Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing and Management 41 (2005) 1207–1223
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Techniques for improving web retrieval effectiveness

Eui-Kyu Park [a], Dong-Yul Ra [a,*], Myung-Gil Jang [b]

[a] *Computer Science Department, Yonsei University, Wonju, Kangwon 220-710, Korea*
[b] *Speech/Language Information Research Department, ETRI, Yuseong-gu, Daejeon 305-350, Korea*

Received 30 October 2003; accepted 17 August 2004
Available online 7 October 2004

Abstract

This paper talks about several schemes for improving retrieval effectiveness that can be used in the named page finding tasks of web information retrieval (Overview of the TREC-2002 web track. In: Proceedings of the Eleventh Text Retrieval Conference TREC-2002, NIST Special Publication #500-251, 2003). These methods were applied on top of the basic information retrieval model as additional mechanisms to upgrade the system. Use of the title of web pages was found to be effective. It was confirmed that anchor texts of incoming links was beneficial as suggested in other works. Sentence–query similarity is a new type of information proposed by us and was identified to be the best information to take advantage of. Stratifying and re-ranking the retrieval list based on the maximum count of index terms in common between a sentence and a query resulted in significant improvement of performance. To demonstrate these facts a large-scale web information retrieval system was developed and used for experimentation.

Keywords: Web information retrieval; Named page finding; Retrieval effectiveness; Sentence–query similarity

1. Introduction

A huge amount of information exists in the Internet web. The number of web pages is enormous. But they are not of much use if users cannot easily find the information they want. In this respect web retrieval has become an important technology. It will be more indispensable as the amount of information in the web gets larger.

To foster research on information retrieval (IR) technology the Text Retrieval Conference (TREC) has been held annually (Harman, 1997). The web track in TREC is specialized for web IR. This track's major

* Corresponding author. Tel.: +82 33 760 2246; fax: +82 33 763 4323.
E-mail address: dyra@dragon.yonsei.ac.kr (D.-Y. Ra).

0306-4573/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2004.08.002

Table 1
Sample topics and relevance judgments

Q. No.	Topic (query)	Relevance judgment (Doc. ID's)
1	America's Century Farms	G00-04-3805407, G00-65-2264297
2	US agriculture changes 20th century	G40-56-0342561
3	Volunteer FEMA World Trade Center	G12-46-0465715
4	Prevent baseball injuries	G09-04-2395783
5	Long term health insurance Maryland tax	G26-05-0000000
6	e-coli beef inspection results	G05-69-2505304
7	ban of exports to Iraq	G19-68-0198027
8	Senator Carnahan biography	G00-06-2764514, G00-90-0514219

goal was to develop technology for the ad hoc retrieval task. In this respect the goal has not been much different from that of the conventional IR systems. As web retrieval gets more commonly and widely used several retrieval modes have been proposed and studied in the web track. The home page finding task was introduced after several years of research on the topic relevance (ad hoc) task. The objective of this task is to find the home (site entry) pages of the web sites that contain pages relevant to the query (Hawking & Craswell, 2002). Then the topic distillation and the named page finding task were proposed. The topic distillation task's objective is to find the key resource pages which are the gates leading to the web pages or documents relevant to the query. The named page finding task is to find the exact web pages or documents that actually contain the relevant information (Craswell & Hawking, 2003).

Research in this paper is to develop techniques that are useful for *the named page finding task*. This task is considered to be one of the types of people's web search activities. The baseline system of ours is the search engine with the vector-space model that is popular in the conventional IR systems. On top of this basic model, several schemes are applied that can improve performance [1] of the overall system. These include:

- using the title sections of web pages,
- using the anchor texts of the incoming links,
- using sentence–query similarity,
- stratifying and re-ranking the retrieval list,
- cutting off the documents that have limited kinds of information.

To justify the validity of these techniques, we developed a web IR system, applied these techniques to the baseline system, and measured the system performance. Effectiveness of each technique was measured quantitatively. The system is not for the commercial use but must be a large-scale system to be able to handle the document collection (called .GOV) of 18 gigabytes consisting of 1.25 million documents (Craswell & Hawking, 2003; CSIRO, 2003). Deliberate consideration had to be exercised to make the system efficient in time and space for indexing and retrieval.

The test collection has 150 topics provided by NIST. The relevance judgments for the topics were prepared by NIST. Table 1 shows some of the topics and the corresponding relevance judgments. Before the announcement of the relevance judgments the participants to the TREC sent the result of runs of their system (called the official runs) to NIST and received the performance evaluation of those runs. The performance data of the official runs in the web track of TREC-2002 can be found in (Craswell & Hawking, 2003). This data is compared with performance measurements of our unofficial runs to assess the techniques pro-

[1] Henceforth, performance refers to retrieval effectiveness of the system in this paper.

posed in this paper. Our experiment demonstrated that our techniques are useful for improving retrieval effectiveness.

This paper is organized as follows. Section 2 describes the related works. In Section 3 the basic model that forms the underlying foundation is explained. Section 4 provides the descriptions of the techniques proposed for improving retrieval effectiveness on the web. Experimental results and evaluations are given in Section 5. Finally Section 6 has the concluding remarks.

2. Related works

Web documents have hyperlinks connecting web pages. This is the feature which conventional information retrieval systems could not take advantage of. [2] Naturally research works on web information retrieval in the early years focused on developing techniques to make use of information obtainable from hyperlinks. Kleinberg (1999) proposed a link-based algorithm which calculates and uses the authority and hub scores of documents. These scores are totally based upon the link connectivity of pages rather than semantic contents. A PageRank algorithm was suggested by Brin and Page (1998) which assigns a page rank (PR) value to a page based upon the PR values of the pages pointing to it. PR values depend on the connection topology as Kleinberg's. Another scheme called spreading activation propagates the page–query similarity through links to other documents. Thus the final retrieval score of a document depends on both link connectivity and contents (Crestani & Lee, 2000). Some approaches used the in-link counts in various ways to reflect the popularity of a document on computing the relevance score (Gurrin & Smeaton, 2001). However, most of the experiments to apply these techniques for exploiting link connectivity did not result in significant performance improvement (Hawking, 2001; Hawking, Voorhees, Craswell, & Bailey, 2000; Savoy & Picard, 2001).

The early attempts to exploit anchor texts of incoming links can be seen in Brin and Page (1998) and McBryan (1994). More recently Fujita (2001) and Singhal and Kaszkiel (2001) also studied on the use of this data representative. However, they could not observe any reliable improvement in retrieval effectiveness. On the contrary, more recent works have reported that in-link anchor text is useful (Craswell, Hawking, & Robertson, 2001; Kraaij, Westerveld, & Hiemstra, 2002). It is now generally accepted that in-link anchor text is helpful in web IR. However, there is no unified agreement on the scheme for making use of it.

Another characteristic of web documents is the document structure. They have the title section and several levels of H sections (where H stand for headlines). From early web IR research, systems attempted to utilize this structure. But their importance was not clear because using the structure did not result in enhancement in retrieval effectiveness in the topic relevance task (Amati & Carpineto, 2002; Savoy & Picard, 2001). However, the systems in recent days tend to make use of the title as a major document representative in the named page finding task (Craswell & Hawking, 2003). The experimental results support this approach.

Another source of information for web retrieval is URL. It was shown that this information could be valuable in the home page finding task (Fujita, 2002; Westerveld, Kraaij, & Hiemstra, 2002). However, it was not confirmed that it is helpful in the named page finding task (Craswell & Hawking, 2003).

Passage retrieval (Callan, 1994; Kaszkiel & Zobel, 1997) has a close relationship with our technique of using sentence–query similarity. A lot of research has been done on passage retrieval and revealed that it can improve the retrieval effectiveness significantly. However, the use of sentences in our sentence–query similarity technique has some aspects different from the passage retrieval. A passage is not confined to a

[2] Academic papers do have citation links. But retrieval systems have not utilized them.

sentence. It is usually a unit larger than a sentence like a paragraph. Many passage retrieval works suggest even using as passages fixed-sized blocks that can start and end anywhere within a sentence. In passage retrieval the system produces as a retrieval result a ranked list of passages and then uses the ranks to adjust the scores of documents containing the passages. In contrast with this method no attempt is made to make use of ranks of sentences in our approach. We also suggest a novel method to assess the similarity between a query and a sentence.

There were attempts to use an interval of text in which all query terms occur for computing the document's score (Clarke, Cormack, & Burkowski, 1995; Hawking & Thistlewaite, 1996). In their approach all such intervals or spans are identified. The shorter the interval the better it is. The documents with good intervals are likely to be relevant to the query. To realize this idea the sum of the inverses of the lengths of all such intervals are computed and used as the major factor in computing the document score. However, an interval can lie across different sentences. There is no direct relationship between an interval and a sentence.

3. The baseline system

We have developed and implemented the whole system including the core search engine. [3] We describe the underlying model of our baseline system. Our system is based upon the vector-space model [4] (Salton, 1989; Salton, Wong, & Tang, 1975). Table 2 contains notations used in the paper. In this model all index terms are arranged in a linear list and each term is given a unique number corresponding to the position in the list. Let t be the total number of index terms. Then a t-dimensional vector is used to represent a document. The representation for document d_j is

$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

The element $w_{i,j}$ corresponds to index term k_i and can be interpreted as the weight of k_i in representing document d_j. It is computed as follows.

$$f_{i,j} = \frac{tf_{i,j}}{tf_{\max,j}} \quad \text{where } tf_{\max,j} = \mathrm{MAX}_i tf_{i,j}$$

$$w_{i,j} = f_{i,j} \times \log \frac{N}{df_i}$$

Similarly, query q is represented by a vector

$$q = (q_1, q_2, \ldots, q_t)$$

The weight q_i corresponding to index term k_i is computed as

$$q_i = \left(0.5 + \frac{0.5tf_{i,q}}{tf_{\max,q}}\right) \times \log \frac{N}{df_i}$$

We use the cosine coefficient (denoted as sim_0) of two vectors as the criterion for retrieval in the basic model.

[3] We could not use the publicly available IR engines because sentence–query similarity mechanism our system use necessitates the major redevelopment of the system.

[4] Other models such as probabilistic models can be used to build our baseline system.

Table 2
Summary of notations

t: the total number of index terms	df_i: the number of documents in which k_i occur
N: the total number of documents	$tf_{i,j}$: the frequency of k_i in d_j
Ω: the collection of documents	$tf_{i,q}$: the frequency of k_i in query q
d_j: the jth document in Ω	$f_{i,j}$: the normalized frequency of k_i in d_j
k_i: the index term numbered i	L: an anchor text

$$\mathrm{sim}_0(d_j, q) = \frac{d_j \bullet q}{|d_j| \times |q|} \tag{1}$$

The numerator is the inner product of the two vectors and $|d_j|$ denotes the length of d. We call the overall criterion for retrieval the retrieval status value, RSV. In the basic model alone just sim_0 constitutes the RSV.

$$\mathrm{RSV}(d, q) = \mathrm{sim}_0(d, q) \tag{2}$$

When a query q is given to the system, the retrieval process first computes the RSV of all documents. The result is a list of documents whose RSV is greater than some threshold (usually 0). The list is ranked according to the RSV in descending order. The rank of a document is the position in the list (starting from 1). As a result of retrieval the system returns the top n documents in the ranked list for some appropriate value n.

4. Techniques for improving retrieval effectiveness

In this section we introduce several techniques adopted by our system to enhance its retrieval effectiveness. Some of them are based on the ideas introduced or suggested previously in other works. We describe the approaches we take to make use of these ideas and will show in a later section how good the results are. Furthermore this section includes some techniques that we propose for performance improvement.

4.1. Use of the title section

Most of web documents have the title section. Savoy and Picard (2001) said that according more importance to keywords appearing in the title or H1 logical sections [5] did not have a significant effect on performance in the topic relevance task of web search. In contrast with their result, we found that utilizing the title is valuable. We think that the index terms in the title section are more indicative about the subject of the document than those in the text body. To apply this idea we use the following scheme: While each occurrence of an index term in the text body makes the term frequency (*tf*) be incremented by 1, an occurrence of the term in the title results in the increment of *tf* by h where h is greater than 1. [6] Thus index terms occurring in the title are given more importance than those in other sections of the document.

4.2. Using sentence–query similarity

The vector-space model is actually a methodology to measure and use similarity between a document and a query. Vectors are used for representation and comparison. However, in the case of the named page finding task, similarity between a query and a sentence seems to be useful because the query describes the

[5] H1 sections stands for sections in HTML files enclosed by ⟨H1⟩ and ⟨H1⟩ tags. Headline-like texts are stored in them.
[6] In our current system, $h = 5$.

Table 3
An example for sentence–query similarity

d_i	It is important to watch the collections in this field of archeology. You'd better to go to a museum.....
d_j	The Field Museum is located on Chicago's Museum Campus, at 1400 S. Lake Shore Drive. It is just south of Roosevelt Rd. ….
q	Field Museum

page by name (Craswell & Hawking, 2003). A name usually appears within a sentence rather than being spread across adjacent sentences. We hypothesize that the relevance of a document can be increased if it has a sentence that contains the whole or large part of the name given in the query. [7] This gives us the motivation for the approach of using sentence–query similarity. We want to take into account the similarity between a sentence and a query for the named page finding task. This information seems to be important.

Let us take an example in Table 3. Note that the query here uses a name to specify the pages to be retrieved. Suppose that the document collection contains the two documents d_i and d_j in the table. Their similarity to the query is almost equal. That is, $\mathrm{sim}_0(d_i, q) \approx \mathrm{sim}_0(d_j, q)$. [8] This is because both documents have the index terms "Field" and "Museum" that the query q has. However, humans can easily decide that d_j is more relevant than d_i to q. This decision is due to the first sentence of d_j. The name specified in q appears in this sentence. The second sentence of d_j provides no contribution. Similarity between the first sentence of d_j and q is substantial and affects decision-making. This example provides a motivation to take into account sentence–query similarity in the named page finding task.

Sentence–query similarity can be most accurately computed when a system can understand meanings of sentences. This is possible if natural language understanding (NLU) technology is fully developed. Ideal IR systems will employ NLU technology to compare the meaning of a sentence with that of a query. Unfortunately, current NLU research is not mature enough to allow ideal IR systems. There have been a lot of research efforts on phrasal or semantic indexing to take advantage of NLU techniques. Unfortunately they were not successful enough (Perez-Carballo & Strzalkowski, 2000).

As illustrated in Table 1, queries in the named page finding task have a form of a list of several words (especially nouns) like information requests in web search being done by people these days. Here we make an assumption that "relevance to a query can be detected by considering sentences separately rather than all sentences together". There might be exceptions to this assumption but it seems to be correct in many cases. This idea is motivated by the consideration that, in a relevant document, query terms appear in concentrated fashion in a sentence rather than distributed among multiple sentences.

To apply this idea we need to compute the degree of closeness between a sentence and a query. This value must be one of the factors determining the RSV. The best way to compute this degree is to compare the meanings of them. As mentioned above it is not practical at present. To circumvent this problem, we propose to use $C(s, q)$ introduced below to approximate the degree of closeness between sentence s and query q.

$$C(s,q) = \begin{cases} \left(\dfrac{|s \cap q|}{|q|}\right)^k & \text{if } |s \cap q| \geqslant \tau(|q|) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here $|s \cap q|$ is the number of index terms in common between s and q. We denote the number of index terms in q by $|q|$. The exponent k is a parameter that is used for controlling importance of the number of terms in common. As k increases, big $|s \cap q|$ becomes more important. A threshold function τ is used to ignore sen-

[7] "Sentence" is taken to be the unit of string delimited by the punctuation marks or by html tags (as in anchor texts) in our approach.

[8] We assume that the vectors have a similar length.

tence–query similarity if $|s \cap q|$ is too small compared with $|q|$. The current values of τ are set as follows: $\tau(1) = 2$, $\tau(2) = 1$, $\tau(3) = 2$, $\tau(4) = 2$, $\tau(5) = 2$, and $\tau(i) = 3$ for $i \geqslant 6$.

For all sentences s_i, $1 \leqslant i \leqslant l$, in document d having l sentences the C values are summed to get the sentence–query similarity $\mathrm{sim}_1(d, q)$ between d and q:

$$\mathrm{sim}_1(d, q) = \sum_{i=1}^{l} C(s_i, q) \tag{4}$$

where s_i's are sentences in d. The RSV for a system that employs sentence–query similarity on top of the basic model is computed as

$$\mathrm{RSV}(d, q) = \mathrm{sim}_0(d, q) + \alpha \cdot \mathrm{sim}_1(d, q) \tag{5}$$

The coefficient α is used to control the weight of sentence–query similarity in the overall RSV value.

The first sentence of d_j in Table 3, say s_j^1, has all index terms that the query q has. Let $k = 5$. The value of $C(s_j^1, q)$ is 1 while $C(s_i^1, q)$ for s_i^1 (the first sentence in d_i) has the value $(1/2)^5 \approx 0$. We find that $\mathrm{sim}_1(d_j, q)$ is bigger than $\mathrm{sim}_1(d_i, q)$. Therefore, d_j is decided to be more relevant to the query than d_i (assuming d_i and d_j have the same sim_0 values). The main reason for this result is that all index terms in q exist in d_i but they occur in different sentences. The observation so far related to this example leads to a heuristic.

• **Heuristic "S–Q similarity"**: The less the index terms of a query are distributed over different sentences in a document, the bigger the relevance score of the document gets.

To be able to compute sentence–query similarity we need to store in the index storage additional information, i.e., the sentence numbers of sentences (of a document) in which an index term occurs. We store this information in *the sentence number list*. Each document node in the index storage structure should have this list as shown in Fig. 1. This figure shows a node for document d_j in the document occurrence list (DOL) for index term x. The sentence number list linked to this node has the numbers identifying the sentences in d_j in which the term x occurs. The sentence number list is implemented as an array. So a sentence number list can be stored or read by one file transfer operation during the indexing or retrieval processing. Therefore, the system does not experience serious slowdown of speed.

4.3. Exploiting link information

Our system does not exploit link connectivity since web IR research so far could not show its effectiveness. Instead we use anchor texts of incoming links. It is known that they are helpful and regarded as one of the important document representatives.

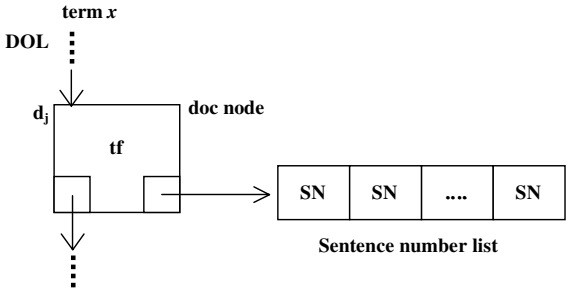

Fig. 1. Storing sentence numbers in index storage.

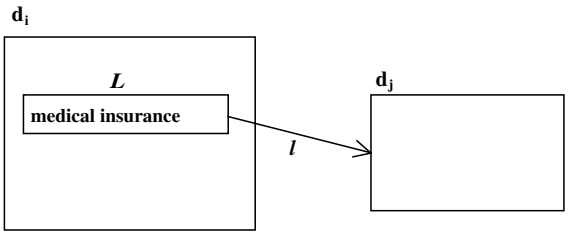

Fig. 2. A link and its anchor text.

4.3.1. Using anchor texts of incoming links

Links are one of the most special characteristics of web pages. A lot of attention has been paid on the use of links since web IR research started. Most of the approaches tried to use the connectivity formed by links (Kleinberg, 1999) as well as the retrieval scores of the linked pages. Surprisingly it was turned out that these approaches did not result in nontrivial improvement in performance (Hawking, 2001). However, several independent research efforts later revealed that anchor texts of incoming links are valuable in retrieval effectiveness enhancement (Craswell et al., 2001; Kraaij et al., 2002).

To calculate the RSV of a document d, we make use of the anchor texts of the links coming into d (the in-link anchor texts of d). For example, the link l in Fig. 2 is an in-link of d_j and out-link of d_i. It is anchored in the text "medical insurance". This text is called the anchor text of link l. The anchor text and the link say that one can look at d_j to find some information about the topic "medical insurance". The anchor text of a link seems to indicate the content of the document being pointed to. In particular an anchor text can be involved in the relevance computation of the destination document of the corresponding link. We take two approaches in using anchor texts.

- Method 1: Similarity between an anchor text and a query is computed using the cosine coefficient. Let document d have m in-links whose anchor texts are L_i, $1 \leqslant i \leqslant m$. Interpreting L_i as the vector representation,

$$\mathrm{sim}_{2a}(d,q) = \sum_{i=1}^{m} \frac{L_i \bullet q}{|L_i| \times |q|} \tag{6}$$

  Contribution by the anchor texts to relevance of d is represented by $\mathrm{sim}_{2a}(d,\ q)$.
- Method 2: The degree of closeness between anchor texts and a query are used in this method as in sentence–query similarity computation explained above. We use the function C introduced in Section 4.2.

$$\mathrm{sim}_{2b}(d,q) = \sum_{i=1}^{m} C(L_i,q) \tag{7}$$

The total contribution by using link information is obtained by adding the values from the two methods.

$$\mathrm{sim}_{2}(d,q) = \mathrm{sim}_{2a}(d,q) + \mathrm{sim}_{2b}(d,q) \tag{8}$$

Incorporating link information to relevance computation leads to the next formula for the RSV. The coefficient β in this formula is used to control the weight of link information.

$$\mathrm{RSV}(d,q) = \mathrm{sim}_{0}(d,q) + \alpha \cdot \mathrm{sim}_{1}(d,q) + \beta \cdot \mathrm{sim}_{2}(d,q) \tag{9}$$

4.3.2. An efficient procedure for using anchor texts

The number of links is huge in the web test collection (Bailey, Craswell, & Hawking, 2003). Therefore, an attempt to use link information requires efficiency in computation. Otherwise, the system can experience

much slow-down of speed. As explained before, the system considers only documents whose sim_0 is greater than 0 to save time. The documents that do not have any query terms are not even considered during the processing. Similarly time constraint does not allow the system to consider all documents in computing sim_2. We need to develop a scheme that those documents whose sim_2 is 0 should never be enumerated or considered.

We take a similar approach to that shown in (Lim, Oh, Maeng, & Lee, 1999). Given a query, all documents whose sim_0 is greater than 0 are identified. They constitute the set A:

$$A = \{d \mid \text{sim}_0(d,q) > 0 \text{ for all } d \text{ in } \Omega\} \tag{10}$$

Computing A in (10) involves consideration of texts of documents but not the link structures. In contrast a document can have relevance to the query via the anchor texts. In other words by having nonzero sim_2 a document can have the RSV greater than zero.

We emphasize that, in our system, anchor texts of outgoing links of a document are *part* of the document (text) body. Let us consider the case in Fig. 2. Assume that the query is "health insurance". The anchor text "medical insurance" is considered to be a part of the text body of d_i in our system. The occurrences of the terms in this anchor text contribute to the *tf*'s of those terms related to d_i. The term "insurance" in the anchor text makes some contribution to sim_0 of d_i since it makes the corresponding *tf* nonzero. Thus sim_0 of d_i is greater than 0. (In contrast, the terms in the anchor text do not contribute to sim_0 of d_j.) Notice here that if sim_2 of d_j is nonzero because of this anchor text then it is certain that sim_0 of d_i is nonzero which holds the anchor text. A little thought based on this observation leads to the fact that a document having zero as sim_0 does not have any anchor text which can contribute to sim_2 of other documents. Rephrasing this is to say that every document holding an anchor text that contributes to sim_2 of other documents belongs to the set A.

$$B = \{d \mid \text{sim}_2(d,q) > 0 \text{ for all } d \text{ in } \Omega\} \tag{11}$$

$$E = \{d \mid \text{sim}_0(d,q) + \text{sim}_2(d,q) > 0 \text{ for all } d \text{ in } \Omega\} \tag{12}$$

Here it holds that

$$E = A \cup B$$

E is the set of all documents that the system considers to be relevant to the query. As a side effect we can say that $F = A - B$ is the set of documents which receive no contribution from anchor texts but only from the text bodies and $D = B - A$ has the documents that have nonzero score contributed to only by anchor texts but not by the text bodies.

We should be able to compute E in an efficient way. For this we use the basic idea (drawn above) that any document's nonzero sim_2 is because of anchor texts in documents that belong to A. We check each document in A whether it has outgoing links whose anchor texts have some relevance to the query. If such case is found the destination document is given the appropriate sim_2 value and is added to E if it is not already there. The procedure for obtaining E is as follows:

(1) Find and insert into A all documents whose sim_0 is bigger than 0 using the basic model.
(2) Copy A into E.
(3) For each document d in A, check every out-link anchor text L in d. If L share some index terms with query q, do the next two steps (Let d' be the document pointed to by the link corresponding to L):
    (i) Add to sim_2 of d' the amount contributed to by L.
    (ii) Put d' into E if E does not have d'.
(4) E is the final result and can be used for further processing.

The number of documents in A is much smaller than the whole document collection. Thus this procedure is efficient because only the documents in A are considered in step (3) above. To implement this procedure, preprocessing (for each document) needs to record those documents pointed to by outgoing links in it and their anchor texts.

4.4. A stratifying and re-ranking stage

The RSV of a document is determined by using the techniques introduced so far. Then the documents whose RSV is bigger than 0 are sorted in descending order and ranked. The result is the RSV-based ranked list and is usually used as the final output.

However, before producing the final output we let the RSV-based ranked list undergo another stage, the stratifying and re-ranking stage. This additional stage is based on the maximum count ϕ of index terms in common between the sentences and a query. For each document in the ranked list, ϕ is computed as follows:

$$\phi(d,q) = \mathrm{MAX}_i \mid s_i \cap q \mid \tag{13}$$

where s_i denotes a sentence in d. As before $|s_i \cap q|$ represents the cardinality of the set of terms that occur in both s_i and q.

The main idea of having this stage is that if the ϕ of d_i is bigger than that of d_j then we want to make d_i more relevant to q than d_j regardless of their RSV values. In other words the ϕ has a higher precedence than the RSV. The documents with the same ϕ belong to the same layer and they are ranked according to the RSV score. It means that the documents are stratified according to ϕ. Let rank(d) be a positional index for d in the final list to be output as the retrieval result. The smaller rank value means the bigger relevance. What is done in this stage can be defined precisely as follows:

$\mathrm{rank}(d_i) \leqslant \mathrm{rank}(d_j)$ if and only if

$$\phi(d_i,q) \geqslant \phi(d_j,q) \quad \text{or} \quad [\phi(d_i,q) = \phi(d_j,q) \text{ and } \mathrm{RSV}(d_i,q) \geqslant \mathrm{RSV}(d_j,q)] \tag{14}$$

4.5. Cutting off documents based upon information sources

Finding the large number of documents including ones with small relevancy is not what the named page finding task wants to do. It wants to retrieve a small number of documents that are exactly the ones wanted by the query. We assume that the query expression is very specific. Therefore, recall is not considered to be important. The system must do its best to find a small number of documents that the query really wants.

This characteristic of the task led us to cut off the documents that do not receive contribution from either sentence–query similarity or anchor texts. The experiment showed that this scheme improves the performance.

5. Experimental results and evaluation

Experiments were done for the named page finding task using the test collection for the web track of TREC-2002. In the named page finding task one or two documents are given as the relevance judgment to each query. Because of the characteristics of the task the mean reciprocal rank (MRR) is used as the criterion for performance evaluation instead of recall and precision. The MRR is computed as follows. The answers in the relevance judgment are located in the final list returned by the system. If there are more

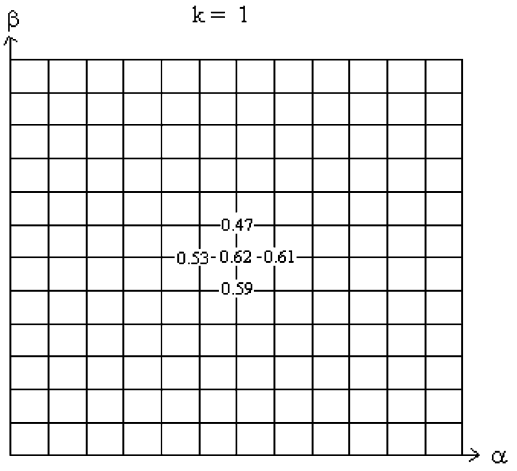

Fig. 3. A grid for determining α and β for a fixed k.

than one answers found in the list, the best one is taken. Then the reciprocal of this document's rank is recorded. [9] After the reciprocals for all queries are computed, their mean is computed as the MRR. The ideal system's MRR would be 1 since the first document in the output will be an answer for every query. The worst system will have 0 as the MRR since no answer is found in the final output list for any query.

5.1. Determining the parameter values

The system has three parameters α, β and k. [10] We need to determine their values. This process can be referred to as the training. We partition the whole query set with 150 queries into the training and test sets. In our experiment, the 30 queries are assigned to the testing set and the rest to the training set. The parameters are determined to be the values that result in the best performance for the training set. The MRR measure explained above is used as the criterion for performance comparison. The performance of the system to report is measured upon the queries in the test set by using the parameter values obtained from the corresponding training set.

Given the training set, the next procedure is used to determine the (optimal) parameter values. We hold k as a constant and vary α and β. This results in a grid of α and β values as shown in Fig. 3. For each grid point (with the particular combination of α and β) the MRR is measured using the queries in the training set. This is illustrated in Fig. 3 where the measured MRR value is shown at some of the grid points. The values of α and β range from 0 to 10 taking a value at every interval of 0.2. For a grid, the grid point with maximum MRR is identified. The α and β values at this point (along with the maximum MRR) are recorded in association with the grid.

This process of creating a grid and finding the best α and β values is repeated for every k between 0 and 10 at every interval of 0.5. Fig. 4 shows the plot of the maximum MRR for various k. The best k is selected. The α and β recorded for the grid of this best k determines their optimal values. Note that the optimal parameter values for α, β, and k depends upon the given training set. Changing the training set will lead to a different optimal parameter values.

[9] If no answers are found in the output for the query the reciprocal is set to 0.
[10] See Eq. (9) for α and β and Eq. (3) for k.

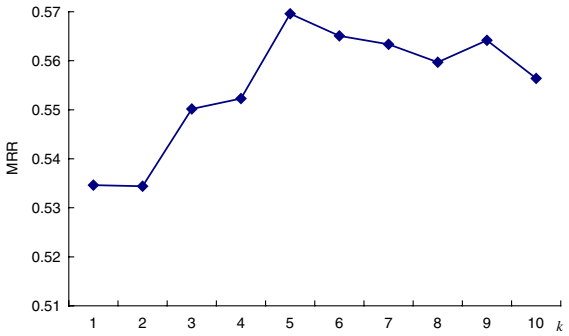

Fig. 4. The plot used for selecting best k.

5.2. Performance data

5.2.1. Multiple sets of training and test data

Recent machine learning technology recommends us to have the training and test data be separated, use multiple sets of training and test data and use a statistical test to assess the meaningfulness of the performance result. We prepared 10 sets of training and test data out of the test collection with 150 queries. The first test and training data set is determined by taking the first 30 queries as the test data and the rest as the training data. The test data for the second set starts from the middle of the first test data and takes the next 30 queries (and the rest as the training data). A half of the second test data overlaps with the half of the first test data. In this way we prepared 10 sets of test and training data.

For each test and training data set we measured the performance of the system. Once training data is given, the optimal parameter values are determined by the procedure explained in Section 5.1. By using these parameter values the performance of the system is measured using *the corresponding test data*. Therefore, we come up with 10 sets of performance measurements.

5.2.2. Performance measurement results

Table 4 shows the performance measurement result. Each number in this table is actually the *average* of 10 measurements for our multiple (training and test) data sets. Each row represents a combination of the proposed schemes. Column "Top 10" has the number of queries (and its ratio) for which an answer was found among the top 10 documents in the retrieval result list. "Not found" has the count of the queries (among the total 30 test queries) for which an answer document was never found in the retrieval result list.

Table 4
Performance data for various combinations of methods

No.	Combinations of schemes	MRR	Top 10		Not found	
			Count	%	Count	%
1	sim_0	0.385	17.2	57.3	7.8	26.0
2	sim_0 + title	0.415	19.0	63.3	6.6	22.0
3	sim_0 + title + sim_1	0.567	23.4	78.0	4.4	14.7
4	sim_0 + title + sim_2	0.445	19.7	65.7	4.6	15.3
5	sim_0 + title + sim_1 + sim_{2a}	0.592	23.7	79.0	3.4	11.4
6	sim_0 + title + sim_1 + sim_{2b}	0.608	24.3	81.0	3.3	11.0
7	sim_0 + title + sim_1 + sim_2	0.623	24.4	81.3	2.8	9.3
8	sim_0 + title + sim_1 + sim_2 + cut	0.630	24.4	81.3	2.8	9.3
9	sim_0 + title + sim_1 + sim_2 + cut + str	0.698	25.6	85.3	2.6	8.7

Row 1 (sim_0) shows performance of the system using the basic model alone. This is the base line performance of the system. Adding the title information improves the system a lot as shown in row 2 (sim_0 + *title*). This shows that exploiting the title section is important in web information retrieval. The remaining rows represent the various combinations that incorporate some of the proposed techniques.

Exploiting sentence–query similarity as stated in Eq. (5) results in the most significant performance improvement as demonstrated in sim_0 + title + sim_1. This is a novel technique that we have suggested for the named page finding task of web information retrieval.

Utilizing hyperlinks results in performance shown in rows 4–7. The symbol sim_2 stands for the use of both Method 1 and Method 2 for exploiting hyperlinks explained in Section 4.3. Rows 5 and 6 show that both Method 1 (sim_{2a}) and Method 2 (sim_{2b}) contribute to system enhancement. The data in the table says that Method 2 was found to be better than Method 1. Using both of them (sim_0 + title + sim_1 + sim_2) is better than using just one of them (sim_0 + title + sim_1 + sim_{2a} or sim_0 + title + sim_1 + sim_{2b}). We observe in the table that using both methods for anchor texts (sim_0 + title + sim_1 + sim_2) is significantly better than not using in-link anchor texts (sim_0 + title + sim_1). This is a clear evidence for usefulness of using in-link anchor texts.

The documents are removed from the retrieval result when they do not receive any contribution from either sim_1 or sim_2. This technique of "cutting off" explained in Section 4.5 seems to achieve a slight improvement in performance as observed in sim_0 + title + sim_1 + sim_2 + cut.

Incorporating the stratifying and re-ranking stage was proven to be quite effective as shown in sim_0 + title + sim_1 + sim_2 + cut + str. The data indicates that stratifying and re-ranking is more effective than using in-link anchor texts. This is an extraordinary result. We need to pay more attention to this technique in the future.

5.2.3. Statistical tests for meaningful comparison

The MRR given in Table 4 is the average of the performance measurements for our 10 (training and test) data sets. To have a better grasp on the significance of any difference between schemes it is necessary to perform the statistical tests. Otherwise, it is not clear whether the difference is just due to natural statistical variations or not. For this purpose we adopted the t test that is widely used for performance comparison between a pair of systems (Manning & Schütze, 1999).

The performance of a system is represented by a sample having n values ($n = 10$ in our case). Let us assume that $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ are the two samples where the elements are the MRR values measured. The t score is computed as follows.

$$t = (\bar{y} - \bar{x})/\sqrt{2s^2/n} \tag{15}$$

where n is the sample size (the number of measurements), $\bar{x}$ and $\bar{y}$ are the averages, and $s^2 = [\sum_{i=1}^{n}(x_i - \bar{x})^2 + \sum_{i=1}^{n}(y_i - \bar{y})^2]/2n$ is the pooled estimate of the variance of the two samples. The result of t score calculation is shown in Table 5. The columns C_1 and C_2 denote the pair of systems (combinations of schemes) under comparison. The next column has the t score. The last column gives the confidence level corresponding to the critical value which is equal to the given t score. (In our case it is a one-tailed test with 18 degrees of freedom.)

They allow us to figure out the significance of performance difference between C_1 and C_2. For example, $t = 7.33$ at row 2 indicates that C_2, sim_0 + title + sim_1 is superior to C_1, sim_0 + title, with confidence level of more than 99.9%. Thus it is quite certain (99.9%) that the sentence–query similarity scheme (sim_1) added to sim_0 + title leads to a better system. In row 7 the mediocre confidence of 65% says that one is not sure that the cutting off technique achieves performance improvement. The techniques sim_1 (sentence–query similarity), str (stratification and re-ranking), sim_2 (anchor texts), and sim_{2b} (Method 2 for anchor texts) were found to be effective for system enhancement (in decreasing order of confidence).

Table 5
Statistical test results for assessing effectiveness of the techniques[a]

No.	C_1		C_2		t	Confidence
1	sim_0	0.385	sim_0 + title	0.415	1.305	89%
2	sim_0 + title	0.415	sim_0 + title + sim_1	0.567	7.330	>99.9%
3	sim_0 + title	0.415	sim_0 + title + sim_2	0.445	1.366	91%
4	sim_0 + title + sim_1	0.567	sim_0 + title + sim_1 + sim_{2a}	0.592	1.320	89%
5	sim_0 + title + sim_1	0.567	sim_0 + title + sim_1 + sim_{2b}	0.608	1.813	96%
6	sim_0 + title + sim_1	0.567	sim_0 + title + sim_1 + sim_2	0.623	2.504	99%
7	sim_0 + title + sim_1 + sim_2	0.623	sim_0 + title + sim_1 + sim_2 + cut	0.630	0.480	65%
8	sim_0 + title + sim_1 + sim_2 + cut	0.630	sim_0 + title + sim_1 + sim_2 + cut + str	0.698	4.290	99.9%

[a] The numbers in column C_1 and C_2 are the averages of the MRR.

5.3. Comparison and evaluation

The performance data of other research works for the named page finding task using the same test collection can be found in (Craswell & Hawking, 2003). Fig. 5 shows the data along with ours (indicated in black). However, the schemes, the components and implementation details of the systems vary. Therefore, simple comparison based on this data cannot be objective enough to evaluate usefulness of the techniques introduced in this paper. For example, some systems used the well-known search engines (as the basic component) made available by others such as the Okapi (Robertson, Walker, & Beaulieu, 2000) or the Smart system (Buckley, Singhal, Mitra, & Salton, 1996). However, some groups including us developed their own search engines and used them for experimentation because special treatments are necessary. There are also some variations in the methods that some data representatives such as titles or links are utilized. Even with this limitation the data in Fig. 5 allows us to do a crude but meaningful comparison. According to this comparison it can be said that our prototype system's performance is quite competitive and the techniques we propose look valuable.

Our system agrees with other top systems in the fact that titles and in-link anchor texts need to be exploited (Craswell & Hawking, 2003; Zhang et al., 2003; Collins-Thompson, Ogilvie, Zhang, & Callan,

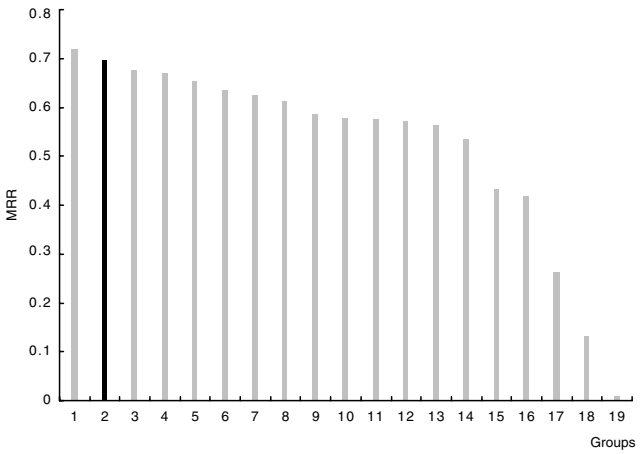

Fig. 5. Performance comparison.

2003; Plachouras, Ounis, Amati, & Van Rijsbergen, 2003; Savoy & Rasolofo, 2003). However, the reports of other systems did not provide any quantitative accounts on how much effective the use of the title section is when compared with other information sources. Our experiment showed that the title section is an important document representative as it can be found in Table 4. This observation is quite different from the previous findings that providing more importance to keywords in the title section did not result in improvement in retrieval effectiveness (Savoy & Picard, 2001). Most recent works acknowledge importance of the title section.

Usefulness of in-link anchor text was pointed out in many works (McBryan, 1994; Brin & Page, 1998; Craswell et al., 2001; Kraaij et al., 2002). This idea has been adopted by most of the recent systems and the experiments have supported the idea. The value of in-link anchor text was also confirmed in our system. Moreover we suggested two different schemes for using in-link anchor texts. They can be used in an integrated way to yield bigger improvement in retrieval effectiveness.

Our research showed that sentence–query similarity has a high potential in enhancing performance. There have been no other research efforts to make use of this information like our approach. According to our experiment this information is most effective for retrieval among the techniques we use including in-link anchor texts. Using sentence–query similarity can be viewed as an approximation to the use of NLU in information retrieval. Realizing that NLU is not expected to be mature enough in the near future to be exploited fully, our scheme of using sentence–query similarity introduced in this paper can be considered to be a good alternative to using NLU.

Another characteristic of our system is to have a stage of stratifying and re-ranking. Applying this stage is more effective than using anchor texts. The source of information used in this stage has a close relationship with sentence–query similarity but is in a different form.

The fact that both using sentence–query similarity and having the stratifying and re-ranking stage improve performance significantly means that they works in some degree for computing similarity between a document and a query.

There is a scheme that has a limited usage. It is a method of cutting off the documents that do not receive contribution from either sentence–query similarity or anchor texts. Its contribution to the performance improvement is rather weak compared to other proposed techniques.

6. Conclusion

This paper introduced several techniques for improving retrieval effectiveness of web information retrieval systems and analyzed their effects. It was found that giving more importance to title section than other sections in web pages leads to performance improvement. We proposed to use a new technique that makes use of sentence–query similarity. It is based on the intuition that if a document has a sentence in which most terms of a query occur the document's relevance tends to get large. It was demonstrated that exploiting links in the form of in-link anchor texts could improve performance a lot. Another effective technique we introduced is to stratify and re-rank the documents in the result list according to the maximum number of index terms in common between a sentence and a query. The effectiveness of this technique was measured to be bigger than that of using in-link anchor texts. Combining these techniques enabled our system to achieve significant improvement in retrieval effectiveness. However, it is noted that these findings are from the named page finding task. Whether they might work in other retrieval tasks such as the normal topic relevance task has to be explored.

The techniques we propose have a limitation. It do not work well for the short queries. Especially for one word query these techniques do not work well. The reason is that these techniques use the number of common words between the query and the strings such as a sentence or anchor text.

Acknowledgements

The authors would like to thank the anonymous reviewers for their detailed comments and suggestions that led to the significant improvement of the paper.

References

Amati, G., & Carpineto, C. (2002). FUB at TREC-10 Web Track: A probabilistic framework for topic relevance term weighting. In *Proceedings of the tenth text retrieval conference TREC-2001* (pp. 182–191). NIST Special Publication #500-250.

Bailey, P., Craswell, N., & Hawking, D. (2003). Engineering a multi-purpose test collection for web retrieval experiments. *Information Processing and Management, 39*(6), 853–872.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international World Wide Web conference* (pp. 107–117). Amsterdam: Elsevier.

Buckley, C., Singhal, A., Mitra, M., & Salton, G. (1996). New retrieval strategies using SMART. In *Proceedings of TREC' 4* (pp. 25–48). NIST Publication #500-236.

Callan, J. P. (1994). Passage-level evidence in document retrieval. In *Proceedings of ACM-SIGIR'1994* (pp. 302–309). New York: ACM.

Clarke, C., Cormack, G., & Burkowski, F. (1995). Shortest substring ranking—multitext experiments for TREC-4. In *Proceedings of the fourth text retrieval conference TREC-4* (pp. 1–10). NIST Special Publication #500-236.

Collins-Thompson, K., Ogilvie, P., Zhang, Y., & Callan, J. (2003). Information filtering, novelty detection, and named page finding. In *Proceedings of the eleventh text retrieval conference TREC-2002* (pp. 107–118). NIST Special Publication #500-251.

Craswell, N., & Hawking, D. (2003). Overview of the TREC-2002 web track. In *Proceedings of the eleventh text retrieval conference TREC-2002* (pp. 86–95). NIST Special Publication #500-251.

Craswell, N., Hawking, D., & Robertson, S. (2001). Effective site finding using link anchor information. In *Proceedings of ACM SIGIR'2001* (pp. 250–257). New York: ACM.

Crestani, F., & Lee, P. L. (2000). Searching the web by constrained spreading activation. *Information Processing and Management, 36*(4), 585–605.

CSIRO (2003). TREC web track home page, http://es.cmis.csiro.au/TRECWeb/.

Fujita, S. (2001). Reflections on aboutness TREC-9 evaluation experiments at justsystem. In *Proceedings of the ninth text retrieval conference TREC-2001*. NIST Special Publication #500-250.

Fujita, S. (2002). More reflections on aboutness TREC-2001 evaluation experiments at justsystem. In *Proceedings of the tenth text retrieval conference TREC-2002* (pp. 331–338). NIST Special Publication #500-251.

Gurrin, C., & Smeaton, A. (2001). Dublin City University at TREC-9. In *Proceedings of the ninth text retrieval conference TREC-2001*. NIST Special Publication #500-249.

Harman, D. (1997). The TREC conferences. In K. Spark Jones & P. Willett (Eds.), *Readings in information retrieval* (pp. 247–256). San Francisco, CA: Morgan Kaufman.

Hawking, D. (2001). Overview of the TREC-9 web Track. In *Proceedings of the ninth text retrieval conference TREC-2001*. NIST Special Publication #500-249.

Hawking, D., & Craswell, N. (2002). Overview of the TREC-2001 web track. In *Proceedings of the tenth text retrieval conference TREC-2001* (pp. 61–67). NIST Special Publication #500-250.

Hawking, D., & Thistlewaite, P. (1996). Relevance weighting using distance between term occurrences. Technical report TR-CS-96-08, Department of Computer Science, The Australian National University.

Hawking, D., Voorhees, E., Craswell, N., & Bailey, P. (2000). Overview of the TREC-8 web Track. In *Proceedings of the eighth text retrieval conference*, NIST Special Publication #500-246.

Kaszkiel, M., & Zobel, J. (1997). Passage retrieval revisited. In *Proceedings of ACM-SIGIR'1997* (pp. 178–185). New York: ACM.

Kleinberg, J. (1999). Authoritative sources in a hyerlinked environment. *Journal of the ACM, 46*(5), 604–632.

Kraaij, W., Westerveld, T., & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In *Proceedings of ACM SIGIR'2002* (pp. 27–34). New York: ACM.

Lim, J.-M., Oh, H.-J., Maeng, S.-H., & Lee, M.-H. (1999). Improving efficiency with document category information in Link-based retrieval. In *Proceedings of the information retrieval on Asian languages conference'1999*.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. The MIT Press.

McBryan, O. (1994). GENVL and WWWW: tools for taming the Web. In *Proceedings of the first WWW conference*. Amsterdam: Elsevier.

Perez-Carballo, J., & Strzalkowski, T. (2000). Natural language information retrieval: progress report. *Information Processing and Management, 36*(1), 155–178.

Plachouras, V., Ounis, I., Amati, G., & Van Rijsbergen, C. J. (2003). University of Glasgow at the web track of TREC 2002. In *Proceedings of the eleventh text retrieval conference TREC-2002* (pp. 645–651). NIST Special Publication #500-251.

Robertson, S. E., Walker, S., & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing and Management, 36*(1), 95–108.

Salton, G. (1989). *Automatic text processing: the transformation, analysis and retrieval of information by computer*. Reading, MA: Addison-Wesley.

Salton, G., Wong, A., & Tang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 614–620.

Savoy, J., & Picard, J. (2001). Retrieval effectiveness on the web. *Information Processing and Management, 37*(4), 543–569.

Savoy, J., & Rasolofo, Y. (2003). Report on the TREC-11 experiment: Arabic, named page and topic distillation searches. In *Proceedings of the eleventh text retrieval conference TREC-2002* (pp. 765–774). NIST Special Publication #500-251.

Singhal, A., & Kaszkiel, M. (2001). AT&T at TREC-9. In *Proceedings of the ninth text retrieval conference TREC-2000*. NIST Special Publication #500-249.

Westerveld, T., Kraaij, W., & Hiemstra, D. (2002). Retrieving web pages using content, links, URLs and anchors. In *Proceedings of the tenth text retrieval conference TREC-2001* (pp. 663–672). NIST Special Publication #500-250.

Zhang, M., Song, R., Lin, C., Ma, S., Jiang, Z., Jin, Y., Liu, Y., & Zhao, L. (2003). THU TREC 2002 Web track experiments. In *Proceedings of the eleventh text retrieval conference TREC-2002* (pp. 591–594). NIST Special Publication #500-251.