Web Search Improvement with Keywords Combination

Web Search Improvement with Keywords Combination

Chao ZENG (Department of Electronics and Information Systems, Fukuoka Junior College of Technology)

Abstract

The World-Wide Web is a large distributed hypertext system on the internet. As the size of the system increases so rapidly the users, who find information by following hypertext links, must traverse increasingly more links to find what they are looking for. Finding information on the Web becomes a very difficult and time-consuming for users. Many pre-computed index based (AltaVista, Lycos, WebCrawler[10]) and on-line (WebCrawler[10], Fish-Search[2, 3], NET-kumo[14, 15]) Web searching tools have been discussed and implemented. In this paper, in order to improve the search effectiveness of on-line search systems, we introduce the concept of keywords separation which considers the user's given keywords as two different types: the first type contains those to be used to check the relevance of documents and the second type, which is attached to some keywords of the first type, contains the keywords which are used as heuristics to lead the search to more relevant next links. An implementation of keywords separation technique to our on-line Web search system NETkumo, stated in our previous papers[14, 15], is described.

Key words: *WWW, on-line search, hueristic, keywords separation, neighborhood information*

1. Introduction

Searching on the Web can be viewed as a relevant document finding process from the document space created by the hyperlinks in documents. When doing search on the Web, the keywords given by user are always used to decide the relevance level of documents to what the user is looking for. Generally given a large number of keywords can describe the query precisely, but at the same time it will cause to retrieve more irrelevant documents. In pre-computed index based (AltaVista, Lycos, WebCrawler[10]) search systems, the keywords are just needed to be used to determine the document's relevance. But in on-line Web searching tools, the keywords should be used as not only the criterion of document's relevance, but also the search heuristics that users can pass to the system for their specific searching task. The researches about on-line searching tools until now, such as WebCrawler[10], Fish-Search[2, 3], NETkumo[14, 15] did not consider the aspect of using keywords as search heuristics particularly. But when developing on-line searching systems for large network distributed information system such as the Web, how to efficiently lead the search to the most possible directions of informations and find the relevant documents becomes the main matter. In papers[14, 15], we proposed some search heuristics to determine the relevance levels of the outgoing links in a document. The implementation of them is the on-line search system NETkumo. The experiments on NETkumo showed satisfactory results. In this paper, in order to improve the search effective-

Received October 31, 1997.

ness further, we introduce the concept of *keywords separation* which considers the users keywords as two different types: the first type contains those to be used to check the relevance of documents as usual and the second type, which is attached to some keywords of the first type, contains the keywords which are used mainly as heuristics to lead the searching to more relevant next outgoing links in documents in the search space. The significance of separating keywords is on the point that the searching system can use keywords in different aims: determining relevance and finding documents. Up to the present, keywords are primarily used on determining relevance of document. But as pointed in[4], when searching on large document space, finding document is more important. In order to avoid retrieving too many irrelevant documents, some keywords are better to be used specially as heuristics to select the next visiting links. For example, considering that we are looking for informations about Prof. Knuth at computer science department of Stanford University. Suppose we start from the home page of Stanford University. The searching keywords may be given as [*knuth computer department faculty staff*]. If we deal with all the keywords as same(the first type), many faculty or staff pages of other departments may be retrieved fruitlessly since *faculty* and *staff* are used to decide the document relevance. But if we simply remove them from the keywords, maybe the corresponding pages of computer science department will be missed too. Obviously, in [*knuth computer department faculty staff school*], it is better to consider keywords *faculty* and *staff* as some of the second type attached to keywords [*computer science*] and utilize them as heuristic to select the links about staffs or faculties when the search reached some pages in which keywords *computer* and *science* are contained. The technique of *keywords separation* described in this paper realizes the concept. Another example to explain the idea well is the search of computer command *cat* problem mentioned in some literatures[9, 7]. It can be solved by considering keyword *cat* to be attached to keyword *computer*.

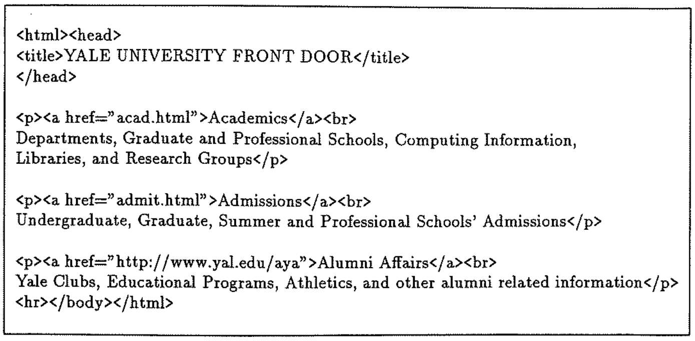
This paper is organized as follows. In Section 2 we present a review of search heuristics proposed in our previous papers and the on-line system *NETkumo* based on them. Then, in Section 3, we give a detailed description of *keywords separation* and an user interface to the NETkumo search system. An implementation of keywords separation will be discussed in Section 4. In Section 5, a summary and some concluding remarks will be presented.

2. NETkumo: an on-line search tool

NETkumo is similar to the Fish-Search[2, 3, 4] but functionally more simple. In order to improve the searching effectiveness on the Web, NETkumo uses some hueristics which are fit for the hypertext structure system. As starting point, one or a set of documents is giving to the system with their URLs, and a sequence of searching keywords follow as the query. The relevance of a document is evaluated by lexical similarity with a count of word's occurrences. The searching result is an ordered list of relevant documents which contain all the given keywords or some of them. In this section, we will give a short review on the heuristics used in NETkumo. A detailed description about NETkumo can be found in the papers[14, 15].

2. 1 Use of hot text

In HTML-format files as used in the Web, link is realized as the form of <a href="URL"> Hot Text</a>, where URL is the internet address of the linking destination, and the *Hot Text* usually contains a brief sentence describing what the content of linking destination is(see Figure 1). Giving a consideration to the similarity of the query and the hot text will give a good hint to select more relevant links to search. Since usually the hot text is just a short sentence, using lexical similarity evaluation can not make a full



```
<html><head>
<title>YALE UNIVERSITY FRONT DOOR</title>
</head>

<p><a href="acad.html">Academics</a><br>
Departments, Graduate and Professional Schools, Computing Information,
Libraries, and Research Groups</p>

<p><a href="admit.html">Admissions</a><br>
Undergraduate, Graduate, Summer and Professional Schools' Admissions</p>

<p><a href="http://www.yal.edu/aya">Alumni Affairs</a><br>
Yale Clubs, Educational Programs, Athletics, and other alumni related information</p>
<hr></body></html>
```

Figure 1: A sample html file

utilization of it. Some other more precise evaluation method, such as semantic analysis based on a thesaurus will give a better result.

Since the hot text is a direct description about the content of link, the evaluation result based on it should be regarded as important. At the same time, since its briefness the relevance evaluation of it based on a simple lexical similarity evaluation method is often fruitless.

2. 2 Use of neighborhood information

Besides the hot text, in NETkumo we also consider the use of what we called as *neighborhood information* of link in a document to evaluate the link's relevance level to the query. When writing HTML file, as shown in Figure 1, many people usually append some longer description to each link in front or(and) behind the link to explain the link's content. We define a link's *neighborhood* in a document as the texts in the place immediately before and after the link, and call the texts as the link's *neighborhood information* in the document. Neighborhood information of link is used to evaluate the relevance level of the link to the user's query.

The scope of link's neighborhood information in a document is determined according to the number of characters and links contained in the document and simply calculated by the following formula.

the scope of neighborhood information=

$$\frac{\text{the number of characters}}{\text{the number of links}} \times 2$$

In NETkumo system, evaluating similarities of hot text or neighborhood information for links and similarities of documents to the user's query (set of keywords) is based on a matching process with a stemming treatment on the keywords according to the Porter's stemming algorithm[11] with some extended exception lists. The frequency of keyword's occurrences in hot texts or neighborhood informations, and in documents is used to decide the relevance levels of hot texts or neighborhood informations, and documents respectively.

3. Keywords separation

The process of information search can be divided into three steps: *finding documents, formulating queries,* and *determining relevance.* Many researches have been done about the *formulation of queries* and the *evaluation of document's relevance* in the literatures in the area of information retrieval[8]. As pointed out by De Bra[4], differing from other database-based Web search systems, in an on-line Web search system how to effectively find the interesting documents in the large Web document's space is most important and unfortunately, always a bottleneck. It is needed that the system should be given some searching heuristics from the user him(her) self to do the user's individual search problem. These

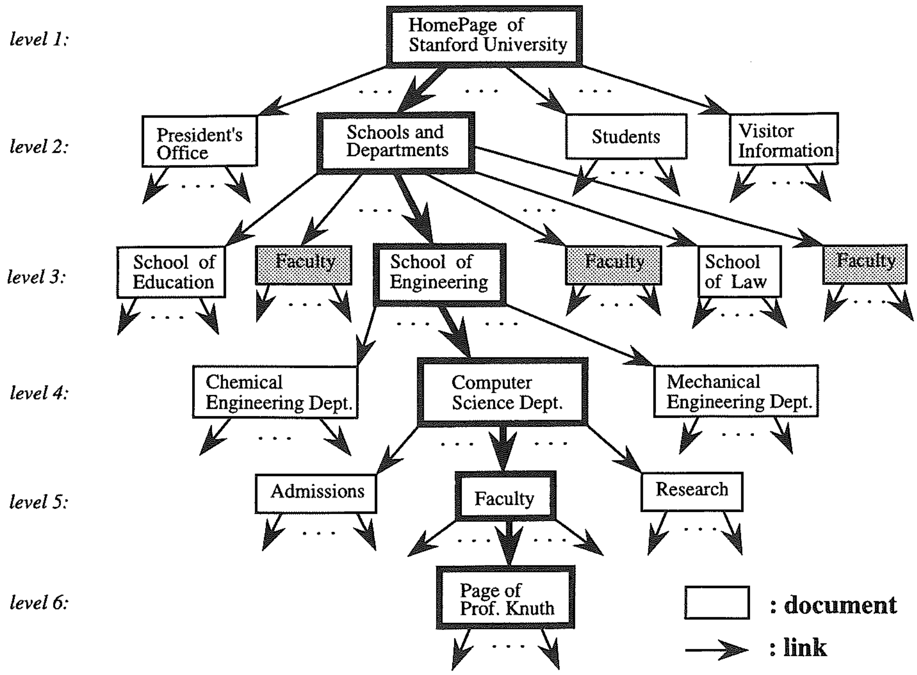


Figure 2: A part of document's search space

heuristics will be used in the search process to guide the search to the more relevant documents in the document's space (see Figure 2).

In order to improve the effectiveness of finding relevant documents[4] more efficiently, getting some more informations about user's information needs from the users themselves is indispensable. We propose here the concept of *keywords separation* to give a means to the users to transform their search heuristics to the search system. The term of keywords separation divides the user's given keywords into two types and provides a mechanism to integrate them for users to define and describe their information needs more precisely to the search system. The first type, we call them as *ordinary keywords,* contains the keywords that should be used to evaluating the relevance and finding documents as usually. The second type, called as *attachment keywords,* contains those keywords which are attached to some ordinary keywords. The attachment keywords are only used in search when their ordinary one(s) appeared in a document. Separating the user's keywords into as ordinary and attachment ones makes its significances in the situation when we are looking for somebody who is a member of the faculty of *computer science department,* but not any other departments. Since in a document of department or school, there usually contains a link about the members of the faculty with the link name as *faculty* or *staff.* At this time we would like to attach the keywords *faculty* and *staff* to the keywords *comput depart* to avoid the fruitless search on the faculty documents of other department which is not computer science department. As in Figure 2, at level 3, which is the content of document of *Schools and Departments,* corresponding to each link of School there is a link named as *Faculty* about the faculties of the school. So there are many links of Faculty contained in the document of *Schools and Departments.* If the word *faculty* is used as one of the search keywords at the step of level 3, many fruitless search will be done on the other school's faculty documents.

Another example to express the idea of keywords separation good is the *cat* and *computer* search mentioned in some literatures[9, 7]. Suppose we want to search some information about the computer command *cat,* if we give the keywords as [comput cat] many documents about the animal Cat should be retrieved contrary to the user' s expectation. Under the consideration of keywords separation, we can give *cat* attached to *computer,* then *cat* is used as keyword to evaluate a document only when ordinary keyword *computer* occurred in the document.

Here we simply give some notations to express the term of keywords separation more formally, and by using it an implementation to our NETkumo system will be described clearly. We use alphabets in upper case, such as A, B, to denote a *query set* that contains at least one element defind as follows. *Element* of a query set is either a single word or an attachment pair defined as follows. A *Boolean expression,* denoted by Greek characters, is a combination of words, parentheses and the connectives **&&** and ||. **&&** and || take respectively the means of logical AND and OR. Query set contains just single words is especially called as *atomic query set.* For example, A=(computer depart school knuth), B=(faculty staff) are two atomic query sets and α=(computer**&&**(depart||school)) is a Boolean expression. We call symbol -> as the *attachment operation.* α->B, which we call as an *attachment pair,* means the query set B is attached to α, where α, called as the *attaching body* of the attachment pair, is a Boolean expression and B, called as the *attached head* of the attachment pair, is an atomic query set. Attachment pair is used as user given keywords cooperated with keywords separation. For example,

C=(α->B knuth)=
(computer**&&**(depart||school)->(faculty staff) knuth)

is a query set which contains a single keyword and an attachment pair.

In an attachment pair, the attaching body is treated as ordinary keywords and the attached head is treated as the attachment keywords of those in the attaching body. For a simple attachment pair, such as (computer depart)->(faculty staff) keywords *computer* and *depart* are used as ordinary ones and *faculty* and *staff* are attachment keywords to them. By some complicated attachment pair, more complex attachment relations can be expressed. For example, the attachment pair

computer**&&**(depart||school)->(faculty staff)

means that keywords *faculty* and *staff* are attached to *computer* and *depart,* and *computer* and *school* simultaneously. All of the keywords in the attaching body are treated as ordinary keywords in the searching process.

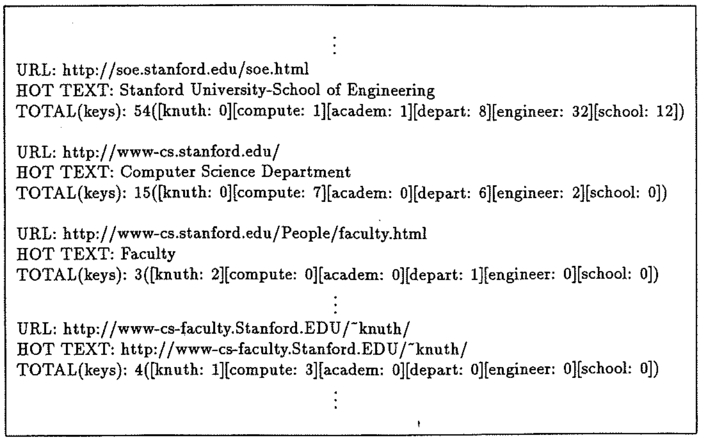
4. Implementation

The idea of keywords separation is now implemented in NETkumo starts the search by taking one or more starting point (an URL) and some keywords separated by space as its arguments as follows.

```
%NETkumo http://www.stanford.edu/ knuth compute
         depart faculty academ school engineer
```

NETkumo executes under the depth-first strategy, which is experimentally demonstrated having a good performance on on-line search[5, 3], with a changing branching factor which limits the number of links to be selected from a retrieved document according to the document's relevance level. Some other features discussed in Fish-Search such as the limited depth in a direction in which no relevant information is found, and the consideration of network access rate to avoid to spent too time to access very slow sites are also implemented.

An analyzer routine analyzes the attachment pairs used in the command line attributes of NETkumo as searching keywords. For each attachment pair, the words in the attaching body are used as ordinary keywords to evaluate docu-



```
                                        ⋮
URL: http://soe.stanford.edu/soe.html
HOT TEXT: Stanford University-School of Engineering
TOTAL(keys): 54([knuth: 0][compute: 1][academ: 1][depart: 8][engineer: 32][school: 12])

URL: http://www-cs.stanford.edu/
HOT TEXT: Computer Science Department
TOTAL(keys): 15([knuth: 0][compute: 7][academ: 0][depart: 6][engineer: 2][school: 0])

URL: http://www-cs.stanford.edu/People/faculty.html
HOT TEXT: Faculty
TOTAL(keys): 3([knuth: 2][compute: 0][academ: 0][depart: 1][engineer: 0][school: 0])
                                        ⋮
URL: http://www-cs-faculty.Stanford.EDU/˜knuth/
HOT TEXT: http://www-cs-faculty.Stanford.EDU/˜knuth/
TOTAL(keys): 4([knuth: 1][compute: 3][academ: 0][depart: 0][engineer: 0][school: 0])
                                        ⋮
```

Figure 3:　Returned search results by NETkumo

ments and links, and the words in the attached head are used as ordinary ones only when the words in the attaching body occurred. For example,

```
%NETkumo http://www.stanford.edu/
         computer&&(depart||school)->(faculty staff)
         knuth academ engineer
```

starts NETkumo from the starting point

http://www.stanford.edu/

with the query as

computer&&(*depart*||*school*)->(*faculty staff*)
knuth academ engineer,

where one attachment pair contained.

Figure 3 is a part of the search results returned by NETkumo.

5. Summary and concluding remarks

In information search, especially on large distributed information system such as the Web, for improving the search effectiveness of search system it is needed to offer a mechanism for users so that they are capable of formulating the set of keywords to retrieve the wanted information. Queries should be needed to contain not only the keywords or some boolean expressions of them, but also some informations to express the mutual relationship of the keywords. In this paper, we proposed and discussed the keywords separation which provides a way to express the mutual relationship among the keywords given by users. In a Web search system, keywords separation can make the relevance evaluation of documents to be executed more exactly and lead the search to the documents that users really want effectively. An implementation of keywords separation has been done to the on-line search system NETkumo, and the experiments with NETkumo shows it to be practical and effective.

Keywords separation differs from the AND connective in boolean queries. *keyword1* AND *keyword2* means that the document should contain both of the two words. But the attachment pair *keyword1*->*keyword2* means that *keyword2* is used as a keyword only when *keyword1* appeared in the document in advance. Obviously, it also differs the logical connective → where *keyword1* → *keyword2* means (*NOT keyword1*) *OR keyword2.*

It is easy to express keywords separation using the HTML (HyperText Markup Language) form tag. As a future work, a CGI program will be written and it serves as a gateway to the NETkumo system. With the input forms implemented in almost all WWW browers, an easy to use interface of keywords separation will be provided.

References

[1] M. Agosti and A.F. Smeaton (Eds.): "Information Retrieval and Hypertext", Kluwer Academic Publishers, 1996.

[2] P.M.E. De Bra and R.D.J. Post: "Searching for Arbitrary Information in the WWW: the Fish-Search for Mosaic", 2nd International World-Wide Web Conference, http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/debra/article.html.

[3] P.M.E. De Bra and R.D.J. Post: "Information Retrieval in the World-Wide Web: Making client-based searching feasible", Computer Networks and ISDN Systems, 1994, 183-192.

[4] P.M.E. De Bra: "Finding Information on the Web", http.//wwwis.win.tue.nl/~debra/cwiqw/article.html.

[5] J.W. de Vocht: "Experiments for the characterization of hypertext structures", Master's thesis, Eindhoven Univ. of Technology, Apr. 1994.

[6] T. Berners-Lee, R. Fielding and H. Frystyk: "Hypertext Transfer Protocol—HTTP/1.0", RFC1945, May 1996.

[7] E. Berk, and J. Devlin (Eds.): "Hypertext/Hypermedia Handbook", New York: McGraw-Hill, 1991.

[8] K. Sparck Jones and P. Willett (Eds.): "Readings in Information Retrieval", Morgan Kaufmann Publishers, 1997.

[9] R.R. Korfhage: "Information Storage and Retrieval", Wiley Computer Publishing, 1997.

[10] B. Pinkerton: "Finding What People Want: Experiences with the WebCrawler", the Second WWW conference, Chicago, 1994.

[11] M.F. Porter: "An algorithm for suffix stripping", Program 14: 130-137, 1980.

[12] G. Salton and M.J. McGill: "Introduction to Modern Information Retrieval", McGraw-Hill, 1983.

[13] N.F. Zhou and C. Zeng: "Cooperative Search in Logic Programming", 1995 Beppu Summer United Workshop on Parallel Distributed, and Cooperative Processing, Technical Report of IEICE, Japan, Vol.95 No.211, 1995, 25-31.

[14] C. Zeng: "A Client-based Information Retrieval System on Internet", Proceedings of the 11th Annual Conference of Japanese Society for Artificial Intelligence, 1997, 474-477.

[15] C. Zeng: "An Implementation of Client-based Retrieval System by Use of Neighborhood Information", Research Bulletin of Fukuoka Institute of Technology, Japan, Vol.30, No.1, 1997, 11-18.