Mika Käki

*f*KWIC: Frequency Based Keyword-in-Context Index for Filtering Web Search Results

DEPARTMENT OF COMPUTER SCIENCES
UNIVERSITY OF TAMPERE

D-2005-3

TAMPERE 2005

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER SCIENCES
SERIES OF PUBLICATIONS D – NET PUBLICATIONS
D-2005-3, JUNE 2005

Mika Käki

*f*KWIC: Frequency Based Keyword-in-Context Index for Filtering Web Search Results

DEPARTMENT OF COMPUTER SCIENCES
FIN-33014 UNIVERSITY OF TAMPERE

ISBN 951-44-6341-2
ISSN 1795-4274

*f*KWIC: Frequency Based Keyword-in-Context Index for Filtering Web Search Results

Mika Käki
Department of Computer Sciences
FIN-33014 University of Tampere, Finland
mika.kaki@cs.uta.fi
Tel. +358-3-215 6181
Fax. +358-3-215 6070

ABSTRACT
Enormous web search engine databases combined with short search queries result in large result sets that are often difficult to access. Result ranking works fairly well, but users need help when it fails. For these situations, we propose a filtering interface that is inspired by keyword-in-context (KWIC) indices. The user interface lists the most frequent keyword contexts (*f*KWIC). When a context is selected, the corresponding results are displayed in the result list allowing users to concentrate on the specific context at a time. We compared the keyword context index user interface to the rank order result listing in an experiment with 36 participants. The results show that the proposed user interface was 29% faster in finding relevant results and the precision of the selected results was 19% higher. In addition, participants showed positive attitudes towards the system.

KEYWORDS
Web search, search user interface, keyword-in-context index, result filtering, information access.

1 INTRODUCTION
Web search engines are one of the most important ways of finding information from the World Wide Web (web). Jacob Nielsen recently stated that in 88% of the cases, users start web navigation by using a search engine (Nielsen, 2004). Even though the use of search engines is popular, people cannot always use them optimally. Log studies have shown that the topics that users are searching for have changed over time, but the search skills and habits have remained largely the same using only a few words in a query (Spink, Jansen, Wolfram, & Saracevic, 2002). When these kinds of queries are combined with enormous databases hosted by the web search engines (Google has reported to index over 4 billion web pages in the end of year 2002), it is apparent that search queries will fail from time to time.

Certainly the problem is not newly discovered. Research is done, for example, in the areas of information retrieval, result ranking (Brin & Page, 1998) and result visualization (Kartoo Search Engine) to overcome the problems. Result categorization studies (Zamir & Etzioni, 1998; Zamir & Etzioni, 1999; Vivisimo Search Engine, Dumais & Chen, 2000; Chen & Dumais, 2000; Käki & Aula, 2005) have proposed solutions for accessing the result lists more effectively. In addition, query refining aids (Anick, 2003) try to reduce the problem in the query formulation side.

We propose frequency based keyword context index (*f*KWIC), a new solution inspired by the keyword-in-context (KWIC) indices. *f*KWIC extracts the most frequent keyword contexts with a new algorithm. The contexts are employed in a filtering user interface that provides an easy access to interesting results. The user interface is build around conventional ranked result listing and the keyword context index is provided as an additional tool for coping in situations where result ranking does not support user's information need. The solution is expected to be most useful when the initial query formulation is too vague or when the user is engaged in an undirected informational search task where a broad understanding of a topic is required.

To find out whether keyword context index is beneficial in actual use, we conducted an experiment with 36 participants. The experiment compared the *f*KWIC user interface to the ranked result list solution (*de facto* standard) with 18 (9 + 9) search tasks. The results show that the new user interface is advantageous, especially when multiple results are needed. Users were 29% faster and 19% more accurate (in terms of precision) with the new system compared to the control condition (ranked list). In addition, the first relevant result is found with fewer selections. This is important in typical web searches where the first relevant result may be the only one the user retrieves.

The paper presents first a review of relevant literature in the field. Next, the algorithm for building the keyword context index is explained, and the user interface design decisions are discussed. This is followed by the explanation of the experiment and the presentation of its results. In the end, conclusions are drawn.

2 RELATED WORK

Two areas of research are most relevant for this work. First, the work done in developing keyword-in-context indices has been a major inspiration for our design. Second, the way in which the new user interface solution is applied, is largely based on the work done in web search result categorization.

2.1 Keyword-in-Context (KWIC) Indices

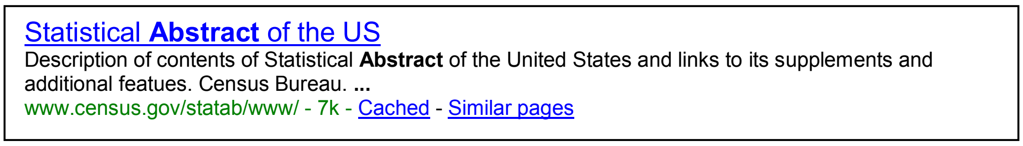
Keyword-in-context (KWIC) index is a form of concordance (word index) where each occurrence of the keyword is displayed together with surrounding words in a list of strings. An example index could list a set of book titles as a response to a library database query. The keyword used in the query is the basis for the index, while the remaining words in the title form the context for it (Figure 1). In these indices, the keyword is typically printed aligned in the middle to make visual scanning of the list easier. Hans Peter Luhn published a paper on KWICs already in 1959.

```
graphic scheme based on  abstract and index cards
  tic information using  abstract and index publications
                         abstract archive of alcohol lireratu
      publishing modern  abstract bulletins
 company pharmaceutical  abstract bulletin
         a punched card  abstract file on solid state and tra
                    the  abstract of the technical report
         relation of an  abstract to its original
from journal article to  abstract
```

Figure 1. An example of a Keyword-in-Context (KWIC) index where the keyword is *abstract* (after Salton (1989)).

In a recent application of the idea, similar word listings were used to display key phrases in a web site user interface (Paynter, Witten, Cunningham, & Buchanan, 2000). Their user interface shows the search results in a list where the keyword is first accompanied by one context word (preceding or subsequent). When user selects one of these contexts, the second list displays three words for each hit: two selected keywords and one additional context word. This interaction forms a hierarchical index where users browse to the desired content by sequential and narrowing context selections.

Popular way of displaying web search engine result summaries (snippets) can also be seen as a KWIC index. The summaries consist of phrases selected from the source text so that the query keyword is shown in a context (Figure 2). The keywords are not aligned spatially, but they are highlighted with a bold face type. This design saves considerable amount of screen space and delivers rich context information.



Statistical **Abstract** of the US
Description of contents of Statistical **Abstract** of the United States and links to its supplements and additional featues. Census Bureau. **...**
www.census.gov/statab/www/ - 7k - Cached - Similar pages

Figure 2. An example of modern counterpart of KWIC index from Google search engine (Google Search Engine), keyword is *abstract.*

2.2 Web Search Result Categorization

So called *cluster hypothesis* (Jardine & Rijsbergen, 1971) states that relevant documents for a query tend to be similar with each other. If such clusters are available, searcher's task is first to locate the relevant cluster(s) and then evaluate only the documents within that cluster(s). This reduces the amount of documents that need to be evaluated and makes the search process more effective. The great promise has inspired a substantial amount of research in the area and the interesting question is if, and how, such a clustering can be made automatically.

In this paper, the concept of *clustering* is used to describe a technique of automatically bringing similar documents together. *Classification*, on the other hand, refers to a technique of putting documents into predefined categories. Both of them can be used to categorize the documents (or search results) in order to realize the promise of the clustering hypothesis.

Scatter/Gather was one of the first systems where the cluster hypothesis was tested with end users and it showed that the clustering approach is technically feasible (Cutting, Karger, Pedersen, & Tukey, 1992). A later user study showed more evidence for the cluster hypothesis as the users were able to select the most relevant clusters during the search process (Hearst & Pedersen, 1996). It meant that the users found the most valuable information sources in that user interface and that the clustering could enhance the user performance.

Recently, a lot of research effort is put to categorizing web search results. Zamir and Etzioni (1998) demonstrated the benefits of a special type of clustering technique in the web environment. They also implemented a web search engine user interface, Grouper, based on the idea (Zamir & Etzioni, 1999). Our Findex system aims to be simpler and more transparent system for the users. It is shown to be beneficial in a laboratory and in a longitudinal study (Käki & Aula, 2005; Käki, 2005). Zhen *et al*. (2004) have improved the quality of the categories by utilizing learning techniques in their clustering algorithm.

In contrast to all the previous prototypes that were based on clustering, Swish and Dynacat prototypes use the classification approach. This approach solves the problem of naming the categories sometimes associated with the clustering techniques. Swish prototype works in the web environment by using a pre-taught classification scheme (Dumais, Cutrell, & Chen, 2001). Dynacat, in contrast, uses a predefined classification that is based on the rich metadata attached to the medical documents (Pratt & Fagan, 2000).

In addition to the categorization techniques, it is possible to focus on the structure of the resulting categories. There are two main types: hierarchical (like folders in the file system) and flat (a list of categories). Many of the above-mentioned systems employ a flat category list, but Scatter/Gather and Dynacat present hierarchies to the users. Swish uses hierarchical classification system, but result categories form a flat list. In addition to these, Ferragina and Gulli (2005) and Kummamuru (2004) have proposed systems that present an explicit hierarchy of categories in a form of a folder structure like in modern file systems.

3 SYSTEM DESCRIPTION

3.1 Building Frequency Based Keyword Context Index

We took the result pages returned by the search engine as the starting point for our design. This means that the full text of the result documents is not available, but only short summaries (snippets) of them. Although it can be seen as a restriction, it also makes it possible to employ the system on any search engine that produces this kind of results. In addition, the snippets are show to be a feasible source of categorization by Zarmir and Etzioni (1998).

Although the original layout of the KWIC index is simple and easy to understand, it has some weaknesses. First, the original form considers only document titles whereas web searches are based on full text search. This means that the matching document titles may not contain any of the query keywords and thus, displaying the keyword contexts in the titles is impossible. Second, short text summaries that display the keyword contexts in the document body improve users' speed and accuracy in search tasks (Tombros & Sanderson, 1998), but such summaries are not available in original KWIC indices. Third issue concerns the screen real estate. KWIC indices were designed at the time when displays were character based. Modern graphical environments are built from a different perspective and thus the original layout of the KWIC indices may not be optimal for the web search results.

To address these issues and to utilize the idea in a modern environment, we altered the original form of KWIC index by emphasizing the index (table of contents) side of the idea. To make the index compact enough, we present the users only with the most frequently occurring keyword contexts. This solution was expected to 1) save valuable screen real estate, 2) let users take advantage of the text summaries, and 3) let users access interesting results easier. All these points are vital in making the result access more efficient.

The computation of the most frequent keyword contexts (*f*KWIC) starts by listing all phrases within result sentences that contain a keyword. If a snippet does not contain a full sentence (from period to period), then the partial sentence is used. A phrase, on the other hand, is a string of words within a sentence. The length of these phrases is two or more words, maximum being the length of the sentence containing the keyword (context sentence). As the phrases are created, stop words (we use publicly available stop word lists) are excluded. For each phrase, information about the associated result is stored. When the list of candidate phrases is complete, instances which are associated with two results or less are removed.

The next phase reduces repetition and redundancy in the keyword contexts and aims to ensure proper coverage. First, similar candidate phrases are merged together. Similar phrases are those that are composed of same words (possibly in different order). Queries composed of proper names often produce phrases where the names appear in various orders, which make them redundant and distracting. In addition to ignoring the order of the words, we ignore simple inflections of the words. For this, we use a non-exact string matching algorithm. This algorithm allows strings to differ three characters in length and requires only the 80% (of the shorter string) of the first characters to match. For example, words “house”, “houses”, and “housing” are considered to be the same (four first characters are the same, the length differs two characters at the most). We acknowledge the danger of oversimplification of this approach, but it is found to work satisfactorily in Finnish, English, French and German.

Second, sub and super-phrases are processed. Sub-phrase is a phrase that appears inside a longer phrase, such as ‘dog breed’ is a sub-phrase of ‘dog breed information’ (two possible keyword contexts for keyword ‘dog’). The sub-phrases are removed if they are associated only with the same results (overlapping) as the super-phrase. If the number of non-overlapping results associated with the sub-phrase is less than half of the number of results associated with the super-phrase, the sub-phrase is removed from the candidate list. Thus, the sub-phrase must contribute enough to the coverage in order to get selected. For example, if ‘dog breed information’ appears in six results, ‘dog breed’ must appear at least in nine results to be kept in the candidate list. Otherwise, the overlap of the keyword contexts is considered to be too great.

Finally, the candidate phrases are sorted according to the number of results that are associated only with the given phrase. This is done to maximize the coverage of the phrases in relation to the results. The top *n* (by default 15) phrases of this final list are selected and shown to the user as the keyword context index.

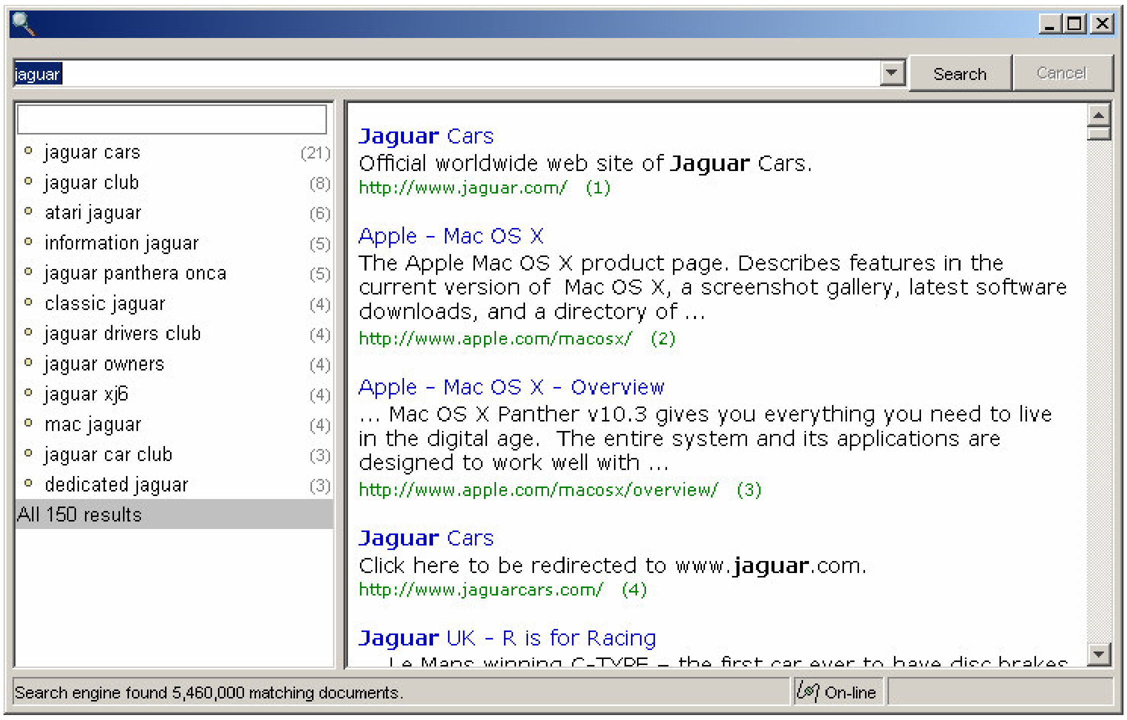
The items in the final keyword context index can be overlapping, meaning that one result can be associated with more than one keyword context. This occurs simply because one result may contain many keyword contexts. This may be confusing if *f*KWIC is thought strictly as an index that covers the whole contents of the indexed result set. On the other hand, this functionality makes the information accessible in various contexts and thus makes the recognition of the relevant information easier.

This technique does not guarantee that all results can be found from the index. In this respect, we favored understandability over the coverage in our design. This shortcoming is solved in the user interface with a functionality that allows users to see the whole result list.

The performance of the algorithm for web environment is of great importance. Our algorithm performs acceptably when the server load is relatively low. It takes about 250 milliseconds (*sd* = 182, n = 74 queries) to compute the keyword context index for 150 results. In our experience, 150 results is a reasonable tradeoff between speed and coverage. The final keyword contexts tend to remain roughly the same even if the number of results is increased.

3.2 User Interface

The idea of adding new features on top of the existing search services is followed in the user interface as well. The current ranked list presentation of the search results works reasonably well in many cases. Thus, we wanted to let people utilize their existing knowledge and experience with it. As a result, the conventional result list is the core of



jaguar cars	(21)
jaguar club	(8)
atari jaguar	(6)
information jaguar	(5)
jaguar panthera onca	(5)
classic jaguar	(4)
jaguar drivers club	(4)
jaguar owners	(4)
jaguar xj6	(4)
mac jaguar	(4)
jaguar car club	(3)
dedicated jaguar	(3)
All 150 results	

Jaguar Cars
Official worldwide web site of **Jaguar** Cars.
http://www.jaguar.com/ (1)

Apple - Mac OS X
The Apple Mac OS X product page. Describes features in the current version of Mac OS X, a screenshot gallery, latest software downloads, and a directory of ...
http://www.apple.com/macosx/ (2)

Apple - Mac OS X - Overview
... Mac OS X Panther v10.3 gives you everything you need to live in the digital age. The entire system and its applications are designed to work well with ...
http://www.apple.com/macosx/overview/ (3)

Jaguar Cars
Click here to be redirected to www.**jaguar**.com.
http://www.jaguarcars.com/ (4)

Jaguar UK - R is for Racing
Le Mans winning C-TYPE – the first car ever to have disc brakes

Search engine found 5,460,000 matching documents.

On-line

Figure 3. Screenshot of the keyword context (*f*KWIC) user interface, list of the most frequent keyword contexts on the left, result listing on the right.

our user interface and it is enhanced by the automatically computed keyword context index on the side (Figure 3).

Keyword context index is used to filter the result listing. When the user selects one keyword context, the result listing is updated to show only those items that contain the selected context. The relationship between the context and the result item is explained by highlighting the context instances in the result text.

In addition to the automatically computed keyword contexts, there is a special built-in item in the index for accessing all the retrieved results. This ‘All results’ item is automatically selected after each search so that initially the user will always see both the keyword context index and all of the results. This makes the default interaction with the system close to the traditional search engine, but new advanced features are readily available.

The visual format of the keyword context index was altered from the original KWIC design. Although the ability to easily scan the index along the aligned keywords is central in the idea, it would have required too much screen space in our layout. Thus, we simply list the most common keyword contexts left aligned.

The order of the results is always determined by the underlying search engine. When one of the keyword contexts is selected, the relative order of the results follows the original order of the search engine, although the results are typically not sequential in the original listing.

One important issue in search user interfaces is the perceived speed of the system. If the system is slow, users may start to hesitate in making queries. The result categorization could decrease the responsiveness, because it has to download many results before the context computation can be completed. To address this issue, we built the user interface so that the user sees the first ten results as soon as the underlying search engine returns them. Result retrieval continues in the background and the keyword context index is displayed automatically when the computation finishes. This does not enhance the actual speed of the system, but the user can use the waiting time effectively making the system appear faster.

4 EXPERIMENT

To evaluate the performance of the *f*KWIC algorithm and the user interface, we conducted an experiment. The experiment was designed to compare the new solution to the reference solution. As the reference user interface we used the ranked search result list that is popular in commercial web search engines.

4.1 Participants

We had 36 participants in the study (28 male, 8 female). The participants were recruited from an introductory level human-computer interaction class where the students were required to participate in a study. The average age of the participants was 24.5 years (*sd* = 4.2) ranging from 20 to 35 years. All participants can be regarded as experienced computer users. They reported to have used computers, on average, for 11.5 years (*sd* = 4.1) and web for 6.9 years (*sd* = 2.2). They said to engage in computer, web, and web search engine use daily (mode).

4.2 Design

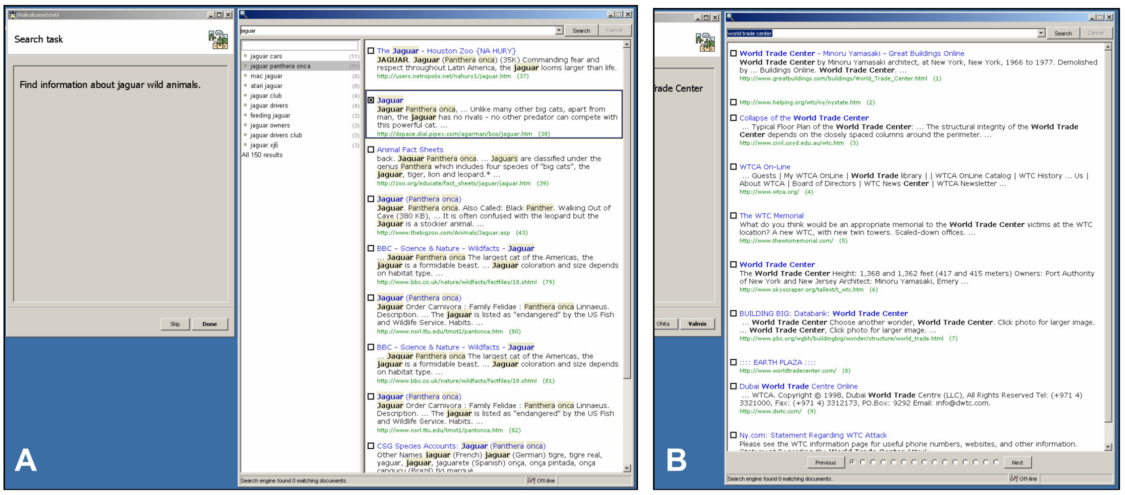
The user interface (UI) was the only independent variable in the experiment. It had two values: 1) keyword context UI and 2) reference UI. The value of the independent variable was manipulated within each participant and thus the measures were analyzed using repeated measures analysis tools. The order of the presentation of the user interfaces was counterbalanced between the participants.

Because search tasks are contaminated upon execution (one task can not be carried out twice by the same participant), there were two task sets. The task sets were designed to be equally demanding. For eliminating the possible differences between the task sets, the order in which they were presented was counterbalanced as well.

As dependent variables we measured search speed, accuracy, and subjective satisfaction. Speed measures were based on result selection times and accuracy on judgments about the result items made by the experimenter. Subjective opinions were elicited with questionnaires.

4.3 Apparatus and Materials

The experiment setup was partly automated. There was a special purpose software application that presented the tasks and collected an event log of the user interactions. During the experiment there were two windows on the computer screen (Figure 4, A). *Task window* was located on the left side and it presented the tasks and questionnaires to the participants. It also controlled the *search window* (on the right) that displayed the tested user interface.



Find information about jaguar wild animals.

- jaguar cars (11)
- jaguar panthera onca (11)
- mac jaguar (8)
- atari jaguar (8)
- jaguar club (4)
- jaguar drivers (4)
- feeding jaguar (3)
- jaguar owners (3)
- jaguar drivers club (3)
- jaguar xj6 (3)

All 150 results

☐ The Jaguar - Houston Zoo {NA HURY}
JAGUAR. Jaguar (Panthera onca) (35K) Commanding fear and respect throughout Latin America, the jaguar looms larger than life.
http://users.netropolis.net/nahury1/jaguar.htm (37)

☒ Jaguar
Jaguar Panthera onca, ... Unlike many other big cats, apart from man, the jaguar has no rivals - no other predator can compete with this powerful cat. ...
http://dspace.dial.pipex.com/agarman/bco/jaguar.htm (38)

☐ Animal Fact Sheets
back. Jaguar Panthera onca. ... Jaguars are classified under the genus Panthera which includes four species of "big cats", the jaguar, tiger, lion and leopard.* ...
http://zoo.org/educate/fact_sheets/jaguar/jaguar.htm (39)

☐ Jaguar (Panthera onca)
Jaguar. Panthera onca. Also Called: Black Panther. Walking Out of Cave (380 KB), ... It is often confused with the leopard but the Jaguar is a stockier animal. ...
http://www.thebigzoo.com/Animals/Jaguar.asp (43)

☐ BBC - Science & Nature - Wildfacts - Jaguar
... Jaguar Panthera onca The largest cat of the Americas, the jaguar is a formidable beast. ... Jaguar coloration and size depends on habitat type. ...
http://www.bbc.co.uk/nature/wildfacts/factfiles/18.shtml (79)

☐ Jaguar (Panthera onca)
Jaguar Order Carnivora : Family Felidae : Panthera onca Linnaeus. Description. ... The jaguar is listed as "endangered" by the US Fish and Wildlife Service. Habits. ...
http://www.nsrl.ttu.edu/tmot1/pantonca.htm (80)

☐ BBC - Science & Nature - Wildfacts - Jaguar
... Jaguar Panthera onca The largest cat of the Americas, the jaguar is a formidable beast. ... Jaguar coloration and size depends on habitat type. ...
http://www.bbc.co.uk/nature/wildfacts/factfiles/18.shtml (81)

☐ Jaguar (Panthera onca)
Jaguar Order Carnivora : Family Felidae : Panthera onca Linnaeus. Description. ... The jaguar is listed as "endangered" by the US Fish and Wildlife Service. Habits. ...
http://www.nsrl.ttu.edu/tmot1/pantonca.htm (82)

☐ CSG Species Accounts: Jaguar (Panthera onca)
Other Names jaguar (French) jaguar (German) tigre, tigre real, yaguar, jaguar, jaguarete (Spanish) onça, onça pintada, onça canguçu (Brazil) tig marqué

Search engine found 0 matching documents.

☐ World Trade Center - Minoru Yamasaki - Great Buildings Online
World Trade Center by Minoru Yamasaki architect, at New York, New York, 1966 to 1977. Demolished by ... Buildings Online. World Trade Center. ...
http://www.greatbuildings.com/buildings/World_Trade_Center.html (1)

☐ http://www.helping.org/wtc/ny/nystate.htm (2)

☐ Collapse of the World Trade Center
... Typical Floor Plan of the World Trade Center: ... The structural integrity of the World Trade Center depends on the closely spaced columns around the perimeter. ...
http://www.civil.usyd.edu.au/wtc.htm (3)

☐ WTCA On-Line
... Guests | My WTCA OnLine | World Trade library | | WTCA OnLine Catalog | WTC History ... Us | About WTCA | Board of Directors | WTC News Center | WTCA Newsletter ...
http://www.wtca.org/ (4)

☐ The WTC Memorial
What do you think would be an appropriate memorial to the World Trade Center victims at the WTC location? A new WTC, with new twin towers. Scaled-down offices. ...
http://www.thewtcmemorial.com/ (5)

☐ World Trade Center
The World Trade Center Height: 1,368 and 1,362 feet (417 and 415 meters) Owners: Port Authority of New York and New Jersey Architect: Minoru Yamasaki, Emery ...
http://www.skyscraper.org/tallest/t_wtc.htm (6)

☐ BUILDING BIG: Databank: World Trade Center
... World Trade Center Choose another wonder, World Trade Center. Click photo for larger image. ... World Trade Center, Click photo for larger image. ...
http://www.pbs.org/wgbh/buildingbig/wonder/structure/world_trade.html (7)

☐ :::: EARTH PLAZA ::::
http://www.worldtradecenter.com/ (8)

☐ Dubai World Trade Centre Online
... WTCA. Copyright © 1998, Dubai World Trade Centre (LLC), All Rights Reserved Tel: (+971 4) 3321000, Fax: (+971 4) 3312173, PO.Box: 9292 Email: info@dwtc.com.
http://www.dwtc.com/ (9)

☐ Ny.com: Statement Regarding WTC Attack
Please see the WTC information page for useful phone numbers, websites, and other information.

Search engine found 0 matching documents.

Figure 4. Computer desktop during the test (gray window is the task window). Picture A shows keyword context UI condition and picture B reference UI condition.

The experiment software collected an event log. The event log contained time stamped data about the actions the participants carried out. Such actions included result selections, keyword context selections, result page selections, etc. Time and accuracy measurements are based on this log data. Due to the features of the current graphical user interface environments, the timing data does not have an extremely high resolution, but the achieved, about ½ a second, resolution is enough for our purpose.

Two sets of nine search tasks were constructed such that they would be equally demanding. Each task required participants to collect links to relevant results pages. The task descriptions contained at least two central concepts, like “Find pictures (1) of Mount Pinatubo (2)”.

For each task there was a predefined query and the participants were not allowed to make or reformulate the queries themselves. This restriction was necessary to eliminate the variance caused by the different search skills of the participants. This increases the internal validity of the experiment by making the measurements more robust. In addition, the exclusion of the search formulation phase from the test does not compromise the external validity of test too much, because our contribution (and the main interest in the study) is in the result evaluation phase. The same approach for the same reasons was employed previously by Dumais *et al.* (2001) and Käki and Aula (2004).

The search results were accessed off-line during the experiment. Each predefined query was executed prior to the experiment and the 150 first results were saved in the local hard disk for fast and equal access. Using a real search engine on-line would have caused unpredictable differences between individual queries due to search engine load, network latencies and the changing content of the search engine database. The actual searches were executed with Google search engine through Google Web API.

4.4 Procedure

The experiments were organized in a usability lab during a week. Each participant was assigned a one hour time slot and one experiment session lasted about forty (40) minutes. The goal of the experiment, the procedure and the role of the participant were explained in the beginning of the test. The features of the user interfaces were explained during practice tasks, which took place in the beginning (for the first user interface) and in the middle of the session (for the second UI). During the practice tasks, the participants were allowed to try the user interfaces on their own and any questions concerning them were answered. When the participants executed the actual tasks, they were alone in the room.

The experiment contained two blocks of tasks, 9 tasks each. The blocks were completed with different user interfaces. After each block, the participant filled a questionnaire about the user interface. The questionnaire was identical for both user interfaces (both blocks). In the end, a questionnaire for comparing the user interfaces to each other was presented and demographic information was elicited. All the questionnaires were computerized and displayed in the task window.

The time reserved for each task was limited to one minute and the timing was automatic. Participants were instructed to first read the task description and then push the ‘Start’ button in the task window. When the ‘Start’ button was pressed, the search interface was immediately shown and the user could proceed with the task. The button press also started the timer. Between the tasks, the search window was visible, but its contents were hidden.

The participants could complete the task before one minute time limit by pressing ‘Finished’ button, which stopped the timer. If one minute time limit was reached, the contents of the search window were automatically hidden, the timer was stopped, and the participant was given the next task. The time between ‘Start’ button press and time-out or

‘Done’ button press constitutes the task time. The participants were informed about the timing scheme prior to the experiment.

The actual task instruction for the participants was to “collect as many relevant results for the given task as fast as you can”. The two competing goals being fast and precise attempted to simulate real world behavior. Collecting the results was done by checking the checkboxes added beside each result item (Figure 4). Participants were told that checking a result would correspond to opening it in the real situation, but during the test they could not open the documents. The participants were encouraged to follow their personal habits while performing the tasks.

5 RESULTS

5.1 Speed Measures

Simple task duration measurements give somewhat biased results in our setup where the total time for each task was limited to one minute. The setup causes a consistent ceiling effect as the allocated time is short for the tasks. As a result, the average task completion time is practically the same for both conditions: 56.3 s. for the keyword context UI and 57.2 for the reference UI with standard deviations (*sd*) of 5.0 and 4.7 respectively. Repeated measures analysis of variance (ANOVA) gives F(1, 35) = 2.36, *ns*.

Another raw speed measure is the number of results the participants collect. With the keyword context UI users collected, on average, 5.1 results (*sd* = 1.9) and with the reference UI 4.7 results (*sd* = 1.6). The difference is statistically significant: F(1, 35) = 4.34, $p < .05$.

These raw speed measures may, however, be misleading. The participants’ speed improvement could be due to poor accuracy and thus we focus next on the relevance of the results.

5.2 Accuracy Measures

The accuracy measurements are based on the relevance judgments of the results used in the experiment. All the 150 results for each of the 18 tasks (2700 results in total) were judged by the experimenter. The used relevance rating has three values: *relevant*, *related*, and *irrelevant*.

The rating is based on the existence of the central concepts of the task in the result summaries (note that the actual documents were not used in ranking because we did not want to measure the success of the summaries). Each task description had multiple (at least two) concepts in them and the relevant results were required to contain all of them in some form (the actual wording could vary). Related results were required to refer to the primary concept. Irrelevant results could contain the same words, but their meaning was different. For example, if the task was to find information about space shuttle Challenger accident, a relevant result would mention the Challenger disaster, a related one would be about space shuttle Challenger and an irrelevant could refer to Challenger learning center.

To capture the participants’ performance in terms of accuracy (rather than the performance of the retrieval engine), we use modified *recall* and *precision* measures. Our recall measure states the proportion of the relevant results the participants were able to find from all relevant results in the result set. Note that recall usually refers to the proportion of relevant results in the result set in relation to the whole database. The recall for the keyword context UI is 16% (*sd* = 5) and for the reference UI 12% (*sd* = 3) as seen in Figure 5. The difference is statistically significant as ANOVA gives F(1, 35) = 24.66, p < .01.

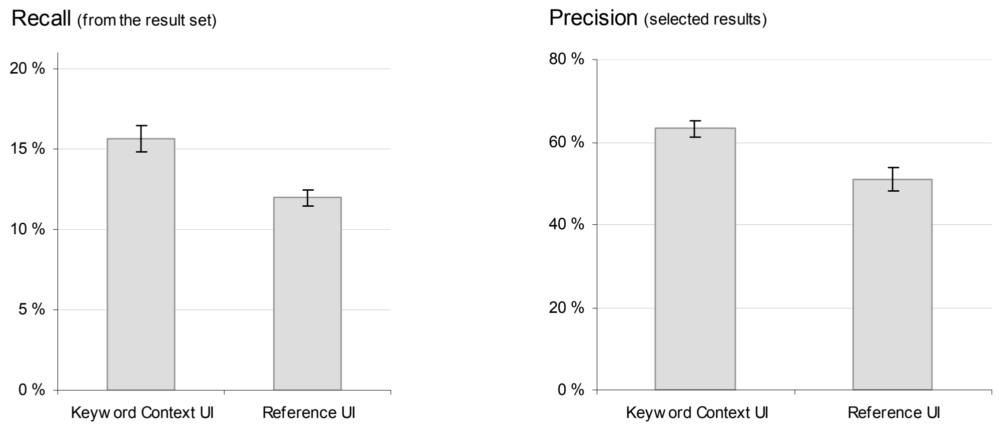


Figure 5. Recall (left) and precision (right) measures according to the condition, error bars represent standard error of mean (SEM)

Our precision measure is analogous to the recall measure and it describes proportion of relevant results within the selected results. This measure shows a difference to the same direction as the recall measure (Figure 5, right). With the keyword context UI the participants' precision was 63% (*sd* = 13) whereas with the reference UI it was 51% (*sd* = 17). Again, the difference is significant, F(1, 35) = 8.94, p < .01.

In addition to these, we analyzed the overall accuracy with qualified speed measure (Käki, 2004). Qualified speed states how fast the users find results with a given rating. The measurements are in Figure 6. The results show that the keyword context UI yields in a faster acquisition of relevant results (3.4 (*sd* = 1.4) vs. 2.4 (*sd* = 9.9) relevant results per minute, F(1, 35) = 11.04, p < .01). In addition, we see that the participants collected less irrelevant results with the keyword context UI. The speeds are 0.8 (*sd* = 0.4) and 1.0 (*sd* = 0.7) and ANOVA indicates statistical significance F(1, 35) = 4.14, p < .05.

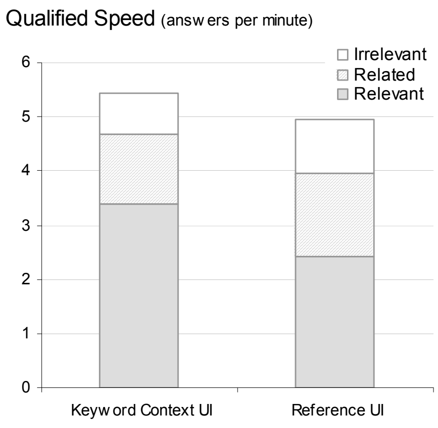


Figure 6. Qualified speed according to the condition

5.3 Immediate Success Measures

In addition to the overall accuracy of the selections, the success rate in the beginning of the selection process is especially important in the web environment. It is typical that the users settle for the first good enough answer they can find and thus select only one or two links in the result listing (Spink, Jansen, Wolfram, & Saracevic, 2002). In order to measure success in such behavior, we use the immediate accuracy measure (Käki, 2004) that states the proportion of the cases in which the user has found at least one relevant answer by the n^{th} result selection.

The immediate accuracy measurements are shown in the Figure 7. There is a difference in favor of the keyword context UI right from the first result selection. However, a statistical analysis shows that the difference in the first selection is not significant (F(1, 32) = 2.70, *ns*.), but from the second selection onwards it is (F(1, 32) ≥ 7.92, p < .01). Note also that the success rate of the reference UI remains below that of the keyword context UI also in the end. This means that there are more cases where not even one single relevant result is found for a given task with the reference UI.

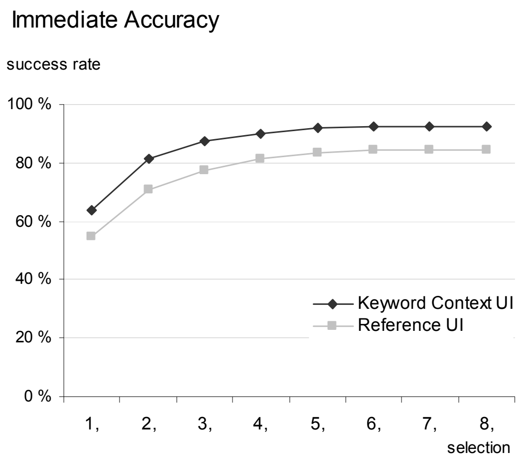


Figure 7. Immediate accuracy by the condition

5.4 Subjective Measures

In order to get insight into the participants' subjective opinions about the tested user interfaces we used three short questionnaires. After each block of test tasks the users were asked their opinions of the user interface just used. The set of questions in these two questionnaires (A and B) were identical. In the end of the test the two user interfaces were put to compete with each other in a questionnaire where their attributes were compared (comparison).

Identical questionnaires A and B contained eight claims and the users were required to respond to them with a six-point scale from agree to disagree.

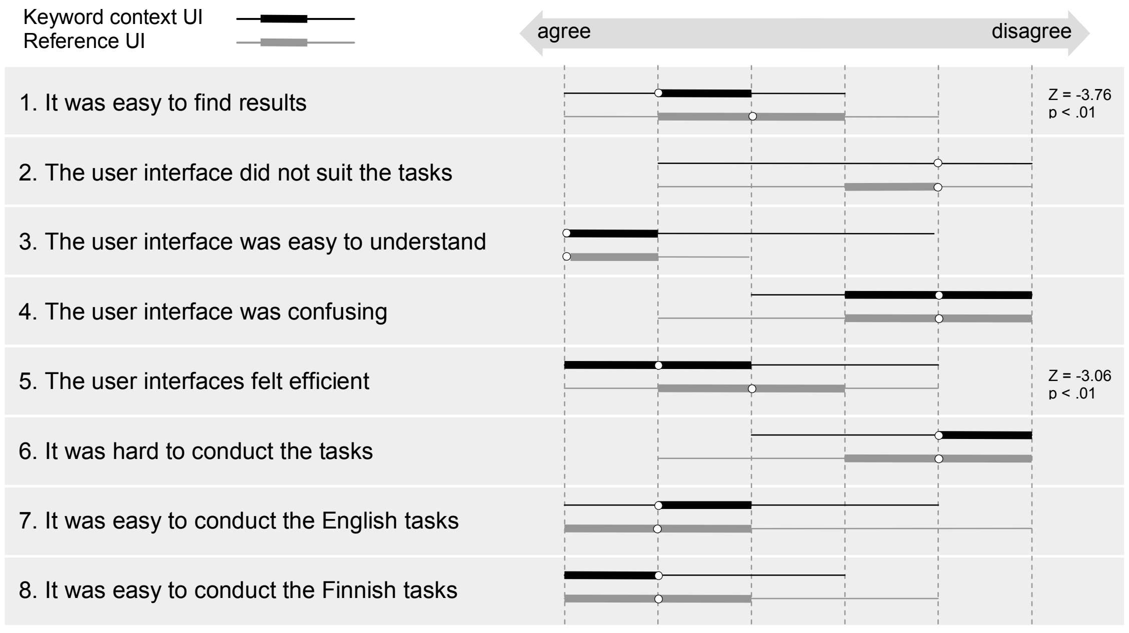


Figure 8. The claims and the results (box plots, thin line = min to max, thick line = 50% of observations, dot = median) of the questionnaires presented after each condition.

The participants rated the two user interfaces surprisingly similarly (Figure 8). Only in claims number one and five the median values of the responses differed from each other. In both cases the difference is in favor of the keyword context UI as it felt easier to find results and more efficient. The statistical significance was tested with Wilcoxon matched-pairs signed-ranks test (values of the statistics are listed in Figure 8). Answers to other claims indicate positive attitudes towards both user interfaces.

The comparison questionnaire consisted of five claims that were also answered using a six-point scale. The claims were consistent with the claims in the questionnaires A and B to introduce redundancy for increasing the reliability of the measurements.

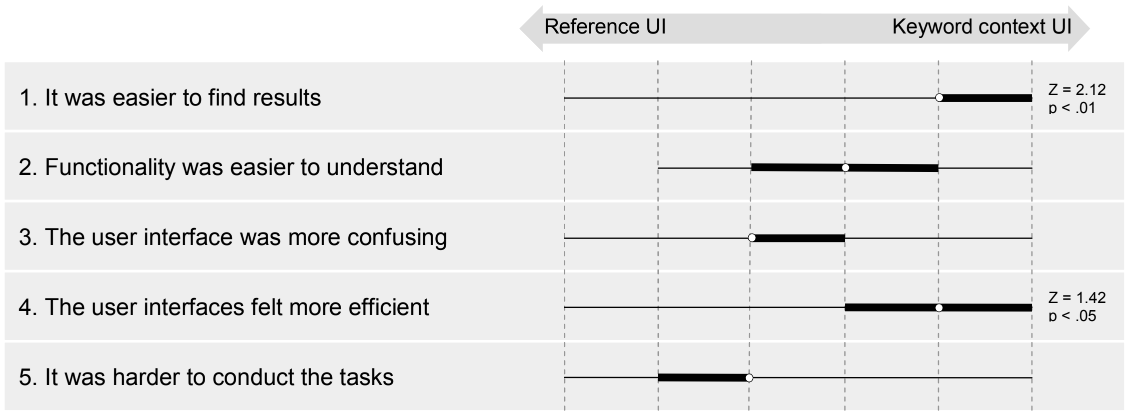


Figure 9. The claims and the results (box plots, thin line = min to max, thick line = 50% of observations, dot = median) of the questionnaires presented at the end of the test.

The results of the comparison questionnaire (Figure 9) are in line with the results of the previous questionnaires (compare to Figure 8). The claims about ease of finding results (1) and the feel of efficiency (4) show statistically significant differences. The differences are again in the favor of the keyword context UI. Statistical significance of these results was tested with one-sample Kolmogorov-Smirnov test and the actual values are seen in Figure 9 where appropriate.

6 DISCUSSION AND CONCLUSIONS

This paper introduced a new user interface based on keyword context index (*f*KWIC) for making the access of the web search results more effective. The solution is based on the old concept of keyword-in-context (KWIC) index. We implemented the idea for the modern web searches by introducing an algorithm that extracts the most frequent keyword contexts from the result items. The associated user interface allows the users to filter the result listing with the index.

To address the research question if the proposed system is beneficial for the users, we conducted a controlled experiment with 36 participants. The experiment compared the keyword context index user interface to the (*de facto*) standard ranked result list approach with a within subjects design.

The results of the study show that the proposed *f*KWIC user interface enhances the performance of a web searcher. The participants' speed of finding relevant results was about 29% higher with the proposed user interface. In addition, the precision of the selected results was significantly higher (63% vs. 51%) meaning that the participants collected less irrelevant results. They also found a greater proportion of the relevant results (within the results set), as evidenced by the higher recall measure (16% vs. 12%). Although the keyword context user interface is expected to be most useful in cases where multiple results are sought for (e.g. undirected search), its immediate success of finding relevant results with as few result selections as possible is better.

In respect to subjective opinions about the system, we found positive attitudes towards the *f*KWIC system. In particular, the participants felt that it was easier to find the results with the keyword context index and it also felt more effective.

In conclusion, the results of the study indicate that the keyword context index is a successful and feasible way of enhancing the access to the web search results. Objective measures evidence its efficiency and subjective attitudes favor it.

7 Future Work

Our line of research on this topic has evolved from the development of search user interface measures to experiments and longitudinal studies with the clustering approach. The current study is a direct continuation looking into an alternate approach. In the future, we will take a look at the properties of the two approaches. This will include a detailed description of the algorithms, their possible refinements, their applicability in various use scenarios and further performance measurements. We are also planning to add natural language descriptions of the search queries to the user interface to give more understandable feedback of the query interpretation.

8 Acknowledgements

This work was supported by the Graduate School in User-Centered Information Technology (UCIT) and the Academy of Finland. I would like to thank Kari-Jouko Räihä, Anne Aula, and Tomi Heimonen for insights and comments that contributed to this work.

9 References

Anick, P. (2003). Using Terminological Feedback for Web Search Refinement - A Log-based Study. *Proceedings of ACM SIGIR'03 (Toronto, Canada),* ACM Press.

Brin, S. & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems,* Vol. 30, No. 1-7, 107-117.

Chen, H. & Dumais, S. (2000). Bringing Order to the Web: Automatically Categorizing Search Results. *Proceedings of CHI'2000 (The Hague, Neatherlands)*, ACM Press, 145-152.

Cutting, D., Karger, D., Pedersen, J., & Tukey, J. (1992). Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *Proceedings of SIGIR 1992 (Copenhagen, Denmark)*, ACM Press 1992, 318-329.

Dumais, S. & Chen, H. (2000). Hierarchical Classification of Web Content. *Proceedings of SIGIR 2000 (Athens, Greece)*. ACM Press, 256-263.

Dumais, S., Cutrell, E., & Chen, H. (2001). Optimizing Search by Showing Results in Context. *Proceedings of CHI'2001 (Seattle, USA)*. ACM Press, 277-284.

Ferragina, P., & Gulli, A. (2005). A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. *Proceedings of WWW 2005 (Chiba, Japan)*.

Google Search Engine. http://www.google.com

Hearst, M. & Pedersen, J. (1996). Reexamining the Cluster Hypohesis: Scatter/Gather on Retrieval Results. *Proceedings of ACM SIGIR'96 (Zürich, Switzerland)*, ACM Press 1996.

Jardine, N. & van Rijsbergen, C. J. (1971). The Use of Hierarchic Clustering in Information Retrieval. *Information Storage and Retrieval*, Vol. 7, Issue 5, 217-240.

Kartoo Search Engine. http://www.kartoo.com

Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., & Krishnapuram, R. (2004). A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. *Proceedings of WWW 2004 (New York, USA)*, ACM Press, 658-665.

Käki, M. (2004). Proportional Search Interface Usability Measures. *Proceedings of the NordiCHI 2004 (Tampere, Finland)*, ACM Press, 365-372.

Käki, M. (2005). Findex: Search Result Categories Help Users when Document Ranking Fails. *Proceedings of the CHI 2005 (Portland, USA)*, ACM Press, 131-140.

Käki, M., & Aula, A. (2005). Findex: Improving Search Result Use through Automatic Filtering Categories. *Interacting with Computers*, Vol. 17, Issue 2, 187-206.

Nielsen, J. (2004) When Search Engines Become Answer Engines. At http://www.useit.com/alertbox/20040816.html

Paynter, G., Witten, I, Cunningham, S, & Buchanan, G. (2000). Scalable Browsing for Large Collections: a Case Study. *Proceedings of the ACM Conference on Digital Libraries 2000 (San Antonio, USA)*. ACM Press, 215-223.

Pratt, W. & Fagan, L. (2000). The Usefulness of Dynamically Categorizing Search Results. *Journal of the American Medical Informatics Association.* Vol. 7, No. 6, 2000, 605-617.

Salton, G. (1989). *Automatic Text Processing: The Transformations, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Spink, A., Jansen, B., Wolfram, D., & Saracevic, T. (2002). From E-Sex to E-Commerce: Web Search Changes. *IEEE Computer,* Vol. 35, No. 3. IEEE Computer Society, 107-109.

Tombros, A. & Sanderson, M. (1998). Advantages of Query Biased Summaries in Information Retrieval. *Proceedings of the 19th International SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98).* ACM Press, 2-10.

Vivísimo Search Engine. http://www.vivisimo.com

Zamir, O. & Etzioni, O. (1999). Grouper: A Dynamic Clustering Interface to Web Search Results. *Proceedings of the 8th International World Wide Web Conference WWW8 (Toronto, Canada)*. Elsevier Science.

Zamir, O. & Etzioni, O. (1998). Web Document Clustering: A Feasibility Demonstration. *Proceedings of the 19th International SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98).* ACM Press, 46-54.

Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., & Ma, J. (2004). Learning to Cluster Web Search Results. *Proceedings of the ACM SIGIR'04 (Sheffield, UK).* ACM Press, 210-217.