and Telecommunications Research Institute
{mgjang, forever}@etri.re.kr

Abstract

As our first TREC participation, four runs were submitted for the ad hoc task and two runs for the home page finding task in the web track. For the ad hoc task we experimented on the usefulness of anchor texts. However, no significant gain in retrieval effectiveness was observed. The substring relationship between URL's was found to be effective in the home page finding task.

1. Introduction

This is the first time that our group, Yonsei University and ETRI, participated in the TREC conference. We participated in the web track for both the ad hoc and home page finding tasks. We developed an IR system based on natural language processing according to our original aim. But we could not carry out enough experimentation to draw any conclusion on a NLP-based system. In this paper we will talk about two aspects of a web document retrieval system: taking advantage of the anchor texts of the hyper links and using the substring relationship of URL's in home page finding.

Many reports in TREC-8 and 9 said that the link connectivity itself did not help much to improve the retrieval effectiveness[5,6,8,9]. There have been some suggestions of using the anchor texts of the links[1,2,7]. We thought that a link's anchor text may give some hint on what the document that the link points to is about. As an ad hoc task we developed a system to pursue this issue. The experimental result showed that even the use of anchor texts does not improve the retrieval effectiveness significantly.

We also produced runs related to the home page finding task. What we experimented with this task is the usefulness of the URL substring relationship in finding the home page, i.e. the web site entry page. We have found that if the URL of a document is a prefix of that of another document (where both documents seems to have some relevancy to the topic) the former document is more likely to be the home page than the latter. The experimental observation indicates that the substring relationship of URL's is a good source of information to raise the reciprocal rank (RR) for the home page finding task.

2. Overview of the system

Our system uses a natural language analysis component as the front end for both indexing and retrieval. It consists of morphological analysis, part of speech tagging and the context-free parsing modules. A two-level model is used for morphological analysis. Part of speech tagging is based on the Hidden Markov Model. The bottom-up chart parsing technique is used in the parsing module. It is a shallow parser whose major objective is to find verbs and arguments associated with them. The result of parsing is used to produce the head-modifier index terms whenever it is possible.

The vector space model forms the basis of our system. The index terms can be either key words or a

pair of words in head-modifier relationship. Having head-modifier index terms made the number of total index terms huge, which slowed down the speed of the system. The size of the inverted file has grown up to the level that the file system could not handle. This problem was solved by storing the inverted file in several files. This is different from the approach of distributed IR. Our system was developed on the PC of server level with 1GB of memory and 60 GB of disk space. The major amount of time was spent in storing index terms in the indexing storage rather than doing natural language analysis.

This is the first time that we participated in the TREC. We experienced much difficulty in producing the result on time and made some mistakes in the creation of the runs that were submitted for assessment. One non-trivial mistake is that no relevance feedback was done. This might be one of the reasons for coming up with rather low average precision. We hope that we can have better systems by not making mistakes.

3. Experiments in the ad hoc task on usefulness of anchor texts

In this section what we did for the ad hoc task in the web track is explained. We used the typical vector space model for indexing and retrieval. But we tried to make use of information that only web documents can provide. The results of experiments done in the previous TREC conferences pointed out that the use of hyper links does not lead to a noticeable improvement in retrieval effectiveness. But most of the approaches so far just tried to use the information given by the connectivity among documents.

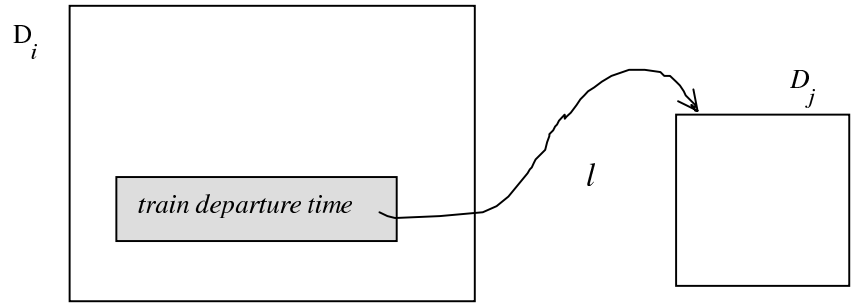

Fig. 1: An anchor text

We thought that the anchor text on which the link is set might be a good source of information.[1] In Fig. 1, the hyper link *l* connects document D_i and D_j. The anchor text of the link is "*train departure time*". What this anchor text says is that one needs to consult the document D_j to know about *train departure time*; one can find some information about *train departure time* by following the link and reading the document D_j.
Even though the document D_j does not contain any key words indicating relevancy to the topic of train departure time it is likely that the document is relevant to the topic.

Thus the content of a document is reflected in some degree in the anchor texts of the incoming links of the document. But this document receives no contribution to the indication of its content from its outgoing links in our approach. We do not use any information from connectivity such as Kleinberg's scheme except the anchor texts[3].

We cannot consider the links of all documents in the collection because it takes too much time. Let *C* be the whole collection of the documents. The consideration of links is confined to the documents retrieved for the query by

[1] After we started development with this aim, we found later that several organizations pursued this issue independently[1,2,7].

Table 1: Performance of our system in the ad hoc task

Run id: yeaht01	Run description: automatic, title-only, link(anchor text)		No. of topics: 50
Total number of documents over all topics			
Retrieved: 44922	Relevant: 3363	Relevants retrieved: 1337	

Recall level precision averages		Document level precision averages	
Recall	Precision	Recall	Precision
0.0	0.6152	At 5 docs	0.3880
0.1	0.3619	At 10 docs	0.3240
0.2	0.2511	At 15 docs	0.2800
0.3	0.1820	At 20 docs	0.2520
0.4	0.0998	At 30 docs	0.2180
0.5	0.0616	At 100 docs	0.1282
0.6	0.0286	At 200 docs	0.0830
0.7	0.0225	At 500 docs	0.0473
0.8	0.0200	At 1000 docs	0.0267
0.9	0.0200		
1.0	0.0200		
Average precision (non-interpolated) : 0.1286		R-precision (exact) : 0.1796	

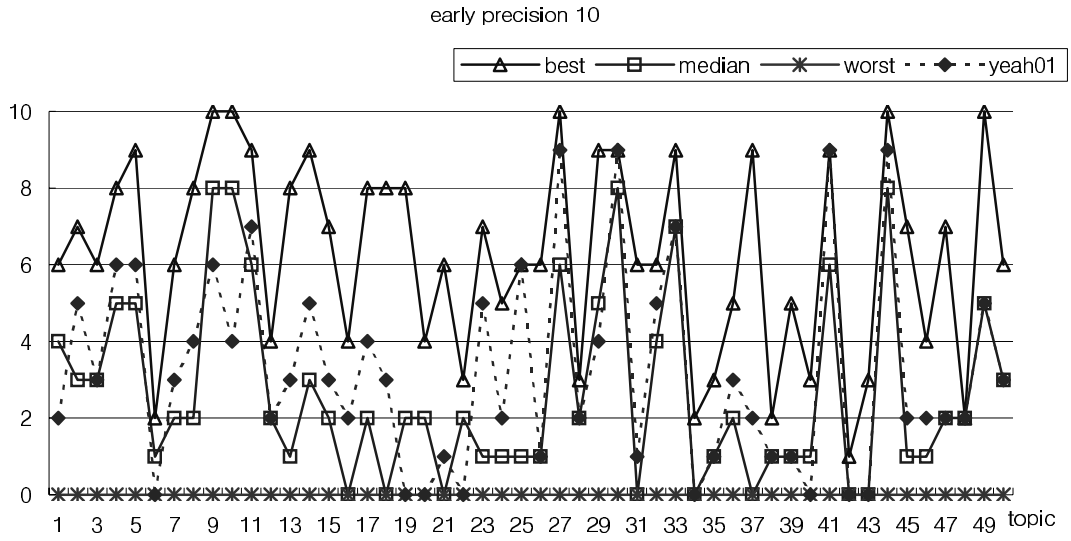

Fig. 2: Performance comparison with others

the typical retrieval engine based on the vector space model[4]. (Let us call it the base set B of the retrieved documents.) The extended set E of the retrieved documents is obtained as follows. Here, $\Phi(l)$ indicates the anchor text of link l; Q denotes the query; $\text{sim}(Q, \Phi(l))$ is the similarity[2] between Q and $\Phi(l)$ returned by the retrieval

[2] The score sim(A,B) stands for the cosine similarity value between the vectors of two texts A and B returned by the vector space model.

engine:

(i) $E \leftarrow B$;

(ii) For each document d_i in E,

add d_j to E if there is a link l out of d_i pointng to d_j and sim(Q, $\Phi(l)$) > 0 ;

Then the score of each document in E is computed again by the following two methods that are different in some way.

- Method link1: The new relevancy score of each document in E is computed as follows:

$$RSV(d) = sim(Q,d) + \alpha \sum_{l \in inlink(d)} sim(Q, \Phi(l))$$

where *inlink(d)* is the set of incoming links to document *d*. The parameter α is the weight given to the contribution of the anchor texts. It is determined by experiments.

- Method link2: In this method the anchor text is regarded as the part of the text of the document.
  (i) We add all anchor texts of incoming links of every document in the extended set *E* as a part of the document.
  (ii) We do indexing on a document including the anchor texts. (However a special scheme is used to include only the anchor texts of the incoming links from the documents in the base set *B*.)
  (iii) The similarity score returned by the vector space model is used for obtaining the final ranked list.

The final ranked list of retrieved documents is obtained by ordering the documents in *E* based on the *RSV* of each document. We could not submit the official runs using Method link2 because of the tight schedule.

One can see the performance of our system in the ad hoc task of the web track in Table 1. Early precision seems to be important in IR systems. The comparison with other systems in this measure can be seen in Fig. 2. This shows that our system is near median. Table 2 shows the difference made by the use of anchor texts. The run yeaht01 (automatic, title only, use of anchor texts) does not have any significant improvement from the run yeahtb01 (automatic, title only, no use of anchor texts).

Table 2: Effectiveness of the use of anchor texts

Recall	Average precision	
	yeahtb01 (no use of links)	yeaht01 (use of links)
0.0	0.6086	0.6152
0.1	0.3618	0.3619
0.2	0.2534	0.2511
0.3	0.1796	0.1820
0.4	0.1002	0.0998
0.5	0.0618	0.0616
0.6	0.0286	0.0286
0.7	0.0225	0.0225
0.8	0.0200	0.0200
0.9	0.0200	0.0200
1.0	0.0200	0.0200

4. The use of substring relationships of URL's for home page finding

A document of a home page (the entry page) of a web site has the same format as other web pages. There is no information or marks attached to the web pages indicating whether it is a home page or not. Thus it is not easy to locate a home page for a web site search query.

We use a heuristic to cope with this problem. There is a tendency that if a home page D_h has an outgoing link to a page D_i and D_i is stored physically in the same server as the home page then the URL string of D_h is a substring (actually a prefix) of D_i's URL.

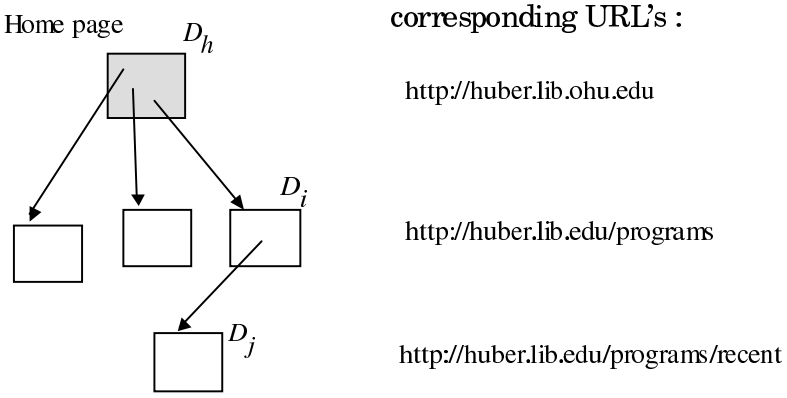

Fig. 3 : Web pages and their URL's

Those pages that are descendants of the entry page D_h will have a tendency that their URL's contain the URL of D_h as shown in Fig. 3. As an example let us assume that the home page finding query is "Huber Library" and retrieval process explained in the previous section produces the following ranked list of documents:

D_j (http://huber.lib.edu/programs/recent) : 17.5
D_i (http://huber.lib.edu/programs/) : 14.3
D_h (http://huber.lib.edu) : 11.8

It is likely that the bottom-most document D_j contains the words "Huber" and/or "Library" more number of times than its ancestors D_i and D_h. Thus the score of D_j is highest. Since URL of D_h is a substring of that of the document D_j in the retrieval list it is given some bonus point , say 4. D_h gets the bonus once more because of D_i by the same reason. Thus the score of D_h will be increased to 19.8. Similarly D_i gets the bonus point of 4 because URL of D_j subsumes that of D_i. But D_j gets no bonus because there is no document whose URL string contains that of D_j. As a result the final score and the ranked list is as follows:

D_h : (http://huber.lib.edu) : 19.8
D_i : (http://huber.lib.edu/programs/) : 18.3
D_j : (http://huber.lib.edu/programs/recent) : 17.5

We take advantage of this observation explained so far to move a home page up in the ranked list (which was

called *E*) of the retrieval result. We apply the following heuristic for all the pages in the final ranked list *E* (produced in the ad-hoc processing explained in the previous section):

> (i) Every document *d* in *E* gets extra bonus point (added to the existing score) whenever there is a document *b* in *E* such that URL of *d* is a part (substring) of URL of *b*;
> (ii) After this processing is done for all the documents in *E*, they are reordered by the new scores.

We submitted two runs for the home page finding task. The assessment for the run yehp01 (that is automatic, uses anchor texts, and uses URL substring heuristic) is as follows:

Average reciprocal rank over 145 topics	0.669
Number of topics for which entry page found in top 10	111 (76.6%)
Number of topics for which no entry page was found	32 (22.1%)

The graph showing the reciprocal ranks (RR's) of the home pages for all 145 topics is shown in Fig. 4. Most of the answers are at rank 1 for the queries for which the home pages are included in the ranked list of 100 documents. The result of subtracting median's RR from the RR of our system for each query is plotted in Fig. 5. It can be said that our system belongs to a class of systems which show high performance in home page finding.

Table 3 is given to illustrate the effectiveness of the heuristic based on URL substring relationship. (The mark "url" in the table indicates that the run used the URL heuristic; “Base” is used to indicate a run not using link information; “Link1” for a run using method link1; “Link2” for using method link2.) We can notice that the performance of runs with the URL heuristic is 3 to 4 times better than the corresponding runs without the heuristic when only the document at rank 1 is considered. It can be seen that the use of link information (actually anchor texts in our method) along with the URL heuristic improves the performance when documents of rank 10 or more are included for consideration. However, the data says that using anchor texts only for home page finding did not result in any performance improvement, which does not agree with the suggestion given in [1].

5. Summary

We participated in the web track of TREC-10. We submitted runs for both ad hoc and home page finding tasks. For the ad hoc task we investigated the effectiveness of utilizing the anchor texts. However, we obtained the same result on this issue as the reports in TREC-9 stating that the anchor texts does not enable the systems to achieve significant improvement in retrieval effectiveness. A heuristic called the URL substring relationship was studied in the home page finding task. It is based on the observation that the URL of a home page is a substring of the URL's of web pages in the same site. The use of this heuristic was found to be effective in making the system to be able to move the home page toward the topmost rank.

References

[1] P. Bailey, N. Craswell and D. Hawking, “Engineering a multi-purpose test collection for Web Retrieval experiments,” *Information Processing and Management*, In press.
[2] S. Fujita, "Reflections on "Aboutness" TREC-9 Evaluation Experiments at Justsystem," In Proceedings of the

Ninth Text Retrieval Conference (TREC-9), National Institute for Standards and Technology, 2000.
[3] J.M. Kleinberg, "Authoritative sources in a hyperlinked environment," In Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms, p. 668-677, 1998.
[4] J-M Lim, H-J Oh, S-H Myaeng and M-H Lee, "Improving Efficiency with Document Category Information in Link-based Retrieval," in Proceedings of the Information Retrieval on Asian Languages Conference, 1999.
[5] W. Kraij and T. Westervel, "TNO/UT at TREC-9: How different are Web documents?" In Proceedings of the Ninth Text Retrieval Conference (TREC-9), National Institute for Standards and Technology, 2000.
[6] J. Savoy and Y. Rasolofo, "Report on the TREC-9 Experiment: Link-Based Retrieval and Distributed Collections," In Proceedings of the Ninth Text Retrieval Conference (TREC-9), National Institute for Standards and Technology, 2000.
[7] A. Singhal and M. Kaszkiel, "AT&T at TREC-9," In Proceedings of the Ninth Text Retrieval Conference (TREC-9), National Institute for Standards and Technology, 2000.
[8] D. Hawking, "Overview of the TREC-9 Web Track," In Proceedings of the Ninth Text Retrieval Conference (TREC-9), National Institute for Standards and Technology, 2000.
[9] D. Hawking, E. Voorhees, N. Craswell and P. Bailey, "Overview of the TREC-8 Web Track," In Proceedings of the Ninth Text Retrieval Conference (TREC-8), National Institute for Standards and Technology, 1999.

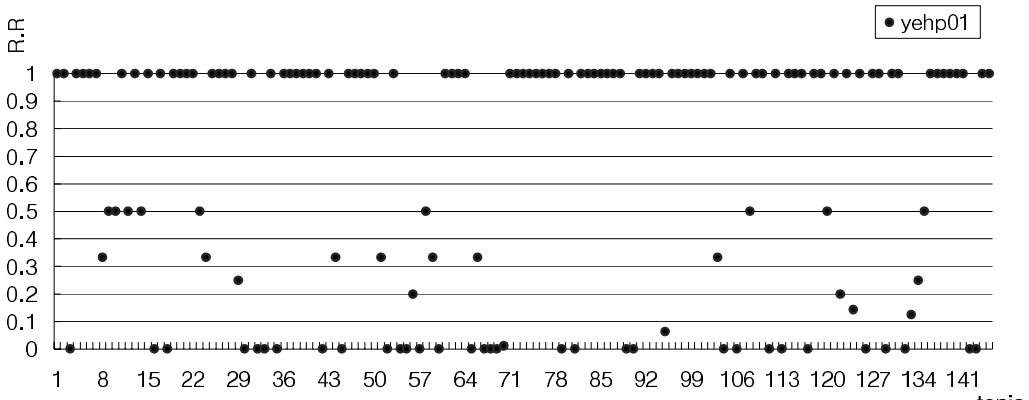

Fig. 4: Reciprocal ranks of topics

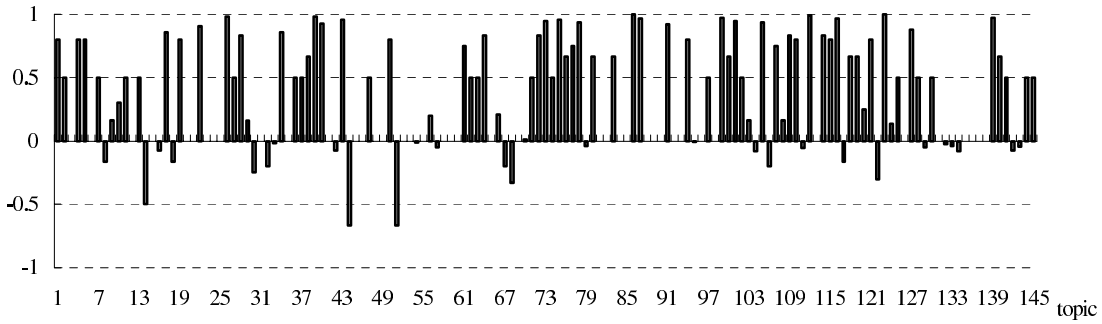

Fig. 5: Difference from median in reciprocal rank

Table 3: Runs using the heuristic of URL substring relationship

Rank	Number of queries with answer within the rank					
	Base	Base/url	Link1	Link1/url	Link2	Link2/url
1	29	80	32	69	30	70
5	65	104	66	106	63	104
10	76	107	76	115	75	115
50	105	112	105	124	103	123
100	111	115	112	125	112	124