関連度の可視的ズーミングによるウェブ検索支援

劉　健全† 陳　漢雄† 古瀬　一隆† 大保　信夫†

† 筑波大学大学院システム情報工学研究科　〒 305–8577 茨城県つくば市天王台 1–1–1
E-mail: †{ljq,chx,furuse,ohbo}@dblab.is.tsukuba.ac.jp

あらまし　ウェブ検索は重要な情報検索技術であり、Google や Yahoo!などの検索エンジンがオンラインサービスとして広く応用されている。しかし、それらは伝統的なインターフェースしか使わなく、可視化検索支援を考えていない。本研究では関連度の可視的ズーミングによるウェブ検索支援手法を提案し、オンラインでユーザに検索キーワードの関連語を提供する。また、関連語の選択・再検索がより簡単にできる。ユーザインターフェースも提供する。
キーワード　情報検索、関連語、ズーミング、可視化、インターフェース

Supporting Web Search by Using Zoomable Interface to Present Relativity

Jianquan LIU†, Hanxiong CHEN†, Kazutaka FURUSE†, and Nobuo OHBO†

† Dept. of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba
Tennodai 1–1–1, Tsukuba-shi, Ibaraki-ken, 305–8577 Japan
E-mail: †{ljq,chx,furuse,ohbo}@dblab.is.tsukuba.ac.jp

Abstract　Web search is important in information retrieval applications, which is widely supported as online service by some well known search engines, such as Google and Yahoo!. However they are only running following the traditional interface which is not visualizable. To focus on this weak point, we proposed a visualized Web search supporting method and implemented a nice interface to suggest user related words to search keywords.
Key words　information retrieval, related words, zooming, visualization, interface

1. Introduction

Web search is becoming more and more important with the rapidly developing Internet. There have been many search engines providing web search service to people, such as the well known Google, Yahoo, MSN and so on. In most of search engines, user can only reach a ranked list of web pages order by ranking algorithm such as Google using PageRank [4]. However, the result set often contains a large number of pages, which make user confused to find out target page. In general, people do not know whether the following several ten pages can meet their requirement, thus they have to click “next page” to check and confirm. In this case, most of search engines will become powerless.

As a result, many researches have been focusing on improvement based on the original searching results, or studying for related words finding model to refine the specification of input keywords. For example, [3], [9] and [10] presented improvement by ranking or re-ranking method. [7] and [5] gave a finding model and a collection method of related words.

In spite of such many researches abroad concerned with improvement algorithms and finding related words, they have not meet our convenient and fast search requirement yet. In this paper, we are interested in zoomable interface, and present the implementation of our Web Search Supporting System (ZmSearch(注1),(注2)). It can refine the search results fetched from Yahoo!JAPAN WebSearch API(注3), and at the same time provide a candidate set of related words with input keywords via an efficient zoomable interface attaching in the window.

Due to our ZmSearch system, it is effective and convenient to visualize search results, and also significant to suggest user search again by selecting some related words, that will be proved by its performance and our evaluation.

The remainder of the paper is organized as follows. In section 2, we discuss some related works and referred applications. Section 3 presents our proposed approach, and then the system structure and its implementation come to section 4 and section 5. Following these, we

（注1）：http://zmsearch.dblab.is.tsukuba.ac.jp/
（注2）：http://www.dblab.is.tsukuba.ac.jp/~ljq/zmsearch/
（注3）：http://developer.yahoo.co.jp/search/

will show its running performance and give the significant evaluation in section 6. Finally, it comes to the conclusion together with future work in section 7.

2. Related work

2.1 Related Terms

In most search engines, such as Google and Yahoo!, only frequently co-searched words can be suggested by analyzing a large number of user searching histories. So there are many studies which have focused on related terms(or named as related words) for information retrieval. Generally, the studies about related words can be divided into two kinds of measurement, co-occurrence and semantic similarity of words. For instance, an early work in [8] , used word co-occurrence combining with traditional word frequency ranking to calculate relevance ranking of documents. On the other hand, Bollegala [2] et al recently proposed a robust semantic similarity measure that uses the information available on the Web to measure similarity between words or entities.

2.2 Zoomable Interface

Recently, studies for zoomable interface have been attracting much attention to apply it for information search. Taking an example, Araki [1] proposed a new browsing method for information in Web environments by relating a continuous zooming operation to a search result window.

Besides, there was a study for comparing textual and zoomable user interfaces published in [6]. It attempted to provide a controlled comparison among three interfaces, Grokker(注4), Grokker Text and Vivisimo(注5), and concluded a better understanding of the potential that a ZUI(Zoomable User Interface) based visualization may offer clustered search results. As the result of ZUI's better understandable potential, we considered to apply ZUI in our Web search supporting system.

3. Our approach

In order to overcome the problems arised in currently existing search engines, such as not providing enough related words in high precision, non-zoomable user interface and so on, we proposed our related words finding method and a zoomable user interface for operating search results and visualizing related words list. Furthermore, with aggregating these beneficial factors, we implemented a Web search supporting system. We give its running performance and experimental evaluation as well.

Definitions and notations are summaried in Table 1. As our approach is to refine the results which are returned by a general search engine, we are using Yahoo! WebSearch API in our implementation. Because it is an online Web search supporting system, we cannot use the formal definitions to calculate TF, DF, and $|D|$ for the large-scale results. Exactly, it is not impossible but very difficult and meaningless if we do so, because it absolutely makes processing very slow and hard to suffer with. For example, the Yahoo! Web search engine always returns about 1 billion results for the query term "NBA". Accordingly, we defined our notions in Table 1, and treat the top 100 returned results as our analysis object, a partial sample of original results.

Especially, in the Web search field, a unit document is a page, and meanwhile, a term means a word in the page. In our definition, $pTF(w_i)$ denotes the partial term frequency of w_i in candidate set after stopwords and stemming processing. Here, because of not the whole results returned by Yahoo! but only the top 100, we take the meaning of partial into our definition. Consequently, $pDF(w_i)$ means the partial document frequency of w_i, and we import a symbol $TR(w_i)$ to present the relativity value of w_i.

And then our related words finding method, zoomable interface design, and the system structure will come to the following sections.

pTF	partial Term Frequency
pDF	partial Document Frequency
TR	Term Relativity
w_i	ith Word(term) in candidate set
d_j	jth document in returned result set
$\|D\|$	total number of original returned results
iTF_j	internal Term Frequency in d_j

Table 1 Definitions

4. System structure

In this section, the system structure is organized by logical and functional structures in the following subsections. The logical structure describes the general view of systemic processing in different phase. On the other hand, the functional structure gives the basic function design by using zooming interface.

4.1 Logical structure

Along the processing flows in Figure 1, we can track three processing engines, Ajax engine, search engine and analysis engine. At first, user's search request comes to the Ajax engine, which posts the request to the search engine running in background on the Web server, and then the search engine fetches the top 100 pages(documents) from Yahoo!JAPAN by its WebSearch API. It also immediately transfers these pages to analysis engine continuing to do text analyzing and related words computation. After that, it outputs the top 20 related words holding their $TR(w_i)$ value as response data to Ajax engine. The Ajax engine will respond to user while dealing with zooming.

4.2 Functional structure

Attempting to meet our proposed interface requirements, we should give the functional structure design in details. As shown in Figure 2, there are three visual areas composing the whole functional interfaces. The main screen is occuplied by the ranking list at the central left side. At the top, it is a slider bar for jumping to ith page of the 100 results in ranking list. You should focus on the right side

(注4)：http://www.groxis.com/

(注5)：http://www.vivisimo.com/

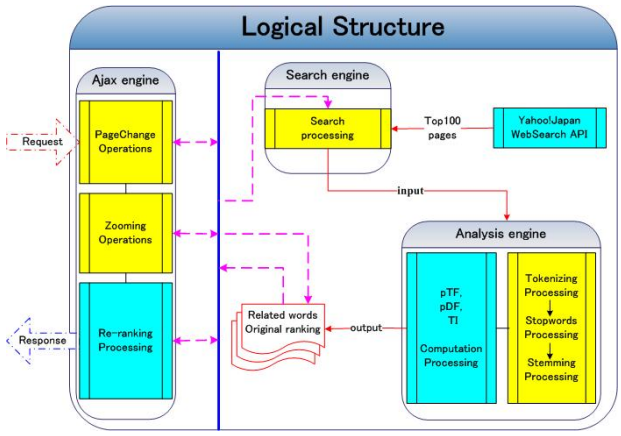

Figure 1

where the zoomable interface is embedding, with attached by a related words list. The list can be changed dynamically according to the moving position of zooming bar, which is connected to the TR value.

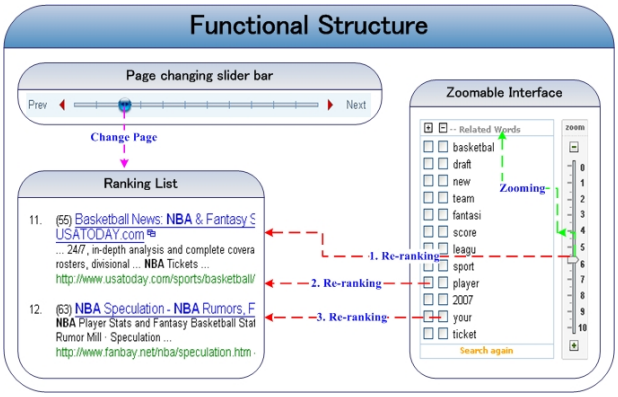

Figure 2

5. System implementation

The implementation of our ZmSearch system consists of two phases as follows.

- Related words analysis
- Zoomable interface implementation

The processing programs for related words analysis are written in PHP language, and we use Ajax engine to deal with the operation of zoomable interface. We also fetch the top 100 search results by the way of Yahoo!JAPAN WebSearch API. Our system implementation consists of opproximately 3500 lines of commented PHP and JavaScript code. In the following subsections, details in each phase of implementation will be given.

5.1 Related words analysis

In the related words analysis processing phase, we defined an analysis procedure $wordAnalysis$ in Procedure 1. It collects the title and summary text of documents D_{top100} as input data, and outputs the top 20 most related words attaching with their TR value for the next processing phase.

From line 1 to line 7 in Procedure 1, the text strings in title and summary of documents are preprocessed by order of tokenization, stopwords elimination, stemming and splitting into a meaning words set. In our implementation, we used a big stop list(注6) containing 900 stopwords, which is provided by Dr. Drott's information retrieval resource. The stemming processing is cited from Porter Stemming Algorithms(注7).

Procedure 1 wordAnalysis(D_{top100})

Begin

1: **foreach** $d_j \in D_{top100}$
2: $sText \leftarrow d_j.sTitle + d_j.sSummary$
3: $sText \leftarrow$ Do tokenization in $sText$
4: $sText \leftarrow$ Eliminate stopwords from $sText$
5: $sText \leftarrow$ Deal with stemming in $sText$
6: $Set_{words} \leftarrow$ Split from $sText$
7: **end foreach**
8: **foreach** $w_i \in Set_{words}$
9: $pTF(w_i) \leftarrow$ Count pTF of w_i
10: $pDF(w_i) \leftarrow$ Count pDF of w_i
11: $TR(w_i) \leftarrow pTF(w_i) \times pDF(w_i)$
12: **end foreach**
13: Output top 20 $TR(w_i)$ and w_i

End

For offering the top 20 most related words to next phase, we consider the most important term as the most related word. Moreover as the reason that it is impossible to fetch all the hundreds of thousands of original search results, we cannot figure out the traditional TF-IDF method, which needs to count $|D|$, TF and DF of the whole documents. In stead of considering computing pTF and pDF, we can figure out the term relativity TR of w_i by the following equation, which is confirmable from line 8 to line 12 in Procedure 1.

$$TR(w_i) = pTF(w_i) \times pDF(w_i) \quad (1)$$

5.2 Zoomable interface

Besides related words analysis, another important idea is that we proposed a zoomable interface to control the visualization of related words list. It is the main idea to divide the zooming ruler into eleven levels from 0 to 10, and to bind each level connecting with the normalized the $TR(w_i)$ value of w_i in the whole related words list. Accordingly, changing the level of zooming bar effects on visualizing the related words dynamically.

The details of zooming functional structure, how to visualize the zooming operations and changing of related words list have been mentioned before.

5.3 Snapshot of system

This section gives a simple description of our ZmSearch's snapshot in Figure 3. Also it is a sample of search result about the keyword "NBA" that is obvious at the head of the page. In the area of page changing slider bar, the information message denotes the summary

(注6)：http://drott.cis.drexel.edu/retrieval.html
(注7)：http://tartarus.org/ martin/PorterStemmer/

of search result, such total returned number, running time, displaying page number and so on. The left main area is a screan holding 10 ranking elements per list page. The key zoomable interface and related words list are embedding at the right side. Lastly, at the bottom of related words list, the orange link “Search again” can be clicked to search again applying the checked words in the “+” column. The checked words are also emphasizing in different colors.

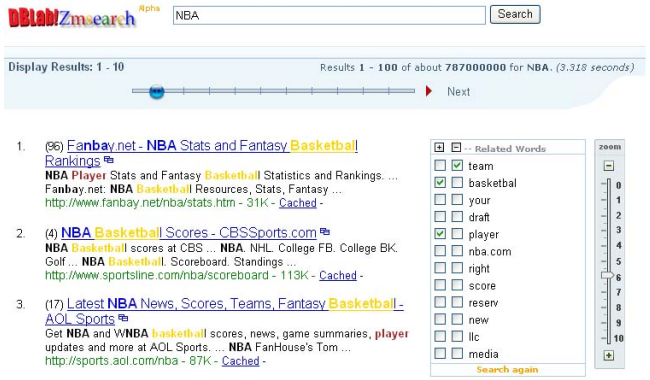

Figure 3 ZmSearch’s snapshot

To be honesty, the alpha version of our ZmSearch system has been published recently. Please track the following accessible URL. *http://zmsearch.dblab.is.tsukuba.ac.jp/* or *http://www.dblab.is.tsukuba.ac.jp/~ljq/zmsearch/* .

6. Evaluation

In this section we describe the evaluation we performed for the ZmSearch system. Section 6.1 describes our study, while section 6.2 discusses the precision of related words in our study. Section 6.3 evaluates the run time performance of our system.

6.1 Experimental setup

To compare the precision of related words provided in different approaches, we performed a user study. We chose 11 hot keywords as our search targets, which are gathered from Google Trends(注8) and Yahoo!Buzz(注9) . These 11 hot keywords are about three different categorizations, 4 keywords about programming language, 3 keywords about software technology and the rest 4 keywords about sports, which are as presented in Table 2.

Search keywords	Quantity
ajax, java, ruby, python	4
database, data mining, open source	3
NBA, ESPN, NHK, Miki Ando*	4

* She is a famous figure skater from Japan.

Table 2 Search keywords

After that, we applied these keywords to search the Web and gathered top 10 related words returned by three different systems, which are related search words service API(注10) (we call it Yahoo!API in the paper) provided by Yahoo!JAPAN, Rerank.jp(注11) and our ZmSearch. And then we put the answers in ascendant order with omission to make sure each word unique, We organized such a questionary sample as Table 3, where each related words list is separated according to 11 different search keywords. Combining 11 tables, we organized a complete questionary sheet in the end, which contains about 300 related words.

Search keyword	python	**Judgement**
Related words	document	
	download	
	eclipse	
	extens	
	interpret	
	monty	
	news	
	programming	
	scripts	
	software	
	syntax	
	that	
	tutorials	
	using	
	version	
	windows	
	with	
Memo: 1. If you consider the word is related, please mark a ◯. 2. If you consider the word is not related, please mark a ×. 3. If you cannot make a decision, please mark a △.		

Table 3 Questionary Sample

We delivered our questionary sheets to 21 volunteers for rating as soon as we finished the preparation. Our volunteers included 1 professor and 20 students, of which there are 2 female and 19 male. There are all computer science professionals (mostly our colleagues at our Database Research Laboratory).

The volunteers were instructed as follows:

> We want to measure how well the related words perform, which were gathered from three different search system and combined together in ascendant order. Please give your judgement according to the memo in our questionary tables after yourself consideration, and please ignore the order without caring which system it came from. The questionary sheet may be finished within about 5 minutes.

6.2 User study results

From the answers we received, we evaluated *average precision* at top 10 related words for three different systems. Our statistic analysis is dealt as the following procedure.

（注8）：http://www.google.com/trends
（注9）：http://buzz.yahoo.com/
（注10）：http://developer.yahoo.co.jp/search/webunit/V1/webunitSearch.html
（注11）：http://rerank.jp/

Firstly, we converted the three kinds of symbol ($\bigcirc, \triangle, \times$) into three kinds of connected weight (1, 0.5, 0) in the returned questionary sheets, which makes all the judgements computable. $\bigcirc$-1 means that the volunteer considered the word as related to search keyword, oppositely $\times$-0 denotes not related. $\triangle$ has 50% probability that the word is related to search keyword, we assigned the weight 0.5 for computation.

Secondly, in allusion to each search keyword, we computed three average precision p_s by Equation 2, where s=1 denotes Yahoo!API, Rerank.jp as 2, and ZmSearch as 3.

$$p_s = \frac{\sum_{j=1}^{21} \frac{\sum_{i=1}^{10} weight_{s,j}(w_i)}{10}}{21}, \quad s = 1, 2, 3 \qquad (2)$$

Thirdly, we achieved numerical results as Table 4 after finishing the computation, as well as plotted a clearly comparable view as Figure 4

Keywords	Yahoo!API	Rerank.jp	ZmSearch
ajax	0.38	0.37	0.79
java	0.69	0.46	0.52
ruby	0.20	0.47	0.57
python	0.50	0.54	0.61
database	0.37	0.37	0.66
data mining	0.00	0.57	0.68
open source	0.00	0.47	0.75
NBA	0.42	0.65	0.81
ESPN	0.79	0.60	0.69
NHK	0.79	0.33	0.74
Miki Ando	0.00	0.67	0.79
Average	**0.38**	**0.50**	**0.69**

Table 4 Statistic result

According to the statistical result in Table 4, it is obvious that our ZmSearch system achieves an average precision of 0.69 at top 10 related words, which is 38% better than Rerank.jp system, moreover 81.6% better than Yahoo!API.

Figure 4 shows a visualized result by comparable curves as well. It predicates that our ZmSearch system performed absolutely better than Rerank.jp system for suggesting the top 10 related words to search keywords. However there are some outliers when comparing with Yahoo!API, such as seeking information about java, ESPN or NHK. Because Yahoo!API does not support multi-word search for related words, its precision leads to zero while searching “data mining”, “open source” and “Miki Ando”.

6.3 Online run-time performance

We also evaluate the online run-time performance of our ZmSearch system to confirm its efficiency. The experiment is performed on a Intel-based computer system running Linux. CPU is Intel(R) Xeon (TM) 2.80 GHz and the amount of main memory is 3.2GB. During our online run-time experimental testing, we repeated the same search task 12 times for each keywords in Table 2, and wrote down the running cost of each repeat. In allusion to each search keyword, we calculated its average running time with omitting the maximum

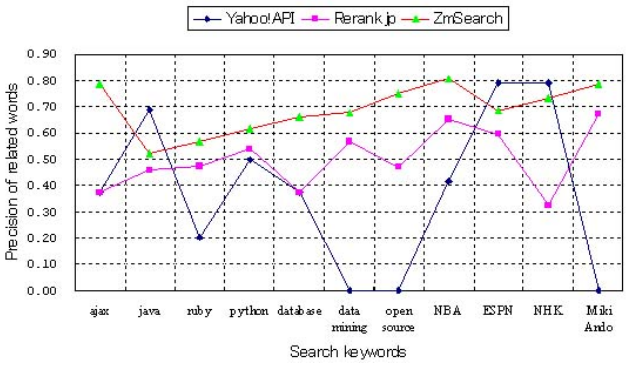

Figure 4 Precision at top 10

and minimal value in the 12 records. We plotted the result in Figure 5 and achieved a final average value of 2.03s as the online run-time performance which is presented by the straight line in Figure 5. The pink curve presents the real average run-time cost for each search keywords. To be honesty, the experimental result gives an acceptable online run-time performance implying that our ZmSearch system just costs about 2 seconds to do online processing for related words analysis. As a Web search assistant system, it is confident that user can suffer from this cost range.

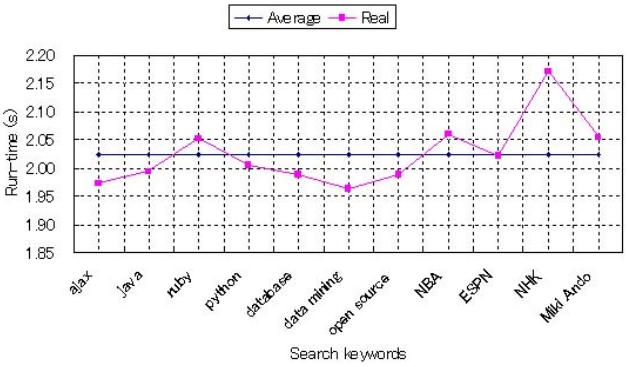

Figure 5 Run-time cost

7. Conclusion and future work

In this paper, we proposed our $pTF \times pDF$ method to compute relativity and suggest related words for search keyword. Moreover we implemented an online Web search assistant system using zoomable interface and released it as an alpha version. Besides, our ZmSearch system was proved effective and convenient to visualize the search result, significant to supply related words in a relatively higher precision at top 10, and its online run-time performance is so acceptable that it is suitable for actual application.

Although we achieved a nice conclusion, our related words finding method should be improved in future work, because they are a little simple and not enough to support complex computation. Or we could extend our idea of using visualized interface to present cluster data together with zooming related words. It’s further that we can try to visualize the results combining with re-ranking algorithm and attempt to apply semantic similarity to figure out related words and so on.

Acknowledgements

This research has been supported in part by MEXT (#19024006).

References

[1] Tadashi Araki, Hisashi Miyamori, Mitsuru Minakuchi, Zoran Stejic, and Katsumi Tanaka. An application of zooming cross-media for information search. Technical report, data engineering, The Institute of Electronics, Information and Communication Engineers, 2005. Vol.105(172), P65-70.

[2] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. In *the 16th International Conference on World Wide Web*, pages 757–766, 2007.

[3] Chen Ding and Chi-Hung Chi. A generalized site ranking model for web ir. In *IEEE/WIC International Conference on Web Intelligence*, 2003.

[4] Page Lawrence, Brin Sergey, Motwani Rajeev, and Winograd Terry. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998. P49-60.

[5] SHINGO OTSUKA, MASASHI TOYODA, and MASARU KITSUREGAWA. A study for related words finding method using global web access logs. *Transactions of Information Processing Society of Japan*, 46(SIG 8(TOD 26)):82–92, 2005.

[6] Walky Rivadeneira and Benjamin B. Bederson. A study of search result clustering interfaces: Comparing textual and zoomable user interfaces. Technical Reports HCIL-2003-36, Human-Computer Interaction Lab., University of Maryland, October 2003.

[7] SATOSHI SATO and YASUHIRO SASAKI. Automatic collection of related terms from the web. *IPSJ SIG Notes*, 2003(4):57–64, 20030120.

[8] Toru Takaki and Tsuyoshi Kitani. Relevance ranking of documents using query word co-occurrences. *Transactions of Information Processing Society of Japan*, 40(SIG 8(TOD 4)):74–84, 1999.

[9] Takehiro Yamamoto, Satoshi Nakamura, and Katsumi Tanaka. An editable browser for reranking web search results. In *The 3rd International Special Workshop on Databases for Next-Generation Researchers*, 2007.

[10] Takehiro Yamamoto, Satoshi Nakamura, and Katsumi Tanaka. Rerank-by-example: Efficient browsing of web search results. In *The 18th International Conference on Database and Expert Systems Applications*, pages 801–810, 2007.