PROCEEDINGS

The 25th IEEE International Conference on

Distributed Computing Systems Workshops

Columbus, Ohio, USA
6-10 June 2005

Sponsored by

IEEE Computer Society Technical Committee on Distributed Processing (TCDP)

IEEE COMPUTER SOCIETY

IEEE

Proc. of Int'l Workshop on Smart Appliances and Wearable Computing (IWSAWC 2005), pp.477–483 (June 2005).

An Information Retrieval System for Supporting Casual Conversation in Wearable Computing Environments

Nga Viet Pham†, Tsutomu Terada†, Masahiko Tsukamoto‡, and Shojiro Nishio†

† Department of Multimedia Engineering
Graduate School of Information Science and Technology
Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
{nga.pham, tsutomu, nishio}@ist.osaka-u.ac.jp

‡ Department of Electrical and Electronics Engineering
Faculty of Engineering
Kobe University
1-1 Rokkodai, Nada, kobe 657-8501, Japan
tuka@kobe-u.ac.jp

Abstract

In the future, wearable computers are going to be exploited for different applications to support human life. In this paper, we present a new wearable computing system that supports daily conversation based on speech recognition and information retrieval technologies. In daily conversation, sometimes we do not know the meaning of a new word or we want to know more information about a topic. In such cases, we can get more information to make our conversation more interesting and fluent by receiving auxiliary information from a wearable computer. The proposed system analyzes conversation content and retrieves information related to ongoing conversations from the Internet to make them more interesting and beneficial. We have implemented a prototype system, and the results of actual use have shown the need for such a system.

1. Introduction

Recently personal computers have become indispensable equipment in daily life, used not only in offices but also at home. Moreover, the Internet has spread worldwide; bringing many new services such as information retrieval, VOD, Internet phones, and video chats. For these reasons, personal computers are used for various purposes, not only for work but also for entertainment, especially for people's communication. On the other hand, the downsizing of portable computers has attracted the field of wearable computing [7]. As shown in Figure 1, in a wearable computing environment, a user always wears a wearable computer, a head set microphone, a pocket input device such as a trackball, and a head mounted display (HMD). The user can watch information by using the HMD, input data into the computer with a pocket input device, and employ voice commands with a head set microphone. Since wearable computers have many new advanced properties such as constanly being available and hands-frees, users can always use computers while doing other work. Therefore, the demand for a wearable computer system that supports human daily life is increasing to solve various living problems.

There are several researches into wearable systems. One is the presentation system that presents exhibit information in museums [3]. When a user looks at an item, the system displays information about the item on the HMD. Other information services that support people's lives such as daily reminders, navigation, and healthy management might also be provided by wearable computers. Our research focuses on applications that support conversation.

Conversation is an important element in human life where people exchange information on various issues. During a conversation, users may want to know the meaning of a word they don't understand. They also might need to learn more about a topic being discussed. For that reason, it is very convenient and valuable to receive support from

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 IEEE

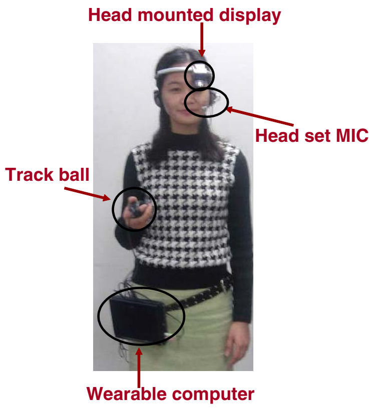

Figure 1. Typical style of wearable computing.

computers during their conversation to get more related information. However, until now no wearable system has met such a demand. From such motivation, we propose a wearable system that supports users by retrieving necessary information in real-time during a conversation. The proposed system can retrieve information related to the content of a conversation by automatically selecting keywords in conversations by using speech recognition and text analyzing techniques. Using these keywords, the system searches the Internet for related information that users might be interested in and displays it on the HMD.

The paper is organized as follows. Section 2 describes the system design issues in terms of the voice input process, text processing, query creation, selections and presentation of results. Section 3 presents a detailed description of a practical implementation of the system. After that, system performance and properties are discussed in Section 4. Finally, Section 5 gives a conclusion.

2. System Design

The system receives voice input from conversations, processes this information, and next presents related information on the HMD. The related information is a set of WWW pages retrieved based on conversation content displayed consecutively on several browser windows. The proposed system operates following this procedure:

1. Voice input and recognition perform the input and recognize the user's voice. Conversation content is inputted as text segments.

2. Keyword extraction processes the inputted text segment and extracts a keyword vector that consists of several keywords and their strength. Keywords represent the content of a text segment.

3. Topic classification evaluates the extracted keyword vector and judges if the inputted text segment belongs to one of the conversation topics.

4. Query creation creates a query using the keyword vector and the content of the classified topic and sends it to a search engine.

5. Search result selection analyzes the obtained results list to choose the optimal results.

6. Presentation of reference results, the selected results are shown to the user.

Hereafter we describe the processing steps of the operation procedure in detail.

2.1. Voice input and recognition

A conversation's voice is imported to the system by using the head set microphone and regconized by a speech recognition program. The system extracts speech text for every segment of a fixed text length denoted as the $text\ segment$.

Since the text segment expresses part of the user's conversation, extraction is fundamentally performed by considering breaks in utterances. In fact, there are cases when an utterance may continue for a long time or the utterance break cannot be recognized due to noise. Therefore, our system carries out extraction when the length of the inputted text exceeds a certain number. In summary, when one of the following conditions is satisfied, the input voice is extracted as a text segment.

- When no voice is inputted more than α seconds.
- When the number of words in input texts exceeds β.

Since the length of a text segment depends on α and β, if they change, the information retrieval results also change drastically. Therefore, setting the optimal value of α and β is an important and difficult issue. In Section 4, we discuss how to set appropriate values for α and β.

2.2. Processing text segments

The system creates a keyword vector based on extracted text segments in the following three steps.

1. *Keyword extraction*: The keywords of a text segment are a set of words used to express the content of the text segment. In our research, we realize that nouns express the content of a text segment better than other components of speech. Therefore, all nouns appearing in the text segment are extracted as keywords of a text segment.

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 IEEE

2. *Weighting keyword*: The system evaluates the importance of each extracted keyword and attaches weight to keywords using the tf-idf methods which is a technique widely used, to evaluate the importance of a word [4]. In this technique, the importance of each word t, $I(t)$ is computed by the following formula:

$$I(t) = tf(t, T) \times idf(t).$$

Term Frequency ($tf(t, T)$) shows the frequency that a certain word t appears in text segment T.

Inverted Document Frequency ($idf(t)$) is a value based on the number of documents in which word t appears and the total number of documents. $idf(t)$ is defined by the following formula:

$$idf(t) = log(N/f(t)).$$

N stands for the numbers of target documents. $f(t)$ is the number of documents that include word t. $idf(t)$ expresses the concentration of word t in some documents. If word t only appears in a few documents, its $idf(t)$ and its importance $I(t)$ increases. However, if a word only appears a few times in T, then its $tf(t, T)$ is small and $I(t)$ also becomes small.

3. *Topic classification*: We acquire information according to the content of a text segment by using a keyword vector. However, if a text segment is processed individually, retrieved information cannot explain the appropriate information concerned the content of real conversations. Generally, a conversation is composed of a number of short conversations (topics) during which a certain subject is mentioned. Conversation content varies with changes in subjects.

   Our research considers a topic to be a set of text segments relevant to each other that relate to a certain subject. Since this system is carrying out information retrieval during ongoing conversations, retrieval frequency has to be set at an appropriate value. Therefore, the length of the text segment used for carrying out information retrieval has to be precisely selected. If the text segment is too short, a query cannot obtain reference results that are in alignment with the subject. Moreover, conversation subjects may change suddenly or return to a previous subject. Therefore, the proposed system judges whether the text segment can be classified into one of the previous topics. By using the content of the classified topics, searching accuracy increases. If the text segment does not belong to any previous topics, the system regards it as a new topic. A topic is created by the following steps:

   (a) When a topic already exists in the system, by comparing the content of the extracted text segment with the content of each old topic, the system judges if the extracted text segment is similar to a former topic.

   (b) When there isn't any topic existing in the system or no appropriate topic having similar content with the extracted segment, the extracted text segment becomes a new topic. At this time, the keyword vector of the text segment becomes the keyword vector of the new topic.

   To process this procedure, the system computes similar degree p, which is the inner product [2] of the keyword vectors of text segment T and a topic. Next it evaluates value p using a threshold value γ.

   - When $p > \gamma$, the system judges that the extracted text segment T has a relation with topic t, and T is then classified into t. At this time, keyword vector K_t of topic t is updated using keyword vector K_T of extracted text segment T. To implement this, K_T is added to K_t. Finally, the system normalizes the result and names it the new keyword vector of the topic.
   - If $p < \gamma$, the system judges that the extracted text segment T is unrelated to topic t. Therefore, T is not classified into t. In this case, the system creates a new topic t' and classifies T into t'.

2.3. Query creation and selection of searching results

After text segmentation and topic classification, the system creates a query. When the extracted text segment does not belong to any known topic, the system uses the keyword vector of the text segment to create a query. Specifically, keywords of a vector are ordered based on their importance values. Then the system selects the most important q words and sends them to a search engine. When the extracted text segment is classified into a certain topic, a query is created in accordance with the content of the classified topic. In this case, we multiply the keyword vector of the topic by r. Then they are totaled with the keyword vector of the text segment, and the most important q words are selected to create a query. Since the value of r and q affects the search results, we will validate them in Section 4.

After the system sends a query to a search engine, it selects and displays the most optimal page from the query results.

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 IEEE

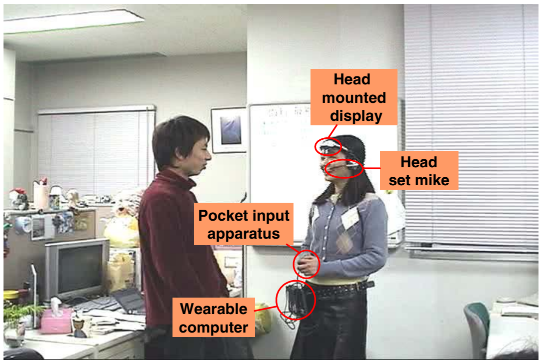

Figure 2. Example of proposed system.

2.4 Presentation

The selected results are presented to the user. In a wearable computing environment, although a HMD is positioned in front of a user's eyes, that doesn't mean the user is always gazing at the display. She only watches the HMD when interesting results appear. In this system, since a number of browser windows are simultaneously displayed, much information can be offered at the same time. However, since reading a page in the HMD may be more difficult than with a regular display, smooth browsing is difficult when a number of browser windows are displayed simultaneously. Thus, it is necessary to devise a method of information presentation, such as expanding the pages in which users are interested. The detailed user interface is described in Section 4.4.

3. Implementation

We implemented a prototype of the web search system that supports conversation in a wearable computing environment by using Microsoft Visual Basic 6.0 on a Sony PCG-SRX7. Figure 2 shows a snapshot of a user equipped with a wearable computer, a head set microphone, and a HMD. We used a Shimazu Data Glass 2 as a HMD, a Sony VAIO PCG-U3 as a wearable computer, and a Soho FDM-G51 track ball as a control device of the system.

At the moment, this system is implemented for Japanese conversation, it uses IBM ViaVoice 98 for Japanese speech recognition. When applying the system to another language, users must use another language recognition program that may need modifications.

The prototype utilizes the Japanese morphological analysis software ChaSen released by NAIST Computational Linguistics Lab [5]. The main functions of Chasen include full-text search and extraction of specific parts of speech. Unfortunately, Chasen is not applicable for other languages.

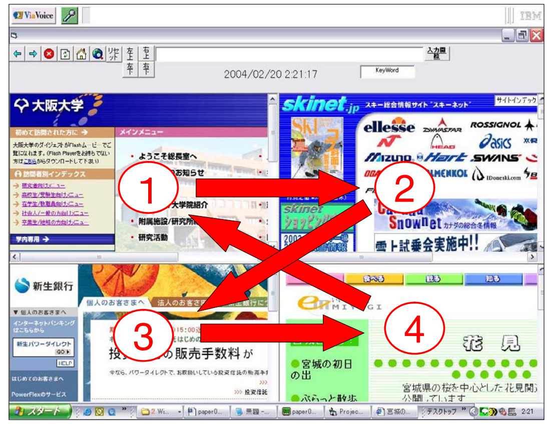

Figure 3. Order of displayed information.

3.1 Query creation parameter

We described the creation method of a query in Section 2.3. In the prototype system, we let $q = 2$ be the number of keywords used for a query. Threshold value γ, which is used to judge whether a text segment is classified into a topic, is set to 0.3. We let the strength of using topic information r be 0.9. Moreover, the first ranked page among the search results is set as the page shown to a user.

3.2 Information presentation

In the prototype system, searching results are displayed consecutively on four internet browser windows as shown in figure 3, i.e., the previous four information pages are displayed simultaneously.

As seen in figure 3, besides operation buttons for performing general web browser functions, we have created specific operation buttons for this system. A cluster of four buttons correspond to four lower browser screens and perform fixation and expansion of each screen. We also created a button to reset the expanded state. Since a expanded window is fixed and then no longer updated, a subsequent search is performed using the three remaining browser screens.

4. Discussion and experimental results

4.1 Parameter Settings

The system extracts q words having the highest importance for each information retrieval. Moreover, the page having the highest rank of Google search results is used as a result. When a page is selected, the system presents it without any change in its the content. In this situation, important issues impacting retrieval accuracy are the number

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 IEEE

of query words and page selection to show to users as reference results. First we consider the number of query words. Generally, when more words are used for a query, retrieval accuracy increases. However, this is not always correct because the system might extract the wrong keywords. The following relations are materialized between the number of query words and retrieval results.

- If the number of words increases, we can make the query concrete but the number of hits decreases. Especially when the length of a text segment is short and the number of query words is high, words, which are not suitable as reference words, sometimes enter a query. In this case, we may not get any reference results.

- If the number of words is reduced, we can get more hits. However, it is hard to get reference results that concretely express conversation content, and therefore retrieval accuracy falls.

We carried out an experiment that evaluated the optimal number of words for a query. As cited below, through an evaluation we determined that a text segment length between 60 and 80 words gives high performance. For such text segment length, two keywords for a query is the best choice. However, during the experiment, we also recognized that sometimes the two words of a query are poorly connected to each other, in such cases, the system could not produce good reference results. In our system, users can also adjust the length of text segment to be extracted depending on interests, and the system changes the number of query words according to text segment length. One future work is to make a system that can set the number of words adaptively to raise reference accuracy. For instance, we can set system search using many words. After considering the number of hits or content of reference results, the system will adjust the number of words.

4.2 Speech recognition

This system uses the Japanese speech recognition system ViaVoice 98 (IBM Corp.) [6]. Since it adopts a speaker that can respond to consecultive utterances, it can continuously input the content of daily conversation.

In the system, retrieval information is presented one after another all the time. The user only watches the display when interested in a page. Thus even if the accuracy somewhat falls or when a meaningless retrieval result appears, users merely disregard such information. Therefore, some changes in recognition accuracy hardly affect the usefulness of our system. The current system only employs user's voice input. However, if other external sounds are used for information retrieval, more useful information may be obtained. For instance, when the user walks along a street, if the system also performs information search based on surrounding sounds, much various and interesting information can be acquired.

Therefore, we are considering not only using a speech recognition system that can recognize voices, but also an advanced speech recognition system that can recognize all sounds.

4.3 Extraction of a text segment

When there is no voice input for α seconds or more, or when the number of words in an input text exceeds β, the extracted input is considered a text segment. When either α or β changes, the length of the extracted text segment changes, and the operation of the proposal system will also change a lot. If α increases, it causes a certain delay from the time the conversation breaks to the time the system actually detects this pause. For this reason, general retrieval processing becomes slow. On the other hand, if α becomes small, the system will extract a segment, even just a breath. For this reason retrieval accuracy will decrease. If β increases, the reaction of the system will become slow as in the case where α is enlarged. And if β becomes small, since the segment length also decreases, retrieval accuracy will deteriorate. Moreover, if either α or β is made too small, a text segment will be extracted that depends only on the parameters. Refering to M. Henzinger [1], we predict that a value of β near 60 words (one Japanese Kanji, Hiragana, Katakana or one digit of number are counted as a word) is the optimal value when applied to Japanese texts. Actually we carried out experiments to find out the optimal text segment length. Optimal length depends on the characteristics of individual speaking styles and conversations. According to our experimental results, a text segment length between 60 and 80 words gives high performace. As mentioned before, this setting will only become the system's initial setting. Users can adjust the length of the text segment to be extracted depending on interests. In the prototype system, we experientially set α= 3. In the future, we will evaluate the influence of α on reference results.

4.4 Presentation method of reference results

In the wearable computing environment, users only watch the HMD when they want to look at some interesting items. So the information presentation has to provide as much information as possible on the HMD. We chose an information display method in parallel using four browser windows. However, by concurrently displaying the number of browsers concurrently, the size of each browser decreases, and vision may become too difficult. Actually we have already tested the system using HMD. The achieved results allow us to claim that simultaneous display with four

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 IEEE

browsers does not increase perusal difficulty. At the moment, one browser window is used for displaying the retrieved information of a query. The use of two or more browser windows at the same time for each query is now being considered. In our system, a user can stop the change of a specified browser, magnify or reduce it. This makes perusal more convenient when using the HMD. A new user interface is considered as future works.

4.5 Input device

In our proposed system, users employ buttons on their interface to control it. For input devices, users can use a handy track ball. One extention of our research is to adapt our system to wearable input devices other than the handy track ball.

4.6 Setting parameters in topic classification

When the similarity of a extracted segment and a topic is more than threshold value γ, the system will judge that they are relevant. We carried out experiments to evaluate the impacts of γ on retrieval results. The results allow us to claim that a value of γ between 0.2 and 0.3 gives high performance. Moreover in our system, before using a topic to make a query, we weight it with r. Citing [1], we set r equal 0.9. When r becomes too big, we get results that depend too much on the topic. On the other hand, if r becomes too small, users get search results that consider conversation flow. The evaluation of the impacts of r on retrieval results are for future works.

4.7 Experimental results

We experimented on the performace of the prototype system. Two objects were evaluated. The first is reference accuracy. The second evaluated whether the URL is interesting. We used a six minute thirty second length conversation as an inputs. Text segment length is 60 words. The system returned a URL after processing each text segment. For the purpose of evaluation, only conversation's log and a list of resulting URLs was given to subjects. The subjects evaluated each URL by scoring each. Min score is 1 and max score is 5. The meaning of each score is shown in table 1. The number of URLs related ($score >= 3$) and very related ($score >= 4$) to the conversation and the number of URLs interesting ($score >= 3$) and very interesting ($score >= 4$) are shown in tables 2 and 3 (the first speaker) and tables 4 and 5 (the second speaker).Total number of URLs returned to speaker 1 is 12, and that for speaker 2 is 31.

The experimental results show that in a short conversation, both speakers got an average of at least five related URLs, two very related URLs, five interesting URLs, and one very interesting URL. For speaker 2 who made many utterances, the numbers were much larger. In a wearable computing environtment, since users do not look at HMD all the time, showing related URLs and interesting URLs even at the above rate is enough to make conversaton fluent and interesting.

Table 1. Rating sheet.

Score	Evaluation
1	unrelated (boring)
2	a litle bit related (a litle bit interesting)
3	related (interesting)
4	very related (very interesting)
5	very related (very interesting)

Table 2. Number of related and very related URLs returned to speaker 1.

Subject number	Number of related URLs	Number of very related URLs
1	5	4
2	3	0
3	2	1
4	7	5
5	6	3
6	6	0
7	5	3
8	3	2
9	9	7
Average number	5.11	2.78

A detailed evaluation with many more subjects and objects is a future work. However, as found from recent experiments, with the present reference accuracy, a user's conversation is supported well.

Since each user in a conversation has different interests in the same conversation subject, we are investigating a method of selecting information pages according to the individual interests of users.

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 **IEEE**

Table 3. Number of interesting and very interesting URLs returned to speaker 1.

Subject number	Number of interesting URLs	Number of very interesting URLs
1	5	0
2	3	2
3	0	0
4	5	1
5	2	0
6	5	1
7	9	2
8	9	4
9	9	3
Average number	5.22	1.44

Table 4. Number of related and very related URLs returned to speaker 2.

Subject number	Number of related URLs	Number of very related URLs
1	8	2
2	12	7
3	5	2
4	13	11
5	9	0
6	6	1
7	11	5
8	20	13
9	20	14
Average number	11.22	6.11

5 Conclusion

In this paper we proposed a novel wearable computing system to support daily conversation for retrieving useful information from the internet to make users' conversations more interesting and beneficial. Our approach fulfills the growing need for support of daily conversation. A system's prototype was implemented and its performance was evaluated by practical human-based experiments. The current prototype is based on speech recognition technology and particular text processing procedures designed for Japanese. Future work is to carry out a more detailed experiment with a larger number of subjects. Another future work is to optimize the rest of the system parameters. Advanced techniques for text processing and retrieval information presentation will also be investigated. More practical experiments will also be carried out to evaluate system performance for indifferent real-time conversation environments.

Table 5. Number of interesting and very interesting URLs returned to speaker 2.

Subject number	Number of interesting URLs	Number of very interesting URLs
1	10	2
2	7	4
3	2	0
4	16	6
5	8	0
6	10	1
7	14	10
8	27	16
9	17	7
Average number	12.33	4

Acknowledgments

Part of this research was supported by the 21st century COE program "New information technologies for building networked symbiosis environment" and Grant-in-Aid for Scientific Research (B) numbered 15300033 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] Henzinger,M., Chang,B., Milch,B., Brin,S.: Query-Free News Search; Proc. The twelfth international conference on World Wide Web (WWW '03)(May. 2003).

[2] H.Lohninger: Inner Product of Two Vectors; http://www.vias.org/tmdatanaleng/cc_vector_innerprod.html (2004).

[3] Koshizuka,N., Sakamura,S.: Tokyo University Digital Museum; Proc. the 2000 Kyoto International Conference on Digital Libraries: Research and Practice, http://www.l.u-tokyo.ac.jp/~koshi/ PAPERS/index-j.html (2000).

[4] J. Stephen Downie: Term Weighting: TF * IDF; http://instruct.uwo.ca/gplis/601/week3/tfidf.html(1997).

[5] NAIST Computational Linguistics Lab: The Japanese morphological-analysis software Chasen; http://chasen.aist-nara.ac.jp/index.html.ja (2003)(Japanese).

[6] IBM Japan: Voice recognition; http://www-306.ibm.com/software/voice/viavoice/.

[7] MIT Media Lab: http://www.media.mit.edu/wearables/; (2000).

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 IEEE