Web Document Retrieval Using Sentence-query Similarity

Eui-Kyu Park, Seong-In Moon and Dong-Yul Ra
Computer Science Dept., Yonsei University
{ ekpark, simoon, dyra, }@dragon.yonsei.ac.kr

Myung-Gil Jang
Electronics and Telecommunications Research Institute
mgjang@etri.re.kr

1. Introduction

For the web document retrieval experiments in our TREC '2002 participation, we used two new methods. One is the use of anchor texts, which has been advocated by many researchers. But the methods used by them is different from our method. The second is the use of sentence-query similarity.

It has been known that the use of links for web retrieval did not show impressive improvement in performance [5,6,8,9]. But Bailey, etc. [1] reported that using anchor texts can improve retrieval performance. However, our home page finding experiment done for TREC '2001 showed that it is not the case. The use of anchor texts did not allow any improvement in performance. Our method to use the anchor texts this year is changed a lot from last year and found that it is pretty effective.

The major focus of our experiment this year is in the use of sentential information in information retrieval. We obtain similarity values between sentences of a document and the query and use them for computing the retrieval score of the document. The main idea is the following: a sentence in a document that is much relevant to the query can support relevance of the document to the query. We compute the similarity between each sentence in the document and the query. The degree of this similarity is incorporated in calculating the document's score (in addition to the similarity between the document as a whole and the query). It has been found that it does not take too much time for this extra processing. Our experiment showed that including the sentential information in the proposed way can significantly improve retrieval effectiveness.

2. Use of sentence-query similarity

2.1 Motivation

Let us start by looking at an example. Assume that the query is "the museums in Philadelphia." Let us consider the two documents D_i and D_j as shown below.

D_i : The *museum* of natural history in Chicago is famous. Its huge size surprised a student from *Philadelphia* who was traveling with his family. .....

D_j : John visited a *museum* located in *Philadelphia* after he looked around the University of Pennsylvania campus. The museum contained a lot of things that reveals the nature of American culture. ......

The set of index terms of the query is {museum, Philadelphia}. Taking the document as a whole to match against the query, the relevance of D_i looks almost same as that of D_j since both documents have all of the index terms in the query.

Note that the query terms are distributed throughout the sentences in D_i, but all the query terms appear in a same sentence in D_j (the first one in this case). We argue that having most of the query terms in the same sentence strongly indicates that the document is relevant to the query. This argument works in this example, i.e., D_j is more relevant to the query than D_i.

The ideal way of retrieving documents for a query would be to use the meanings of the sentences in the documents. This indicates that the similarity between a sentence and the query have to play an important role. We try to find out how similar each sentence in the document is to the query. This result must be involved in determining the retrieval score of the document.

The best way to compare a sentence with a query would be to compare their meanings. But the state of the art of natural language processing does not allow this. There does not exist any system yet that can interpret meanings of arbitrary sentences stably. Therefore, it is not possible to build a practical information retrieval system that compares the meanings in computing similarity between a sentence and a query. However, we still want to use sentence-query similarity for information retrieval.

The method that is adopted should be a one that can lead to the practical systems. We decided to use a simple measure for similarity, i.e., the number of common words between the sentence and the query. (This measure is very crude now. But it seems to work and can be replaced by a better one if it is found later.)

2.2. Similarity computation

We adopt the vector-space model to compute the document-query similarity $sim(D,Q)$. Cosine coefficient is used to measure this similarity. Thus the retrieval relevance score of a document D is

$$RSV(D,Q) = sim(D,Q) \qquad (1)$$

To include the sentence-query similarity in the relevance score of D, the next formula is used.

$$RSV(D,Q) = sim(D,Q) + \alpha \sum_{i=1}^{n} C(S_i,Q) \qquad (2)$$

The second term on the right-hand side is the contribution by the sentence-query similarity. (The number of sentences in the document is denoted by n.) $C(S_i,Q)$ denotes the similarity between S_i (the i^{th} sentence in D) and the query Q. Instead of using sophisticated techniques such as natural language processing, computing $C(S,Q)$ is based on the degree of co-occurrence of words between S and Q. It is computed as

$$C(S,Q) = \begin{cases} \left(\dfrac{|S \cap Q|}{|Q|} \right)^k & \text{if } \; |S \cap Q| \geq \tau(|Q|) \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

$|S \cap Q|$ represents the count of the common indexing terms between S and Q. $|Q|$ denotes the number of indexing terms in Q.

The constant α in equation (2) works as a weighting factor for the contribution by the sentence-query similarity. The exponent k in equation (3) is used to control the degree of importance of the high values of the ratio $|S \cap Q| / |Q|$ compared to the lower ones. As k increases, the high ratio becomes more important than the lower ones. $\tau(|Q|)$ is used to nullify the sentential contribution in the cases where the number of common words is small. Currently $\tau(1) = 2$, $\tau(2)=1$, $\tau(3)=2$, $\tau(4)=2$, $\tau(5)=2$, and $\tau(i)=3$ for $i \geq 6$.

3. Using anchor texts

Even though the use of anchor texts did not result in any noticeable performance improvement last year we decided to continue to use anchor texts. But we used it in a different way this year. Let's assume that document D is pointed to by links with anchor texts L_i, $i=1\ldots l$. Let $D_{a(i)}$ denote the document which contains the anchor text L_i.

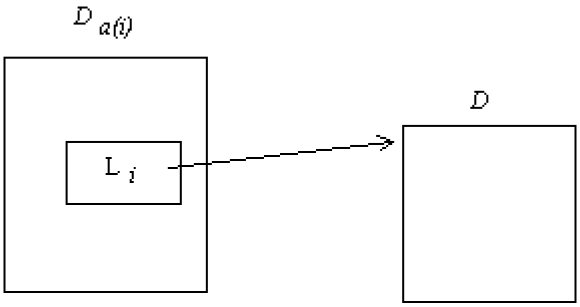

Fig.1: Using anchor texts of incoming links

For a document its incoming links' anchor texts takes part in computing its relevance. In Fig. 1 the anchor text L_i is involved in computing the relevance of D.[1] But outgoing link's anchor text is not used in this processing. Thus the anchor text L_i does not give any contribution to $D_{a(i)}$ as far as links are concerned. We have two methods for utilizing anchor texts.

- Method1 (AT): This one is what was used last year. It uses the cosine coefficient measure to compute the similarity between the anchor text and the query. The contribution by the anchor texts is computed as

$$\sum_{i=1}^{l} sim(L_i, Q) \qquad (4)$$

where *sim* represents the cosine coefficient measure.

[1] An anchor text is considered to be a part of the text of the document containing it. Thus L_i is also used in computing the relevance of $D_{a(i)}$ based upon the vector-space model.

- Method2 (BETA): We additionally use this second method this year. In this method the way we use anchor texts is similar to the use of sentences in the previous section. The incoming link's anchor text L_i in $D_{a(i)}$ is treated like a sentence in *D*. But the weight given to the similarity between an anchor text and the query can be different from that between a sentence and a query. The importance of the anchor text for the relevance of *D* can be different from that of a sentence in *D*. The contribution by the anchor texts L_i's whose links point to document *D* to the relevance of *D* is computed as

$$\beta \sum_i C(L_i, Q) \tag{5}$$

The constant β is used as the weighting factor for the anchor text-query similarity. The same similarity measure *C* is used.

4. The named page finding task

For the named page finding task the relevance score is obtained by incorporating all contributions discussed above.

$$RSV(D,Q) = sim(D,Q) + \alpha \sum_i C(S_i, Q) + \sum_{i=1}^{l} sim(L_i, Q) + \beta \sum_{i=1}^{l} C(L_i, Q) \tag{6}$$

We show the results of experiments related to this task ARR stands for the average(mean) reciprocal rank which is used for indicating performance of systems for this type of tasks.

- Official run :

We submitted one official run on this task whose evaluation is given below. Total number of topics is 150. Column 2 shows the average reciprocal rank(ARR). The third column displays the number of topics for which the answer exists among the top 10 documents of the ranked list. The fourth column is the number of topics for which the answer does not exist in the top 50 of the ranked list. The second row is data from the official run. The third row is about the run with better performance which was obtained at a later experiment with more tuning.

Table 1: Performance in named page finding task

Run	ARR	# topics found in top 10	# topics not found
Official	0.671	124 (82.7%)	13 (8.7%)
Best	0.697	128 (85.3%)	14 (9.3%)

α = 2 β= 10 *k*=3

- Experimentation on the effects of α, β, and k :

Our relevance computation is dependent on the constants α, β, and k. We performed some experiments to find out the best combination of the values of these constants. Fig. 2 shows the result of this experiment ("A" in the figure denotes the parameter α). According to the result, setting α to 1 is recommended. For this best α value, the system performs best when β = 4.

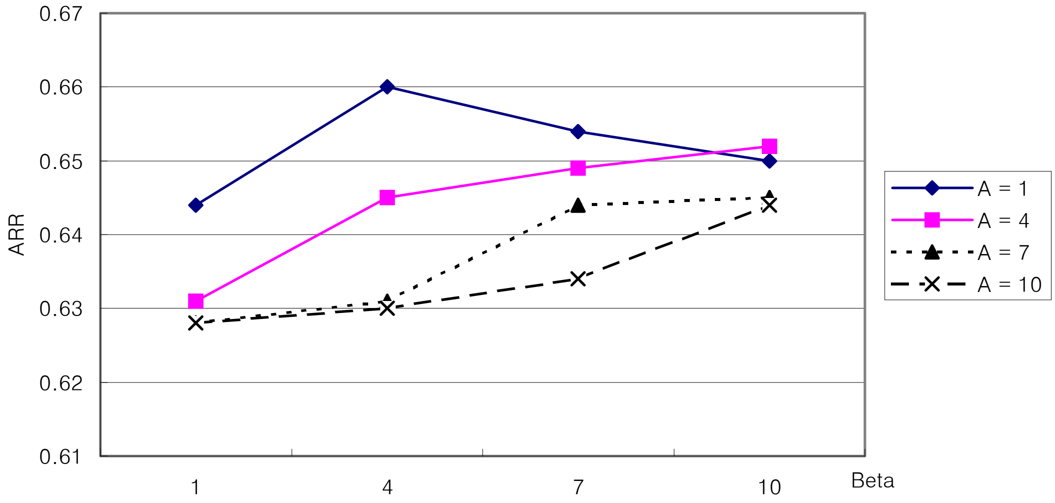

Fig 2: ARR plots for various α, β (with fixed *k* = 3)

- The effect of *k* can be shown in Fig.3 . The best performance is obtained when *k* = 5.

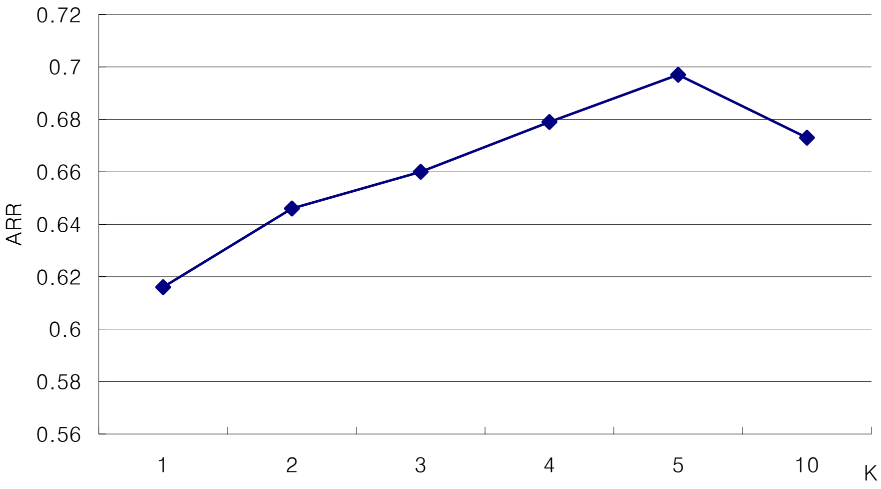

Fig 3: ARR plotted for various k (α=1, β=4)

- Experimentation on the effects of information sources of retrieval

There are several sources that contribute to the relevance of a document. They are document-query similarity obtained by the vector space model (VS), sentence-query similarity (S), similarity between the anchor texts and query by cosine(AT), and anchor text-query similarity obtained by counting common words(BETA). The runs with some of these were generated. CUT in the last row in Table 2 means that the documents are removed from the ranked list when they do not receive positive values from either S or BETA.

Table 2: Various information sources' effects

Runs	ARR	top10	% of top 10	NF	% of NF
np_VS.txt	0.567	118	78.7	16	10.7
np_VS_S.txt	0.641	122	81.3	17	11.3
np_VS_S_AT.txt	0.667	126	84.0	17	11.3
np_VS_S_AT_BETA.txt	0.695	126	84.0	15	10.0
np_VS_S_AT_BETA_CUT.txt	0.697	128	85.3	14	9.3

(α =1, β=4, k=5
NF stands for "not found in top 50".)

This table shows that using sentence-query similarity enables the system to achieve a significant increase in performance. We also observed a noticeable improvement in performance by the use of anchor texts, which were not seen last year.

5. Topic distillation task

In this task we need to find the key resources. They are the documents from which the low and more specific ones can be reached. We want to return a few key resources rather than many low quality or peripheral documents. To make the key resources go up in the ranked list we use the following heuristic:

> For any two documents D_i and D_j in the relevant list (the result of search by a metric such as Eq. 6), increase score of D_i by the amount of score of D_j if D_j is pointed to by a link that exists in D_i.

Therefore, the relevant documents which have many relevant children will get the increased score. In the current implementation only the immediate child can increase its parent's score. The run submitted is shown in Table 3. The result illustrates that our system is not good at the topic distillation task.

One of the reason for the poor performance is that the concept of topic distillation is not clear. It can be either a home page or a specific web page while it can be a sub site. We could not come up with a technique that can identify key resources since our understanding on this concept is obscure.

6. Summary

In computing the retrieval status value of a document sentence-query similarity is incorporated. In addition the anchor text of incoming links is used in a similar way. The requirement for building practical systems made us use a simple scheme in computing this similarity, which is actually the word co-occurrence count. It has been observed that incorporating sentence-query similarity leads to significant increase in performance in the named page finding task. Using anchor texts in a similar way also leads to a better system. For the topic distillation task a simple heuristic has been used but the experimental result showed that it did not worked well.

7. 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.

Table 3: Performance on topic distillation task

Run id: yedi01	Run description: automatic, title only, link(anchor text)	No. of topics: 50
Total number of documents over all topics		
Retrieved: 31072	Relevant: 1574	Relevants retrieved: 640

Recall level precision averages		Document level precision averages	
Recall	Precision	Recall	Precision
0.0	0.3886	At 5 docs	0.1755
0.1	0.2797	At 10 docs	0.1510
0.2	0.1945	At 15 docs	0.1361
0.3	0.1223	At 20 docs	0.1255
0.4	0.1060	At 30 docs	0.1136
0.5	0.0753	At 100 docs	0.0639
0.6	0.0411	At 200 docs	0.0424
0.7	0.0282	At 500 docs	0.0215
0.8	0.0207	At 1000 docs	0.0131
0.9	0.0081		
1.0	0.0045		
Average precision (non-interpolated) : 0.0986		R-precision (exact) : 0.1298	