SURFIN' THE WORLD WIDE WEB WITH JAPANESE

Kazuhiro Kondo and Charles T. Hemphill

Texas Instruments Personal Systems Laboratory
P. O. Box 655303, MS8374, Dallas, TX 75265
{kkondo, hemphill}@csc.ti.com

ABSTRACT

Previously, we have developed Speech-Aware Multimedia (SAM) which controls a WWW browser using English speech[1]. We recently extended its capability to use Japanese speech to browse Japanese pages, and developed a prototype using speaker-independent, continuous speech recognition with Japanese context-dependent phonetic models.

Some challenges not seen in English include: segregation of Japanese text into word units for optional silence insertion, Japanese text to phone conversion and accommodation of English link names embedded in Japanese pages. In order to accomplish the first two, we modified a public-domain dictionary look-up tool for segmentation and to accommodate heruristics required for improved text-to-phone conversion accuracy. Preliminary tests show that the conversion result contains the correct phone sequence over 97% of the time, and the prototype correctly understands the input speech 91.5% of the time.

1. INTRODUCTION

Since the introduction of the World Wide Web, traffic has been growing at an enormous rate due to its increasing popularity. Graphics-rich Web pages have attracted many newcomers who have never used computers before. For these naive, perhaps keyboard-shy users, as well as expert users who want to use all the input methods they can, voice control has the potential to make browsers more friendly and powerful. We have introduced Speech Aware Multimedia (SAM) in accordance with such features[1].

SAM uses speaker-independent speech recognition to enable users to navigate the World Wide Web by voice. Some specific features of SAM include:
(1) Speakable commands: Allows users to speak browser control commands, *i.e.* "go back" and "scroll down."
(2) Speakable hotlist: Allows users to go to a frequently-visited Web page by speaking user-configurable phrases, *e.g.* "My school home page" and "John's page."
(3) Speakable links: Allows users to speak any link name on the current page to go to that link.
(4) Smart pages: Pages which have embedded grammars allow flexible voice interaction. As an example, a user can search for the forecast of a particular city on a weather page with smart grammars by speaking the city name in a query sentence, *e.g.* "What's the forecast for Los Angeles, California?"
SAM has been released to a small number of groups for beta testing and received exceptional response.

During the past year, Japan has had its share of internet enthusiasm, especially for the Web. Perhaps, the timing was delayed by more than a year for a few reasons: lack of a Japanese capable web browser (now there are a number of browsers, including Netscape Navigator and some flavors of NCSA Mosaic with localization patches including Microsoft Internet Explorer), lack of service providers (many nationwide service providers have gone into service during 1995), and the cost (this still seems to be a problem although competition is lowering the prices). But with the gradual resolution of these problems, we are seeing a sudden increase in traffic and the number of Web pages written in Japanese has increased drastically. The general Japanese population, which had much lower access to computers compared to the U.S., has suddenly started using keyboards. If these naive users were allowed to use speech to access the Web, they would have a much smoother introduction to the computer and the internet. Accordingly, we modified SAM to also accept Japanese speech to access Japanese Web pages. There were a number of problems not seen before when we were dealing with English. These will be described in the next section. The architecture and features of the prototype will be described in Chapter 3. Chapter 4 gives the result of preliminary user tests. Finally, conclusions and plans are given.

2. ISSUES IN JAPANESE

There are two major issues that need to be solved in order to incorporate Japanese into SAM; problems occurring from the language itself and from its electronic notation. These will be described in turn below.

Japanese is a language which does not explicitly delimit written text at a sub-sentence level. English words are explicitly separated into words by spaces. Having a known unit boundary in the sentence provides two major advantages when converting the sentence to a grammar used in speech recognition.

Copyright 1997 IEEE

Table 1: Examples of Ambiguities in Unit Segmentation

Example	Parsed result	Romaji notation
1	音声	onsei
	音 声	oto koe
2	学校へ いく	gakkooe iku
	学校 へいく	gakkoo heiku

First, an explicit unit boundary decreases the ambiguity in converting from text to phonemes. Table 1 illustrates examples where the text can be parsed in two or more ways, possibly giving quite different meanings and pronunciations. In the parsed results column, each segmented unit is divided by a space. In the first example, both parsed results are legitimate, but we must use higher level context to distinguish the two. The first result, "onsei" refers to "speech", and the second result, "oto koe" refers to "sound, voice". They are related but their pronunciations differ significantly. In the second example, only the first result is legitimate, but the correct segmentation to units requires grammatical knowledge of the language.

Second, most pauses within a sentence appear at unit boundaries. In English, optional silences are often inserted between words. However, in Japanese, we need to identify appropriate segments to insert optional silences.

The next problem lies in the electronic notation of Japanese on the internet. Currently, there are three major coding systems used for Japanese[2]: (1) Extended Unix Code(EUC), (2)JIS code, and (3) Shift JIS code. All these codes are 2-byte codes. JIS code is mode-oriented; they use explicit character sequences to "shift in" and out of Japanese mode. Shift-JIS and EUC do not have modes. As its name implies, EUCs are mostly used on UNIX workstations, and so WWW servers on UNIX workstations commonly use EUC. Shift-JIS code is mainly used on PCs (both Wintel and Apple Macintosh), and so is commonly seen on servers on PCs. JIS codes are still the most popular for transmission. JIS codes use only the lower seven bits. Since some old transmission systems used to drop the eighth bit, JIS codes were the only codes which were received intact. Recent systems generally are now capable of transmitting all bits transparently. We have seen all kinds of combinations of the above three codes, even on the same page. Thus, it is necessary to detect and deal with all possible combinations of coding systems.

3. JAPANESE SAM

3.1 Overall Architecture

Japanese SAM is a flexible vocabulary Japanese speech recognition system which can adjust the vocabulary on the fly according to the page content. Figure 1 shows the overall architecture of the system. From the Web browser, the HTML which encodes the current Web page is sent to the code-conversion/parser module which converts all incoming code to EUC and parses the result to obtain link-name/URL pairs. A URL is the location of the resource (this may be another page, a GIF image, or an address to which the link name points). The link names are then fed to the dictionary look-up module for conversion first to tokens and then to a phonetic string. The phonetic string and the URL are fed to the grammar preparation module for conversion to a recognition grammar to be fed to the speech recognizer. Smart page grammars[1], grammars pointed to by embedded tags in some pages to define a specific recognition grammar to be used for that page, are also fed to the grammar preparation module. Speakable hotlinks, a list of favorite page locations with corresponding user-defined key phrases, as well as speakable commands are also fed to this module. The grammar preparation module outputs grammars which define valid phonetic sequences for each item. The speech recognizer uses these grammars to recognize valid phrases from its speech input, and ouputs recognized utterances to the interpretation module. The interpretation module then converts these key phrases into browser commands. For example, if a link name was recognized, the interpreter outputs a "goto *URL*" command. The same thing happens for a recognized speakable hotlist item. If the recognized item is a speakable command, then the interpretation module issues the appropriate browser command, *e.g.* "scroll up."

Some of the key modules will be described in detail in the following sections.

3.2 Code Conversion

As stated before, there are three major coding systems for Japanese, plus some minor variations. We have seen all combinations of these codes used in a mixed manner on Web pages. Even though there are tools that convert one coding system to another, none of these systems were robust enough to handle all combinations encountered on the network. Thus, we detect possible combinations up front, and convert each code into a single coding system, the EUC. EUC was selected for the internal processing code since it is not a mode oriented code and is well suited to text processing. With a mode oriented code, we must track the various modes in different subphrases. EUC is also easy to distinguish from ASCII codes since they occupy different ranges in the 8-bit notation.

3.3 Segmentation, Phonetic String Conversion

In order to convert Kanji-Hiragana mixed notation into a Hiragana sequence, we decided to use "kakasi[3]," a public-domain dictionary look-up tool. This program uses a simple left-to-right matching between its input and the dictionary entry. However, the program is fairly

Copyright 1997 IEEE

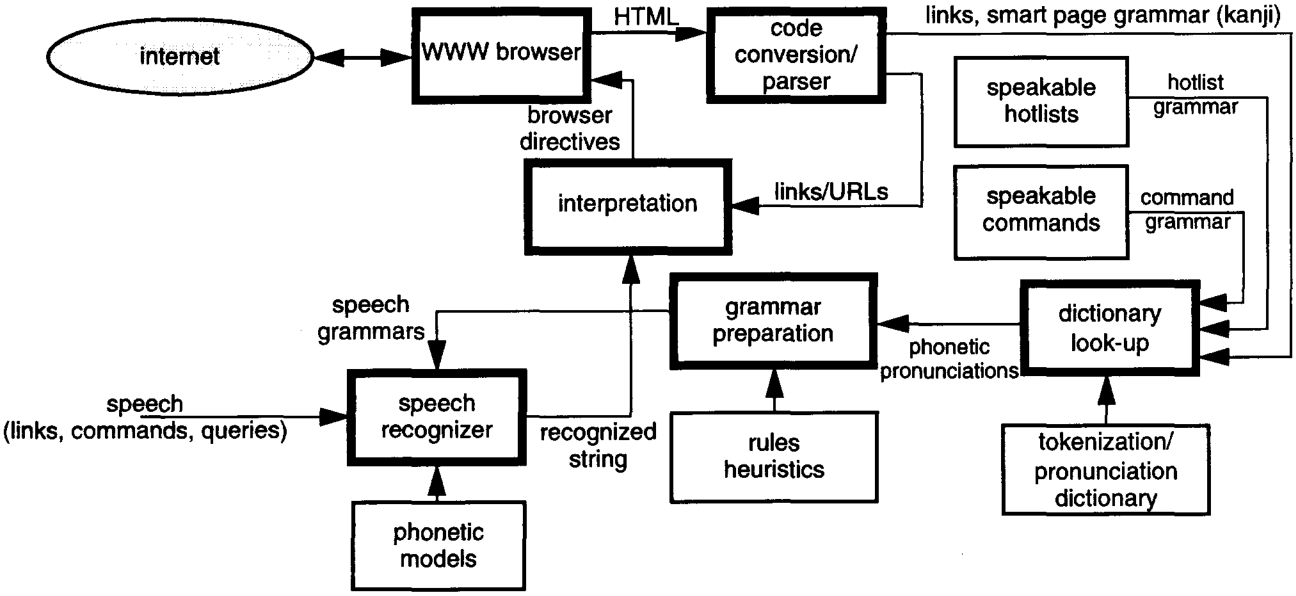

Figure 1 Architecture of Japanese SAM.

accurate if the dictionary is large enough. Currently, our dictionary contains 357,203 entries. Moreover, this program uses caching and hash tables very efficiently, and thus executes quickly.

The conversion of Hiragana to phonetic strings is accomplished using rules. This conversion process is fairly straightforward except for the conversion of certain particles and allophones. All possible alternate pronunciations are generated using the defined rules and incorporated as alternate paths in the pronunciation grammar.

Some other considerations during the conversion process include:

(1) Segmentation of flat Japanese normally involves morphological analysis. This is a very expensive process, and was not appropriate for a real-time system. Previously, we have constructed a large dictionary[4] in which entries are approximately in morphological units. We used this dictionary with "kakasi", and found that this gives surprisingly accurate boundaries.

(2) Some heuristics were necessary to convert phrases involving numbers. There are some traditional readings of dates, age, and counters. These needed to be dealt with separately. Fractional numbers also required heuristics.

(3) We added an optional exit after three units in a link name. In other words, the user does not have to speak the whole link, but can stop after speaking three segmented units.

(4) The majority of Japanese Web pages have some English embedded in them. These range from just the key technical terms embedded in a mostly Japanese sentence to a whole sentence entirely in English. It was observed that even users capable of speaking English fluently will have some tendency to pronounce English words embedded in Japanese pages with a heavy foreign accent. Thus, we decided to use a Japanese approximation of these words and Japanese phonetic models. This also cuts down on the amount of memory required to load the models. Preliminary experiments show fairly good accuracy.

3.4 Speech Recognition

We use a speaker-independent continuous multi-variate single mixture HMM recognizer developed at Texas Instruments. The recognizer uses a set of context-dependent phonetic models, whose context is defined by phonological features[5]. The context was clustered using a binary decision tree. The recognizer incorporates grammars dynamically derived from the link names and smart-page grammars in each new Web page visited.

4. EVALUATION RESULTS

4.1 Text-to-phone Conversion Test

In order to get a rough idea of how accurate the link name-to-phone conversion was performing, we ran a small benchmark test. We collected 117 link names from a few Japanese newspaper web pages, namely Asahi, Mainichi and Nikkei. The link names contained political, financial, and technical terms. These link names were processed with the dictionary look-up module described above. Table 2 shows the result along with the result for the original "kakasi." The expanded dictionary, along with the added heuristics, improved the accuracy significantly.

4.2 System Usability Test

We have built a prototype system on a UNIX platform. In order to get an idea of how usable the prototype is, we conducted a small-scale user test. We

Copyright 1997 IEEE

Table 2: Text-to-phoneme Accuracy

Module	Accuracy [%]	
	Sentence	Phone
original "kakasi"	57	92.4
revised	82	97.2

asked 5 users, 3 female and 2 male, to actually use the prototype and surf Japanese WWW pages.

In order to keep the tests fairly simple and small, the users were restricted to use only the following six commands:

(1) "Sukurooru" or "Kai peeji": Scroll down.
(2) "Ueni Sukurooru" or "Ueni kai peeji": Scroll up.
(3) "Peeji sai roodo": Reload page.
(4) "Hoomu peeji": Go to home page.
(5) "Maeni modoru": Go back.
(6) "Sakini susumu": Go forward.

The users were asked only to use these commands, or read the links. A local page was created as a starting point with some links of general interest: two major Japanese newspaper home pages ("Asahi" and "Yomiuri"), one technical magazine page ("Nikkei Electronics"), the Japanese Prime Minister's page, the Sumo Wrestling Association Page, the Japanese Yahoo Yellow Page, and the Texas Instruments Japan home page. The users were given approximately 30 minutes of free surfing. However, the users were asked to generally stay away from English links and English pages. Table 3 summarizes the results.

Table 3: User Test Results

Speaker gender	Accuracy [%]		
	All utterances	Commands	Links
All	91.5	94.7	88.4
Male	89.4	87.2	92.5
Female	92.9	98.8	85.8

In these tests, utterances which had insertions and deletions were counted as correct if the resulting action was what the user intended. Thus, for 91.5% of the speech input, the task was accomplished successfully.

Some observations from the tests were as follows:

(1) Approximately 44% of all utterances were links, while the remaining 56% were browser control commands.

(2) An average of 65.89 lines per visited WWW page was seen, ranging from maximum of 395 to 0. An average of 21.36 links were in one page, ranging from maximum of 331 to 0.

(3) The cause of errors were about evenly split between speech detection error, invalid speech input, and speech recognition errors. Invalid speech input can be classified into a number of types: stuttering, simple misread links or commands and insufficient length *i.e.* the user did not speak the required three segments.

As expected, users with little computer experience seemed to prefer surfing with speech, while the more experienced users had mixed preferences.

5. CONCLUSION

We have built a prototype system which uses Japanese speech to browse through Japanese World Wide Web pages. The system enables the user to speak the links, to speak a key word to access a bookmarked web page, and use speech commands to control the browser.The system detects and converts three Japanese electronic coding systems into one, and segments the link names into appropriate phrases automatically as each new Web page is visited. Since the number of links on one page is fairly limited, the system shows surprisingly high accuracy. Preliminary tests show that the system understands the over 91% of the speech commands correctly. The system currently runs on a UNIX workstation.

Because of the increased resource requirements due to the addition of Japanese capabilities, the system can benefit from a reduction in the overall resource requirements. It could also use some improvement in the speech models for even higher accuracy. We would also like to conduct further evaluation of the system. We plan to have a real-time demo during the conference.

REFERENCES

[1] C. Hemphill, P. Thrift, and J. Linn, "Speech-Aware Multimedia," *IEEE Multimedia*, vol 3, no. 1 (Spring, 1996).

[2] K. Lunde, Understanding Japanese Information Processing," O'Reilly & Associates Inc., Sebastopol, CA (Sept. 1993).

[3] H. Takahashi, "KAKASI: Kanji Kana Simple Inversion Program," version 2.2.5 (June, 1994). Available from the ftp site at ftp.uwtc.washington.edu.

[4] J. Picone, T. Staples, K. Kondo, and N. Arai, "Kanji to Hiragana Conversion Based on a Length-Constrained N-Gram Analysis," *Texas Instruments Technical Memorandum,* TRDC-TM-93-02, (April, 1993).

[5] Y. H. Kao, C. T. Hemphill, B. J. Wheatley, and P. K. Rajasekaran, "Toward Vocabulary Independent Telephone Speech Recognition," *Proc. ICASSP 94*, pp. I117-120 (Apr. 1994).

Copyright 1997 IEEE