Multimodal Interface with N-best Display Including Candidates of Spoken Word Fragments

Yonggee Jang†, Atsuhiko Kai† and Longbiao Wang†
† Shizuoka University, Hamamatsu 432-8561, Japan
Email: jang@spa.sys.eng.shizuoka.ac.jp, kai@sys.eng.shizuoka.ac.jp, wang@sys.eng.shizuoka.ac.jp

***Abstract*—We introduce a multimodal interface to present multiple candidates for a large-vocabulary spoken-word input task. We also propose a method of presenting multiple candidates, including common word fragments, to prevent a decline in efficiency due to degraded automatic speech recognition (ASR) accuracy when assuming a large vocabulary task. Our system is based on an N-best framework that presents candidate lists generated from the N-best output of an ASR system, and the user chooses one candidate from a list using a graphical user interface. In this paper, we present an interface that users can interact with through a mobile information system, such as smart phone and car navigation systems supporting a Web browser. To improve the input efficiency under a condition of poor recognition accuracy, our proposed method presents not only the N-best recognition candidates but also search candidates that are word fragments. Search candidates are generated automatically from a dictionary and an N-best output of the ASR system, and the fragmental candidates can be used to narrow the candidate list. We introduce a method to determine the optimal proportion of search candidates in the list. Experimental results for the input task of about 13,000 vocabulary words show that the proposed user interface attained higher input efficiency than an interface that presents N-best word candidates only.**

I. Introduction

With the development of speech recognition technology, applications of speech recognition systems are being becoming more widespread. In addition, it is expected that mobile information systems and speech recognition systems will be combined owing to recent developments of the mobile information systems. We consider a speech input interface combined with a graphical user interface (GUI) that displays recognition results. The listing of multiple candidates from the N-best hypotheses in order of likelihood score is a simple and effective solution to the problem of error in speech recognition systems [1,2,3]. However, in a large-vocabulary task, a spoken word often does not appear in a candidate list because of degrading recognition accuracy. As described in Section II, we know that the probability of an input word appearing in the N-best list is not proportional to the number of candidates.

In this paper, to improve the input efficiency of such an N-best framework, we propose a speech interface system that presents search candidates and search options to narrow the range of candidates, together with word candidates. The search candidates are word fragments that are common in the vocabulary of a task. This approach has been shown to improve efficiency in a previous study [1]. However, the previous study focused on the coverage of words using word fragments, and thus, did not consider the display of a limited number of candidates due to the constraint of a display device.

In our previous study, we investigated definitions of word fragments and the generation of candidate lists including search candidates, and found the possibility of improving the input efficiency of the interface system [4]. However, further research is needed to verify the effectiveness, and an interface system has not yet been realized. Therefore, we consider how to extract word fragments and generate a candidate list, and formulate them by considering the user's desires and input efficiency. According to the experimental results, the proposed method has better input efficiency than the method of presenting only recognition candidates. In addition, the proposed method is robust against changes in the number of candidates in the list and contributes to usability.

Section II overviews the proposed speech input interface, and Section III discusses word fragments and a method for generating a candidate list, which are the core concepts of the proposed system. Section IV presents experimental results to compare the performances of the proposed method and a baseline method.

II. Overview of the proposed interface

Listing multiple candidates in order of likelihood score from the N-best automatic speech recognition (ASR) output mentioned above is the simplest way to present multiple candidates [3]. For example, if the system lists candidates on the display, then a user can select a spoken word or move to the next candidate list employing the GUI functionality of an information device. However, this typical method often has a poor input efficiency because the probability of a candidate list containing a spoken word does not increase in proportion to the size of the list.

Table I shows the relationship between the number of candidates displayed and the probability of the list containing the correct word. It is seen that the inclusion probability (recognition accuracy) does not increase even in the case of 100 candidates under a low signal-to-noise (SNR) condition (SNR = 10 dB), and does not increase proportionally with the number of candidates. In other words, a typical method of presenting multiple candidates cannot be expected to improve the input efficiency when recognition accuracy is poor.

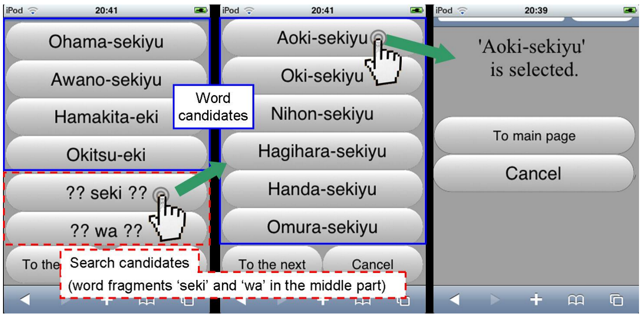
The proposed speech interface is based on the principle of a typical interface that presents multiple candidates; however, it also presents search candidates with the word candidates from the ASR system. The search candidates, which have the form of a word fragment, enable users to narrow the range of candidates and are generated dynamically according

Proceedings of the Second APSIPA Annual Summit and Conference, pages 478–481,
Biopolis, Singapore, 14-17 December 2010.

TABLE I
PERFORMANCE OF THE BASELINE SYSTEM AT DIFFERENT NOISE LEVELS

	SNR = 10 dB	SNR = 15 dB
Number of candidates	Recognition accuracy (Mean number of displays)	
1	45.39% (—-)	83.54% (—-)
10	61.35% (1.00)	96.51% (1.00)
20	64.84% (1.05)	98.25% (1.02)
50	71.57% (1.30)	99.50% (1.05)
100	75.81% (1.65)	99.75% (1.06)

to the ASR N-best information including the confidence score. We describe the definition of word fragments and a method to optionally select and insert search candidates into the candidate list in Section III.



'Aoki-sekiyu'
is selected.

Fig. 1. Example screenshots of the proposed speech interface with a GUI

Fig. 1 shows example screenshots of the proposed speech interface when using a mobile information device. In this case, the user inputs speech and receives a list of candidates on a GUI display. It is then possible to select a spoken word candidate, select a search candidate so that the system presents word candidates that have the same word fragments, or select the next list comprising word candidates and search candidates. In our implementation, the processes of speech recognition and making a candidate list are done on the server side, and the client side only receives the list, as in the case of Google voice search [6] for mobile phones. The interface adopts the standard output format of a Web browser as a GUI; therefore, it is easy to use on information devices such as mobile phones, personal digital assistants, personal computers, and car navigation systems.

III. INCORPORATING SEARCH CANDIDATES INTO THE LIST

A. Definition of search candidates

Search candidates are in the form of word fragments, and a set of all possible search candidates are created in advance from an assumed task vocabulary. We refer to these word fragments as *common word parts*, which are a set of word fragments common to several words in the system's vocabulary. In our previous study [4], we used the beginning and tail parts of words as the common word parts. However, we also use the middle parts of words in this study so that the interface can provide a wide range of choices to users. The user interface requires that the common word parts correspond to meaningful units by themselves, and thus, we define the morpheme[1] as the basic unit of common word parts and extract them from a word dictionary as follows.

1) Execute morphological analysis for each word in the dictionary to divide all the words into morphemes.
2) Extract morphemes that are a common part of three or more different words as their beginning, middle or end.
3) Remove word fragments that are substrings of other fragments and have the same frequency.

On the proposed interface, the system calculates confidence scores of common word parts that are extracted in the processes mentioned above using scores of word candidates from the N-best ASR output, and uses them in selecting an optimal number of search candidates.

B. Selecting search candidates by confidence measure

Our proposed interface system assumes that a GUI is used to show a candidates list and allow a user to select a candidate. The list consists of the word candidates selected from the N-best ASR output and the search candidates selected from a predefined set of common word parts. The effectiveness of the interface varies depending on the proportion of word candidates and search candidates because the number of candidates that can be displayed is limited. In this paper, we denote the size of the candidate list that can be viewed at once as L, which includes N_k word candidates and N_s search candidates (where $L = N_K + N_S$). We propose an algorithm that aims to minimize the expected number of lists that need to be displayed for usability. First, we define the confidence score for common word parts. Given $\mathbf{W}_R = W_1, W_2, \cdots, W_N$, the set of word candidates ranked by the ASR system, a confidence score of a common word part w is computed using the following equation.

$$C(w) = \frac{\sum_{i=1}^{N} P_A(X|W_i)\delta(W_i, w)}{\sum_{i=1}^{N} P_A(X|W_i)}, \tag{1}$$

where $P_A(X|W_i)$ is the acoustic likelihood of the word W_i for the utterance X, and $\delta(W_i, w) = 1$ if the common word part w is a part of the word W_i and $\delta(W_i, w) = 0$ otherwise. After computing confidence scores of word candidates and search candidates, the algorithm chooses a candidate in descending order of score and inserts the candidate into a list.

We consider several ways of determining the ratio of search candidates to word candidates and selecting the candidates themselves. We consider the confidence score of candidates as an estimated a posteriori probability of an uttered word and determine the expected number of candidate lists that need to be displayed for the correct word to appear. For example, a word candidate with a confidence score of 0.5 is considered to have a probability of 0.5 that the input task will finish at once if the word is present in a list. In contrast, a search candidate with a confidence score of 0.5 is considered to have a probability of 0.5 that the interface has to display at least two lists since there may be no uttered word in the narrowed list for a limited size L.

Assuming that the candidate list includes N_K word candidates $\{W_k\}$(k=1,2,$\cdots$,N_K) and N_S search candidates $\{w_s\}$(s=1,2,$\cdots$,N_S), we consider an algorithm to determine the proportions of word candidates and search candidates that

[1] Each word part separated by '-' is a morpheme unit in Fig. 1.

minimize the expected number of lists that will need to be displayed. The expected number can be calculated as

$$S_{count}(\{W_k\},\{w_s\}) = \lambda_1 \sum_{k=1}^{N_K} C(W_k) + \lambda_2 \sum_{s=1}^{N_S} C(w_s) + \lambda_3 (1 - \sum_{k=1}^{N_K} C(W_k) - \sum_{s=1}^{N_S} C(w'_s)). \quad (2)$$

Equation (2) estimates the expected number of displays of the candidate list including N_K word candidates, where $\lambda_1,\ \lambda_2,\ \lambda_3$ are constants that are the expected numbers of displays when selecting a word candidate, search candidate, and the next (list), respectively. While it is natural that $\lambda_1 = 1$, the other constants are determined approximately as $\lambda_2 = 2$ and $\lambda_3 = 4$ in our evaluation experiments. Since finding an optimum candidate list ($\{W_k\}$ and $\{w_s\}$ that minimize S_{count} in Equation (2)) requires too much time, we now describe an algorithm to approximately minimize the expected number of displays in detail.

When generating the t-th ($t \geqq 1$) candidate list to display, we execute the following.

```
1. Set $m = 1$ if $t = 1$, $m$=0 if $t > 1$.
2. For $k = m \sim L$, execute step 3.
3. $CandList(k) \leftarrow \emptyset$.
  (a) $CandList(k) \leftarrow CandList(k) \cup W_i$ for $i = 1, 2, \cdots, k$
      : Add a word candidate.
  (b) $\mathbf{W}_R \leftarrow \mathbf{W}_R - W_i$ : Remove the word candidate selected.
  (c) Repeat the following processes $N_S = (L - k)$ times.
      (c1) Compute confidence score $C(w)$ for all of common
           word parts $w$ corresponding to a part of $\mathbf{W}_R$.
      (c2) $CandList(k) \leftarrow CandList(k) \cup \hat{w}$,
           where $\hat{w} = \underset{w}{\mathrm{argmax}}(C(w))$.
      (c3) $W_R \leftarrow W_R - W_i (\forall W_i \supset \hat{w})$ : Remove word candidates that correspond to the search candidate.
  (d) Calculate $S_{count}(k)$ using Equation (2).
4. Compute $\hat{k} = \underset{k}{\mathrm{argmin}}\, S_{count}(k) (m \leqq k \leqq L)$,
   and choose the candidate list $CandList(\hat{k})$ in which
   $\hat{k}$ word candidates are inserted.
```

Word candidates and search candidates in the list selected by the algorithm above are more likely to contain the spoken word or part of it. Since the search candidates in effect reduce the number of word candidates, the probability of directly finding the spoken word decreases for a fixed number of candidates in the first list displayed, but this decrease is minimized using the introduced algorithm.

IV. Evaluation experiment

A. Task and Data

In this paper, experiments are conducted for an institution search task assuming that a user interacts with a car navigation system. This task has the size of 12,346 vocabulary words, which relate to landmarks, shops, and various institutions in Shizuoka, a Japanese prefecture, and has a variety of common word parts. The common word parts are extracted from a dictionary using the method discussed in Section III-A. As a result, 546 beginning parts, 1262 middle parts and 157 end parts of word fragments are extracted. They account for about 15.92% of words.

The test set of speech used in the experiments is a total of 401 utterances of isolated words spoken by four male speakers. The speech data were recorded in an acoustically clean environment, but environmental noise was added. We made two artificially noisy test sets with SNR = 10 and 15 dB, and employed spectral subtraction for each test set to simulate real conditions. In all experiments in this paper, we used SPOJUS++, an HMM-based continuous speech recognizer for the Japanese language and an HMM acoustic model comprising 124 different categories of syllables and trained with clean speech.

B. Evaluation Method

In evaluation experiments, we executed an offline simulation in which it is assumed that a user is using the speech interface. Specifically, the user utters a word and searchers for the spoken word in the candidate list displayed. The priority in selecting candidates has the order of the spoken word candidate, the search candidate with the highest confidence score, the beginning part of search candidates, the end part of search candidates, and the middle part of search candidates. The user chooses to display the next candidate list if no appropriate candidate exists in the current list. In that case, the interface presents candidates that are not associated with the candidates already displayed.

To evaluate the performance of the interfaces, we consider that the list size L is fixed to 10, and take the success rate and average number of displays as a measure of the performance. The success rate is the probability that a user finds the spoken word directly in the limited list of candidates. Both the cases of selecting a word candidate from the list directly and selecting a search candidate and selecting a word candidate from the narrowed list are regarded as input successes. The average number of displays is the mean value of display counts in the case of input success.

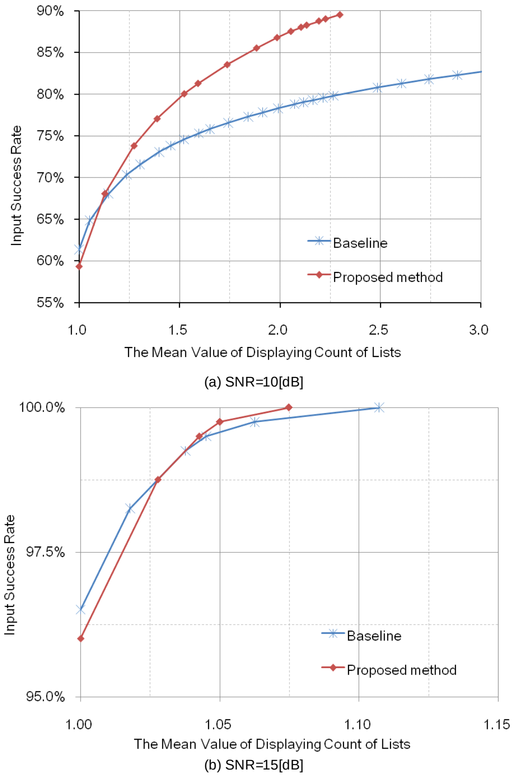
$$S_{count}(Test\ set) = \frac{\sum_{t=1}^{T} Utt(t) \times t}{\sum_{t=1}^{T} Utt(t)}, \quad (3)$$

where T is the maximum number of displays and $Utt(t)$ is the number of utterances for which the input succeeds when the display count is t.

In the simulation, we consider that a user never stops searching for a spoken word until the maximum number of lists are presented. Thus, the average number of displays relates to the success rate, and it is obvious that an interface that requires a smaller number of displays is better if the interface has the same success rate. We compare the performances of interfaces between the traditional method of presenting word candidates only and the proposed method of presenting candidates using the algorithm discussed in Section III-B.

C. Results

First, Table I in Section 2 presents the performance of the baseline. For example, if the length of the candidate list is fixed to 10 and the maximum limited number of lists is set



(a) SNR=10[dB]

(b) SNR=15[dB]

Fig. 2. Performances of the speech interface systems (baseline and proposed method)

to 2, the probability that the user can find a spoken word in the lists is 64.84%. If the maximum limited number of lists increases to 10, the input success rate rises to 75.81%. The display count of 1.65 means that the system displayed 1.65 lists on average in the case that the user found a spoken word in the candidate list. By increasing the maximum number of lists to be displayed, both the success rate and the mean number of displays increase. Therefore, when we consider the horizontal axis as the mean value of the display count and the vertical axis as the input success rate, we can draw a curve that runs upward and to the right. If the gradient of the curve is greater, the interface system has better input efficiency, and thus, it can be considered to perform better.

Fig. 2 presents the results of experiments that compare the performances of baseline and proposed interface systems under the two conditions of SNR = 10 and 15 dB. The figure shows the performance of each interface as a graph, and the horizontal and vertical axes indicate the mean number of displays and the success rate, respectively. Each point on a curve farther from the bottom left corresponds to the performance in the case of increasing T, the maximum number of lists to display.

Fig. 2 shows that the performance of the proposed method is slightly lower than that of the baseline in the first display. This is because the proposed interface inserts search candidates instead of word candidates, which decreases the probability of finding the spoken word directly. However, the performance of the proposed method surpasses that of the baseline as the maximum number of lists to display increases. For example, to obtain an input success rate of 80% for SNR = 10 dB, the baseline needs to display about 2.3 lists but the proposed method requires only about 1.5 displays. Generally, the performance of the proposed method surpasses that of the baseline when the maximum number of lists to display is greater than one, and the difference in the input success rate between the interfaces is a maximum of 9.5% for 2.3 displays.

Under the condition of better recognition accuracy, the difference between the interfaces is not so prominent, but the proposed method still performs better than the baseline when the maximum number of lists to display is greater than one. The maximum difference in the input success rate between the interfaces is 0.5% for SNR = 15 dB. In any case, the performance of the interface using the algorithm that minimizes the expected number of displays is better than an algorithm that maximizes the sum of confidence scores [4].

Finally, we carried out an experiment in which the size of the candidate list was 5 or 20. The trend of the performance curve is similar to the results in Fig. 2, and thus, we consider that the length of the candidate list has little effect on the performance of our system.

V. Conclusions

In this paper, we presented an approach in which word fragments are used as search candidates to improve the input efficiency of a user interface that displays multiple candidates. While the traditional method uses only the word candidates from the N-best output, the proposed method also uses search candidates that enable users to narrow the range of candidates from the recognition result. In addition, we proposed methods to extract common word parts used to search candidates and to generate a candidate list to minimize the expected number of candidates lists that need to be displayed until the input task is accomplished.

According to the results of the experiment, the proposed interface improves the input efficiency compared with the baseline, and the effectiveness was noticeable under a noisy condition assuming a realistic acoustic environment. In addition, using the search candidates, users can achieve partial success using the interface, which is able to give provide the user with more flexible choices.

As future works, we will conduct an experiment to test the proposed interface with real users evaluating the usability of the interface. Furthermore, to further improve input efficiency, we will deal with the problem of unknown words by presenting word fragments.

References

[1] Kitaoka N., Oshikawa H., Nakagawa S., “Multimodal interface for organization name input based on combination of isolated word recognition and continuous base-word recognition,” *Proc. of INTERSPEECH 2005*, pp. 1201-1204, 2005.

[2] Oshikawa H., Kitaoka N., Nakagawa S., “Speech interface for name input based on combination of recognition methods using syllable-based N-gram and word dictionary,” *Proc. ICSLP-2004*, pp. 177-180, 2004.

[3] Cho K., Miyayama A., Yamashita Y., “Determination of the number of candidates using recognition scores for N-best based speech interface,” *IEICE Trans.*, vol. J88-D-II, pp. 1003-1011, 2005, In Japanese.

[4] Yonggee J., Atsuhiko K., Longbiao W., “Speech interface for isolated words based on combination of search candidates from the common word parts,” *Proc. WESPAC X*, Code 0261 (7 pages), 2009.

[5] http://www.google.com/mobile