

A Click-Search Interface for Web Browsing Using Cellular Phones

Daijiro Komaki[†], Kenji Ohnishi[†], Yuki Arase[†],
Takahiro Hara[†], Gen Hattori[‡], Shojiro Nishio[†]

[†]Dept of Multimedia Engineering,
Grad. Sch. of Information Science and Tech., Osaka Univ., JAPAN
{komaki.daijiro, ohnishi.kenji, arase.yuki, hara, nishio}@ist.osaka-u.ac.jp
[‡]KDDI R&D Laboratories Inc., JAPAN
gen@kddilabs.jp

ABSTRACT

Cellular phones are widely used to access the Web, and it has become common to search for information on the Web using cellular phones. However, since cellular phones only have numerical keys for inputting characters, users have to conduct laborious operations to input a keyword for Web search. In our previous work, we assumed information appliances equipped with pointing devices and proposed a Click-Search method that enables users to perform a keyword search simply by pointing and clicking the target word on a Web page. This approach might be effective for Web searches using cellular phones. However, since many cellular phones only have a direction pad as a pointing device, users still have to conduct laborious operations using the pad to move a pointer and point to a word. To solve this problem, we propose a Click-Search interface for cellular phone users that uses an expanding circle to select a word for search. Using this interface, users can specify a keyword simply by pushing a few buttons.

Categories and Subject Descriptors

H.4.3 [Information Systems]: Communications Applications—
Information browsers

General Terms

Design

Keywords

Cellular phone, Web browsing, Web search, Interface

1. INTRODUCTION

Because of the recent advances in a processing and communication facilities, cellular phones are widely used to access the Web. It has also become common to browse not

only Web pages designed for cellular phones but also those designed for desktop PCs. Web searches are essential for finding Web pages containing the information users need. For Web browsing using desktop PCs, it is common to use search engines such as Google and Yahoo!. Therefore, some cellular phone carriers have started offering searching services, e.g., Google, and it is expected that more and more users will use such searching services in the near future. However, since many cellular phones only have numerical keys for inputting characters, users have to conduct laborious operations to input keywords for Web searches.

In our previous work, we assumed information appliances equipped with pointing devices, such as a remote controller of the Nintendo Wii[12], touch panel, and mouse for PCs, and proposed a Click-Search[4] method to avoid laborious operations for inputting a search keyword. In this method, when a user points to and clicks a word on a Web page, a Web search using the word is automatically performed with a search engine, and the most relevant page is displayed as the search result on the user's screen. This approach might also be effective for Web searches using cellular phones. However, cellular phones only have a direction pad as a pointing device. Therefore, it is not effective to directly adopt this method, because users have to conduct laborious operations using the direction pad to move the pointer and point to a word.

In this paper, we propose an easy-to-use Click-Search interface for cellular phone users. To reduce users' laborious operations, it uses an expanding circle to specify a word for Web searching instead of directly pointing to the word. Specifically, when a user pushes a key, a circle appears on the screen and it keeps expanding as long as the user keeps pushing the key. The user releases the key when the targeted word is covered by the circle. Then, the system selects the words for search from the words inside the circle, considering the attributes of the words and their distance from the center of the circle. Using this interface, users can specify a keyword for search by pushing only a few buttons.

The remainder of this paper is organized as follows. Related works are described in section 2. In section 3, we describe the Click-Search method, which we proposed in our previous work. In section 4, we describe our proposed interface in detail. In section 5, we describe the results of a user experiment for evaluating the effectiveness of the proposed interface. We provide some discussion in section 6. Finally, we conclude the paper in section 7.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WICON '08 November 17-19, 2008, Maui, Hawaii, USA.
Copyright 2008 ICST 978-963-9799-36-3.

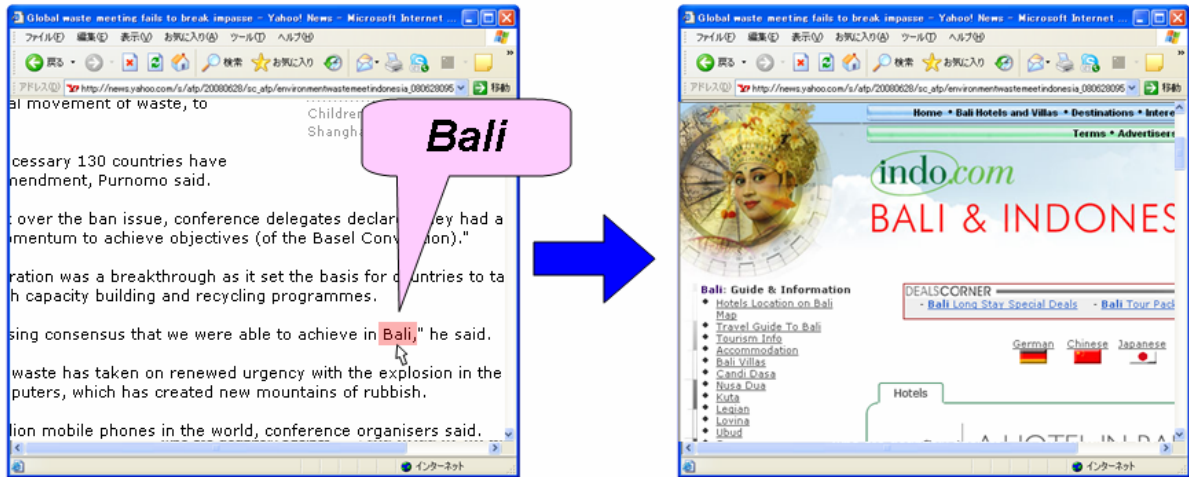


Figure 1: An example of executing the Click-Search method

2. RELATED WORKS

Since cellular phones have small screens and poor interfaces, it is inconvenient to browse Web pages designed for desktop PCs on them. Many studies have attempted to solve this problem.

ZoneZoom[8] divides an overview of a Web page displayed on the screen of a cellular phone into nine segments and assigns each segment to a numerical key. When a user pushes a numerical key, the corresponding segment is scaled up. Minimap [9] changes the widths of text paragraphs and scales down images and tables, while keeping as close as possible to the original layout of the Web pages. In [1], we proposed a Web browsing system that adaptively presents Web contents. This system first displays an overview of a page and a user selects a content of interest. Then the system automatically displays the content in the best way according to its characteristic in order to reduce laborious scrolling operations to browse the page. These conventional techniques improve Web page presentation on cellular phones. However, they do not address Web search using cellular phones.

SoftBank[10] provides a commercial search service for cellular phones called Yahoo! oneSearch[14]. It predicts what information the user needs based on his/her preferences, and displays it at the top of the search results. For example, when a user inputs a place name, the map of the place is displayed as the top search result. NTT docomo[6] and au[2] provide a Google search engine for their mobile portal sites. Users can find their target pages not only by searching the directories at the portal sites but also by querying the Web searching engine.

These services require inputting characters for specifying keywords to use the search facility. This is burdensome for cellular phone users. On the other hand, [11] proposed a search method for PDAs, which are equipped with a stylus and a touch screen. When a user traces and specifies a keyword with the stylus, the system presents a menu suggesting options for the user's search intention. This menu is automatically generated based on the words located around the keyword. However, tracing a keyword is still a burden task for cellular phone users.

3. CLICK-SEARCH METHOD

We propose a Click-Search interface for cellular phone users, which enables the users to specify a keyword without laborious operations. In this section, we describe the Click-Search method, which we proposed in [4] and which is the basis of the Click-Search interface proposed in this paper.

3.1 Outline

The Click-Search method is intended for information appliances equipped with a pointing device, such as a remote controller of the Nintendo Wii, touch panel, and mouse for PCs. When a user points and clicks a target word on a Web page, a Web search using the word as the keyword is automatically performed, and the result page is displayed on the user's screen. Figure 1 shows an example of executing the Click-Search method. In this example, a user is interested in the word "Bali" and clicks the word with the mouse, and a Web page related to Bali is presented. In the following, we describe the procedure of the Click-Search method.

1. Extracting n characters before and after the clicked position
The character nearest the clicked position is considered the center character, and n characters before and after the center character (in total, $2n$ characters) are extracted.
2. Extracting free-standing and compound words
Nouns are extracted from the $2n$ characters extracted in step 1. Consecutive nouns are considered to be a compound word.
3. Extracting "core" word and "neighboring" words
Among the nouns and compound words extracted in step 2, the nearest noun to the clicked position is considered the core word, and the others are considered neighboring words.
4. Calculating "importance"
The "importance" of each neighboring word is calculated. Here, a few neighboring words with high im-

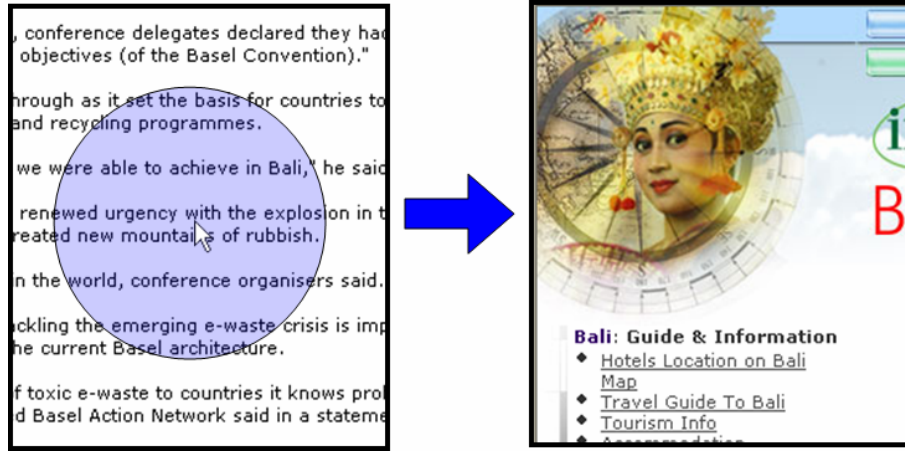


Figure 2: Click-Search interface

portance are combined with the core word in order to narrow down the search result. The details of calculating the importance are described in section 3.2.

5. Conducting Web searches
The core word is combined with the neighboring words with high importance, and a Web search is performed using the API provided by a search engine.
6. Presenting the resulting Web page
The top search result's Web page is displayed on the user's screen.

3.2 “Importance” of neighboring words

The “importance” of a neighboring word is defined as the degree to which it can narrow down the search result by being combined with the core word. The importance is calculated based on the importance of the word itself and the importance in relation to the core word, i.e., how the word is related to the core word. The importance of neighboring word w with core word w_a is expressed as $E(w, w_a)$ by the following equation:

$$E(w, w_a) = \alpha E_o(w) + (1 - \alpha) \cdot E_r(w|w_a). \quad (1)$$

Here, $E_o(w)$ is the importance of word w itself, and $E_r(w|w_a)$ is the importance of word w in relation to core word w_a .

$E_o(w)$ is calculated by the following equation:

$$E_o(w) = (F_m(w) + F_v(w) + 1) \cdot p(w). \quad (2)$$

Here, $F_m(w)$ is 1 if w is judged as a proper noun, and 0 otherwise. $F_v(w)$ is 1 if w contains a highlighting expression such as “()”, “< >”, or tag in HTML, and 0 otherwise. $p(w)$ is a statistical value calculated by $tf \cdot idf$ using documents collected from news archives.

$E_r(w|w_a)$ is calculated by the following equation:

$$E_r(w|w_a) = \frac{F_c(w, w_a)}{F_d(w, w_a)}. \quad (3)$$

Here, $F_c(w, w_a)$ is the degree of co-occurrence of w with core word w_a in the currently displayed Web page, and $F_d(w, w_a)$ is the distance between w and w_a on the page. They are

expressed by the following equations:

$$F_c(w, w_a) = \frac{freq(w|dist(w, w_a) < k)}{freq(w|w \in Doc)}, \quad (4)$$

$$F_d(w, w_a) = dist(w, w_a). \quad (5)$$

Here, $freq(w)$ is the number of word w in Doc , which is the currently displayed Web page. $dist(w, w_a)$ is the distance between w and w_a , which is defined as the number of characters between the two words. k is the window size to examine the co-occurrence of w and w_a .

After calculating the importance of each neighboring word, neighboring words with high importance are combined with the core word to conduct a multiple keyword search.

4. CLICK-SEARCH INTERFACE FOR CELLULAR PHONES

This section describes the proposed interface that enables the Click-Search method to be comfortably used on cellular phones. We focus particularly on how to determine the core word. Here, we assume a Web browser for cellular phones which presents Web pages in the same way as desktop PCs, and users browse pages by scrolling them up, down, right, and left using the direction pad. At the center of the screen, a pointer of the similar shape with the mouse cursor is displayed, i.e., the pointer is fixed at the center position.

As an approach to select a search keyword by simply using the direction pad and center action button, we use an expanding circle, which is like netting a keyword. Specifically, when a user pushes a specific key, a circle appears on the screen centered around the pointer, and the circle keeps expanding as long as the user keeps pushing the key. When the circumference of the circle reaches the edge of the screen, the circle starts shrinking until it gets back to its initial size. After that, it starts expanding again.

If the user is interested in a certain word on a Web page, he/she simply pushes the key and keeps pushing it until the circle includes the target word. After the user releases the key, the resulting Web page is displayed on the screen. Figure 2 shows an example of a search using the Click-Search interface. In this example, the user was interested in the word

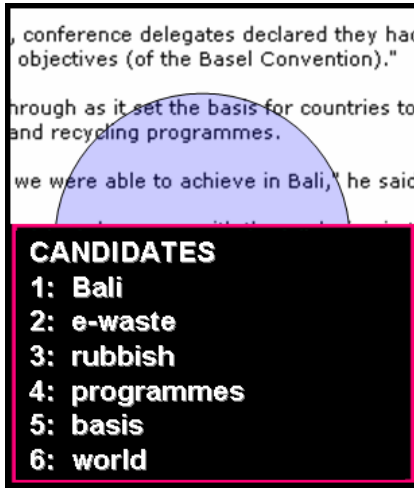


Figure 3: The Listing method

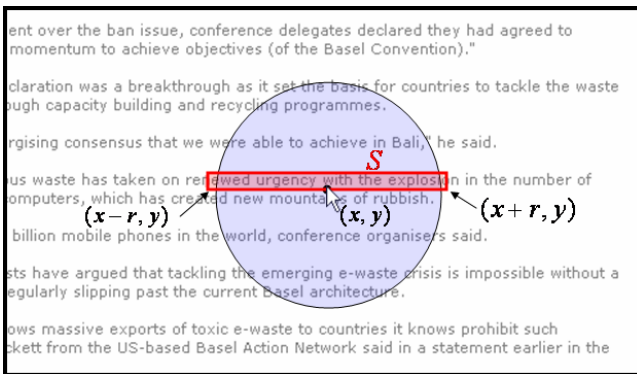


Figure 4: String extraction

“Bali” and expanded the circle until it covered the word. Then, the resulting Web page was presented.

In order to select a search keyword, i.e., the core word, from words contained in the circle, we designed two methods: “automatic selection” and “listing”. The automatic selection method selects just one word as the search keyword automatically. The listing method presents a list of candidate keywords, and users select one from the list using numerical keys (See Figure 3).

4.1 How to select a search keyword

In order to determine search keyword candidates, the system calculates the importance of each word. Here, we define the “importance” of a word as a measure representing how suitable the word is as a search keyword.

4.1.1 Extraction of search keyword

First, we describe how to extract words contained in the circle. Let us denote the coordinates of the circle center as (x, y) , and denote the radius as r .

First, the system extracts a partial string between the two characters located at the nearest positions to $(x - r, y)$ and $(x + r, y)$, respectively, which we call S (See Figure 4). This process is called “string extraction with center (x, y) and

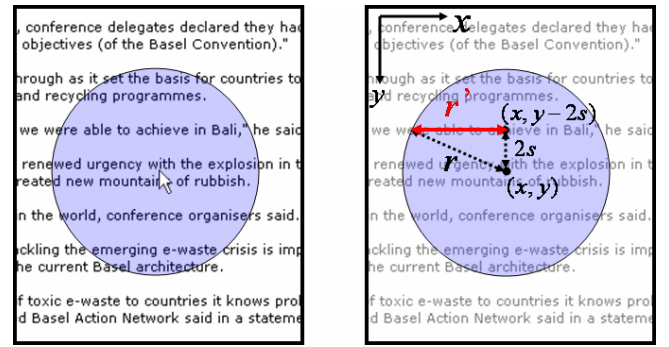


Figure 5: Extraction of strings contained in the circle

radius r ”. In order to extract all characters contained in the circle, the same process is applied to each line of text in the circle. Specifically, where the space between two adjacent lines (line space) is constantly s [pix], string extraction with center $(x, y \pm s \cdot k)$ and radius r' is applied to the line which is k lines away from the circle center. Here, r' is calculated by the following equation:

$$r' = \sqrt{r^2 - (s \cdot k)^2}. \quad (6)$$

Figure 5 shows an example of string extraction at the line which is two lines away from the center. Here, $r' = \sqrt{r^2 - 4s^2}$.

This process is repeated for all lines until the condition $s \cdot k > r$ is satisfied.

4.1.2 Calculation of importance

Next, each of the extracted words is assigned an importance value based on both the attributes of the word and its distance from the circle center. In the following, we describe the details.

First, free-standing words and compound words are extracted from the characters extracted in the previous step. The system considers consecutive nouns to be a compound word. Following the Click-Search method proposed in [4], the system calculates $F_m(w)$ based on whether the word is a proper noun or not and $p(w)$ based on $tf \cdot idf$. In our approach, we newly adopt $D(w)$, which is calculated based on the distance between the word and the circle center. Then, the sum of these values determines the importance of this word. The reason we give a higher priority to proper nouns, e.g., the name of a person or a place, is that they are often used as search keywords.

Specifically, the system gives a weight to each word based on whether the word is a proper noun as described in section 3, i.e., $F_m(w)$ is set as 1 if the free-standing word is a proper noun. As for a compound word, $F_m(w)$ is set as the number of contained proper nouns. On the other hand, we thought that characteristic words in the page often attract users. Therefore, statistical weight, $p(w)$, which is based on $tf \cdot idf$, is calculated by the following equation:

$$p(w) = tf(w) \cdot \left(1 + \log_2 \left(\frac{N}{df(w)} \right) \right). \quad (7)$$

Here, $tf(w)$ is the number of appearances of word w in the current page. In our implementation, we use the API provided by Yahoo! JAPAN to determine N and $df(w)$. N is

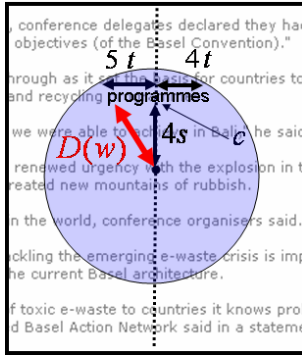


Figure 6: Calculation of $D(w)$

the total number of pages that the Yahoo! JAPAN search engine can get, which is set as 19,200,000,000. $df(w)$ is the number of resulting pages of a search using word w as the keyword.

Since users specify a search keyword by using an expanding circle in our interface, they stop enlarging the circle just when the circumference reaches the target word in most cases. Therefore, it is reasonable to assume that words located near the circumference are more important for users than ones located near the center. Let us denote $D(w)$ as the distance between the circle center and word w . Let us also denote $s[\text{pix}]$ as the line space and $t[\text{pix}]$ as the space between two adjacent characters (character space). Here, we focus on character c on the line which is k lines away from the line including the circle center. c is at the center position on the string in the circle. Then, the distance between the circle center and character i which is characters away from c , $d[\text{pix}]$, is calculated by the following equation:

$$d = \sqrt{(k \cdot s)^2 + (i \cdot t)^2}. \quad (8)$$

Consequently, $D(w)$ is defined as the larger value between the distance to the first character in w (from the circle center), d_L , and the distance to the last character, d_R :

$$D(w) = \max(d_L, d_R). \quad (9)$$

Figure 6 shows an example of calculating $D(w)$. The word “programmes” is 4 lines away from the circle center and the first and last characters are 5 and 4 characters away from c , respectively. Therefore, $d_L = \sqrt{25t^2 + 16s^2}$ and $d_R = \sqrt{16t^2 + 16s^2}$, and $D(w)$ is set as d_L .

Finally, the importance $E(w)$ of word w is calculated by the following equation using $F_m(w)$, $p(w)$, and $D(w)$:

$$E(w) = F_m(w) + p'(w) + D'(w). \quad (10)$$

Here, $p'(w)$ and $D'(w)$ are normalized values of $p(w)$ and $D(w)$ so that each maximum value becomes 1.

4.1.3 Automatic Selection and Listing methods

We designed the automatic selection and listing methods to determine a search keyword based on the importance of each word in the circle, which is calculated as described in section 4.1.2. The automatic selection method mainly aims at reducing users' operations and selects just one keyword as the word with the highest importance like the Google search's “I'm Feeling Lucky” feature. On the other hand, the

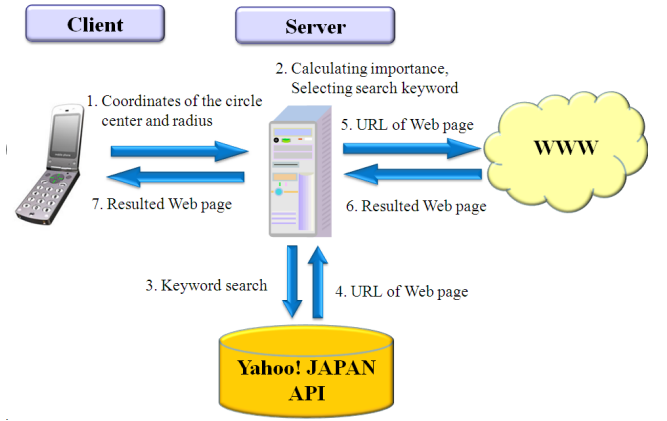


Figure 7: Architecture of the Click-Search interface

listing method mainly aims at avoiding the wrong keyword selection due to the single keyword selection. It selects the six words with the highest importance as search keyword candidates and lists them on the user's screen. Then, the users manually select the search keyword by pushing the corresponding numerical key.

4.2 Implementation

We implemented the Click-Search interface as a server/client architecture (See Figure 7) since cellular phones lack processing power at this time. The client sends both the coordinates of the circle center and radius after the user completes to adjust the circle. Then, the server determines the search keyword from the words contained in the circle. Note that in the listing method, the keyword selection requires the user's decision. The server performs the search operation by specifying the keyword using the API provided by Yahoo! JAPAN[13], and get the Web page which is the top search result. Then, the server sends the resulted Web page to the client, and the client presents the received page on the user's screen.

The server was implemented on a PC with Windows XP and coded with Visual C# and PHP (hypertext preprocessor). We implemented a prototype of the client using “BREW simulator”, which is a BREW emulator provided by Qualcomm[7], and coded with BREW[3]. However, our proposed system does not restrict the programming platform to BREW, and can be implemented using other languages such as C++ and Java.

Since the technical progress of cellular phones is remarkably rapid, we expect that it will be possible to implement the whole system (both the server and the client) on a cellular phone in the very near future.

The current version of our system can deal only with Japanese pages. Therefore, though examples of interfaces in this paper are in English, the real interfaces in the implemented system is in Japanese. The Click-Search interface uses MeCab[5] for Japanese morphological analysis.

5. EVALUATION

In this section, we describe the experiment conducted to verify the effectiveness of our proposed interface.

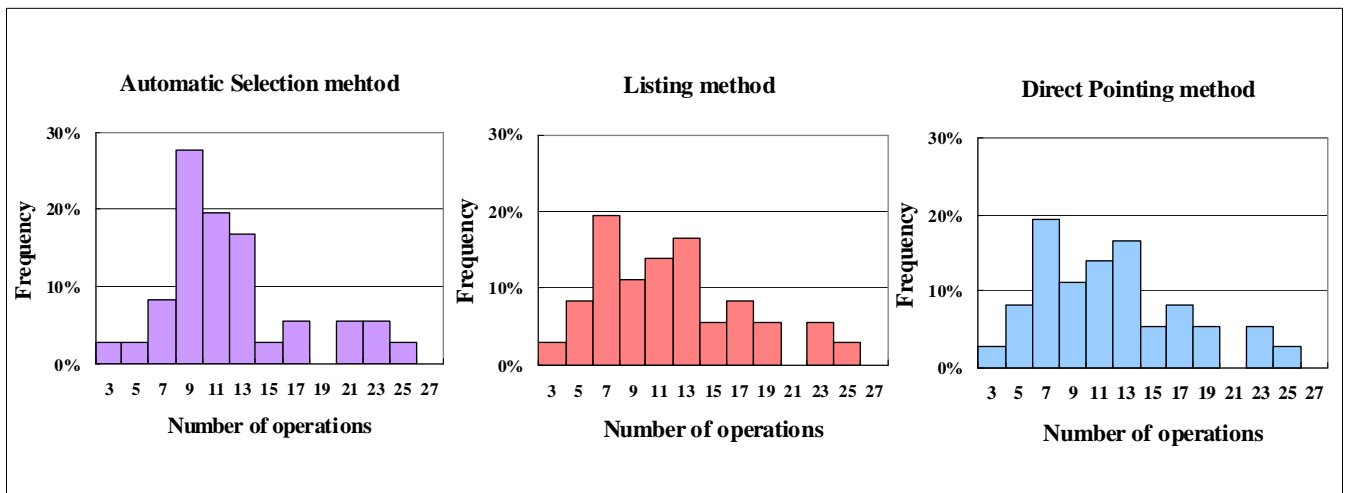


Figure 8: Number of operations

5.1 User experiment

Participants were ten men and two women from our laboratory. The participants executed nine tasks using three interfaces: the automatic selection and listing methods proposed in this paper and the “direct pointing” method. The direct pointing method is a version of the Click-Search method in [4] which has been modified for usage with cellular phones. In this method, users have to scroll the screen using the direction pad and specify a search keyword by pointing and clicking a word using the pointer displayed at the center of the screen. Since we implemented the client using a BREW simulator, the participants operated the BREW simulator using a mouse on the screen of the desktop PC.

We gave participants a few tasks for searching for a specific word on a Web page. For these tasks, we used Web pages of news articles and chose various types of words as search keywords to achieve fairness in the experiment: four person names, two place names, two general terms, and one unknown word. In order to simulate a common situation in Web browsing using a cellular phone, we instructed the participants not to immediately search a specified keyword, but to read the whole article first and then search the keyword.

Furthermore, in order to dissociate the interfaces and tasks, we divided participants into six groups and assigned each group a different combination of interfaces and tasks.

During the experiment, the number of operations for each task was automatically recorded. We asked the participants to evaluate subjectively the number of operations on a scale of five levels (1: few - 3: usual - 5: many) after finishing each task. We also asked them to evaluate how easy it was to use each interface on a scale of five levels (1: not easy - 3: usual - 5: easy) after finishing all tasks. In addition, the participants wrote free-form comments about the advantages and disadvantages of our proposed interface.

5.2 Results: Number of operations

Figure 8 shows the distribution of the numbers of actual operations for all tasks, which were obtained from the recorded operation log. The numbers of operations were widely distributed, thus, a parametric test is not suitable for

examining the result. Therefore, we conducted the Kruskal-Wallis test, which is a non-parametric test used for examining differences among groups, with 5% significance level. We found no significant difference among the three methods. This is because usual scrolling requires much more operations to read articles on a Web page than specifying a search keyword, thus, the differences in the number of operations among the three methods are relatively not very large.

Figure 9 shows the subjective impressions of the number of operations for each method, which the participants answered after finishing the task. From this result, we can see an interesting fact. 61% of participants selected level 1 or 2 (little or very little) for the automatic selection method and 83% for the listing method, whereas 11% selected 1 or 2 for the direct pointing method. This result shows that the number of operations when using the automatic selection and listing methods is perceived to be less than that when using the direct pointing method, i.e., our proposed interface enabled participants to specify a keyword with fewer operations. The listing method was perceived as requiring fewer operations, despite the fact that it requires one more operation than the Automatic selection method to select a keyword by using the numerical key from the list of the candidates. This seems to be because users prefer to know which word was used for the search. We got the following comment from a participant: “In the automatic selection method, the system should provide feedback to users about what word was set as the search keyword.”

5.3 Result: questionnaire survey

Figure 10 shows the results of the questionnaire survey asking how easy it was to use each interface. The average score was 3.25 for the automatic selection method, 4.25 for the listing method, and 2.75 for the direct pointing method. This result shows that the participants thought the automatic selection and listing methods were easier to use than the direct pointing method; The listing method was evaluated as the best. We got the following comments from the participants: “In the listing method, the target word I wanted to search was always on the list of the candidates,”

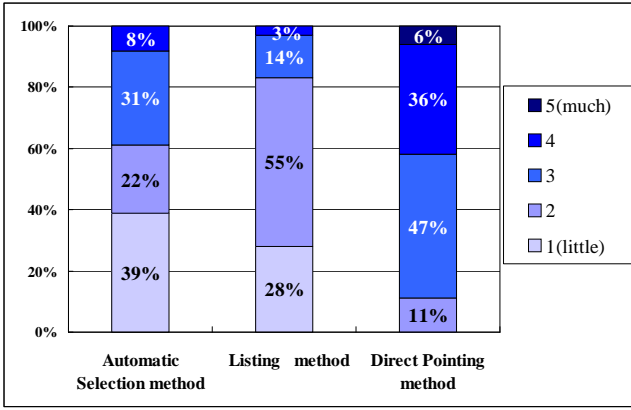


Figure 9: Subject number of operations

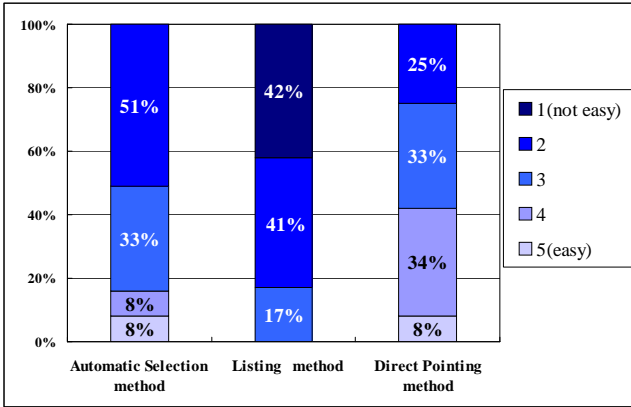


Figure 10: Easy to use

and “I could not understand which word was selected as a search keyword in the automatic selection method.” Therefore, in the automatic selection method, we should provide feedback, such as highlighting the selected word, so that users can recognize which word was selected as a search keyword. The listing method was preferred by the participants because they could always recognize the search keyword by selecting it from the list by themselves.

5.4 Result: accuracy

Figure 11 shows the accuracy of keyword selection including mis-operations, which are defined as failure cases where the participants’ targeted keywords were not included in the extracted words. Figure 12 shows the accuracy when ignoring mis-operations. Here, the accuracy for each keyword is defined as the ratio of search operations in which the keyword was correctly selected to all the search operations regarding the keyword. This result shows that the listing method always included the keyword successfully in the list of the candidates if the corresponding word was successfully contained in the circle.

As for the task using person name 3, the accuracy of the automatic selection method was 0%, even when the participants correctly included the word in the circle. The importance of person name 3 was low because there were many

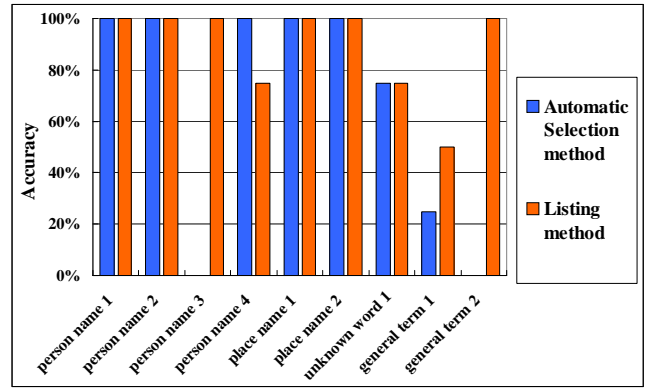


Figure 11: Accuracy of keyword selection including the mis-operations

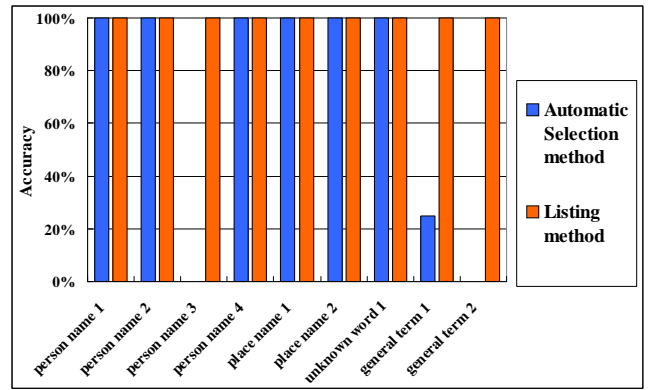


Figure 12: Accuracy of keyword selection when ignoring mis-operations

other person names near person name 3. The same thing also happened for general term 2, i.e., the automatic selection method was not able to select general term 2 as the keyword even when the participants included the word in the circle. This is because there were some proper nouns near general term 2 in the Web page, and the importance of general term 2 was smaller than those of the proper nouns. On the other hand, as for general term 1 and unknown word 1, the participants often could not include these words correctly in the circle. This seems to be because these words were long; eight characters and nine characters, respectively, and thus, it was difficult to include these words completely in the circle. Therefore, we should modify our methods to extract long words even if the words are only partially included in the circle.

6. DISCUSSIONS

6.1 Importance of words

The proposed interface selects a search keyword considering proper nouns, $tf \cdot idf$ and the distance between the word and the circle center. It is expected that the accuracy of keyword selection can be improved by using users’ browsing context. For example, when a user scrolls from left to

right and tries to select a keyword using our interface, we can expect that the user's target word is on the right side of the circle. Therefore, it might be effective to take the scrolling direction into account when calculating the importance of words.

6.2 Evaluation using cellular phones

In our experiment, we used an emulator on a desktop PC and evaluated tasks that required the participants to perform keyword searches. However, this is different from a real situation of Web browsing using a cellular phone. Specifically, operations on the emulator using a mouse are easier and simpler than operations using the direction pad on the cellular phone. Additionally, finding a specific keyword (requested by us in advance) is like "searching" rather than "browsing". Therefore, we need to conduct additional experiments using actual cellular phones in which users browse Web pages freely and perform keyword searches by specifying words of interest. Furthermore, we should compare our approach with some general Web searching methods, in which users input a keyword on a form using numerical keys.

7. CONCLUSION

In this paper, we proposed a Click-Search interface for cellular phone users. In the proposed interface, when a user pushes a key, a circle appears on the screen and keeps expanding as long as the user keeps pushing. The user releases the key when the target word is included in the circle. Then, the system selects a search keyword from the words inside the circle considering the attributes of the words and their distance from the circle center. Using this interface, users can specify a keyword simply by pushing a few buttons.

We conducted a user experiment and confirmed that our proposed interface enables users to specify a keyword with fewer operations.

In future work, we will consider a selection method to consider user's search intention and an effective presentation method of the resulting Web pages. Furthermore, we will conduct additional experiments using actual cellular phones on which users read Web pages freely and search for words of interest, and verify the effectiveness of our proposed interface in a real environment.

8. ACKNOWLEDGMENTS

The authors wish to thank Dr. Shigeyuki Akiba, President & CEO of KDDI R&D Laboratories Inc. for his continuous support for this study.

This research was partially supported by The Global Center of Excellence Program and Grant-in-Aid for Scientific Research on Priority Areas (18049050) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

9. REFERENCES

- [1] Y. Arase, T. Maekawa, T. Hara, T. Uemukai and S. Nishio: "A Web Browsing System for Cellular Phone Users based on Adaptive Presentation," Springer Universal Access in the Information Society, Vol. 6, No. 3, pp. 259–271, 2007.
- [2] au: <<http://www.au.kddi.com/>>.
- [3] BREW: <<http://brew.qualcomm.com/brew/ja/>>.
- [4] G. Hattori, T. Hara, F. Sugaya and S. Nishio: "Keyword Extraction for Click-search System," in Proc. DBWeb 2007, 2007, in Japanese.
- [5] MeCab: Yet Another Part-of-Speech and Morphological Analyzer: <<http://mecab.sourceforge.net/>>.
- [6] NTT DoCoMo: <<http://www.nttdocomo.co.jp/>>.
- [7] Qualcomm: <<http://www.qualcomm.com/>>.
- [8] D. C. Robbins, E. Cutrell, R. Sarin and E. Horvitz: "ZoneZoom: Map Navigation for Smartphones with Recursive View Segmentation," in Proc. Advanced Visual Interfaces (AVI 2004), pp. 231–234, 2004.
- [9] V. Roto, A. Popescu, A. Koivisto, and E. Vartiainen: "Minimap – A Web Page Visualization Method for Mobile Phones," in Proc. Conference On Human Factors in Computing Systems (CHI 2006), pp. 35–44, 2006.
- [10] SoftBank Mobile: <<http://mb.softbank.jp/>>.
- [11] M. Suzuki, K. Fume, and Y. Ishitani: "Successive Information Retrieval Method with Pen-based Interaction based on Estimation of User Intention," in Proc. Interaction 2006, 2006, in Japanese.
- [12] Wii: <<http://www.nintendo.co.jp/wii/>>.
- [13] Yahoo! developer networks : <<http://developer.yahoo.co.jp/>>.
- [14] Yahoo! oneSearch: <<http://mobile.yahoo.com/onesearch/>>.