Input Prediction Method of Speech Front End Processor Using Prosodic Information

Masahiro Araki, Hiroyoshi Ohmiya & Satoshi Kida

Department of Electronics and Information Science
Kyoto Institute of Technology, Japan
araki@dj.kit.ac.jp

Abstract

In general, prosody of speech contains various information. For example, in Japanese, accent information is used for distinguishing homonyms and identifying word boundaries. In this paper, we propose a combination method of phonetic and prosodic information in speech applications, that is, an input prediction front end processor for dictation. From a few morae inputs, completion candidates that are sorted by input history and by the accent pattern are listed up. We examined two accent usage methods for both registered words and unregistered words and implemented an input prediction system combining a speech recognizer, a prediction server and an accent usage module.

1. Introduction

In general, prosody of speech contains various information. For example, in Japanese, accent information is used for distinguishing homonyms and identifying word boundaries. If we can use prosodic information in speech applications, such as dictation systems, the performance of the system is expected to increase in various aspects.

However, accurate extraction of prosodic information from casual speech is a difficult problem. Therefore, it is not appropriate to deal with prosodic information in the same way as phonetic information (the accuracy of which is expected to be above 90%). We need a suitable method of combining prosodic and phonetic information in order to design good speech applications.

In this paper, we propose one such methods of combining phonetic and prosodic information in speech applications. As an application of this accent recognition method, we implemented an input prediction front end processor for dictation. This processor works as a front end of a dictation system. From a few morae input, the processor lists up completion candidates sorted by the input history and accent patterns of each candidate. By using accent information, the processor can narrow down the completion candidates not only in homonyms but also in compound noun phrases. Compound nouns which have the same head noun sometimes show different accent patterns because of combinational accent change rules.

In previous research, Goto *et. al.* proposed a speech completion method which is triggered by filled pause [1]. Their method is useful in case in which the user forgets some part of the input phrase, such as the last name of the musician in a jukebox application. In their method, the set of completion candidates, which is a set of words for target task and domain, are calculated based on the likelihood of speech recognition. Our method is intended for use in a dictation system. Therefore, we have to deal with a larger number of completion candidates than is the case in their method. We use accent information in narrowing down this larger set of completion candidates.

This paper is organized as follows. Section 2 shows the system architecture of our input prediction front end processor for speech. Section 3 explains a narrowing down method of completion candidates using accent information. Section 4 describes system implementation and Section 5 states conclusions and subjects for future study.

2. Architecture of input prediction system

2.1. Input prediction in dictation

Based on recent progress in speech recognition, several dictation systems have been developed [2] and are now in commercial use. However, despite the advantages offered by small computers, such as personal data assistants (PDAs) or mobile phones, which generally do not have a full keyboard, dictation systems are not widely used.

The main reason for this is the unavoidable recognition error in dictation systems. In addition, in Japanese sentence input, there exists a front end processor which can predict input words or phrase through only a few characters of input.

This predictable input method is useful. However, it has two problems. One is that users have to input a few characters using the keypad of a mobile phone or PDA. The other problem is the front end processor sometimes returns a large candidates list as result of commonality of prefix.

In order to deal with such problems, we propose an input prediction front end processor by speech. Speech input is likely to be easier than key input in writing a personal memo or e-mail on small devices. Using speech input, users do not have to enter characters using a small, unwieldy input device. In addition, prosodic information of input speech, such as accent pattern, can help to narrow down the candidates.

We use POBox (Predictive Operation Based On eXample) [3] as a subsystem for making a prediction candidate list from a few phoneme transcriptions. POBox is originally a front end processor for keyboard input, especially for PDAs and mobile phones. POBox incrementally searches candidate set of words according to the key input.

POBox provides its function as a client server model (Figure 1). The POBox client acts as a front end processor for user input and displays a completion candidates list to the user. The POBox server accepts key strokes, searches for candidates in its dictionary and returns a completion candidates list to the client. The client also passes the information to the server as to which candidate is selected. The server reflects this selection information as a user history in the dictionary. In addition, POBox has the ability of ambiguity searching of the dictionary in order to deal with a slight input error by a small keyboard. This property can be used for correcting speech recognition errors.

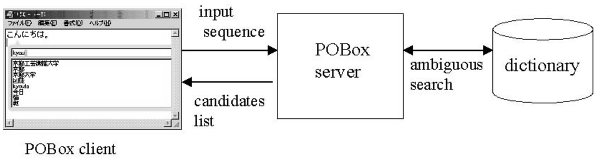


Figure 1: *Client-server model of POBox.*

We replace the POBox client so as to accept speech input. In addition, we implement a prosodic analyzer and re-scoring module for the candidates list using this prosodic information. We use a Japanese large vocabulary continuous speech recognition engine, Julius [2], as a speech recognition module. Julius returns a sequence of roman character lists which can be an input of a POBox server. The overall architecture of our system is illustrated in Figure 2.

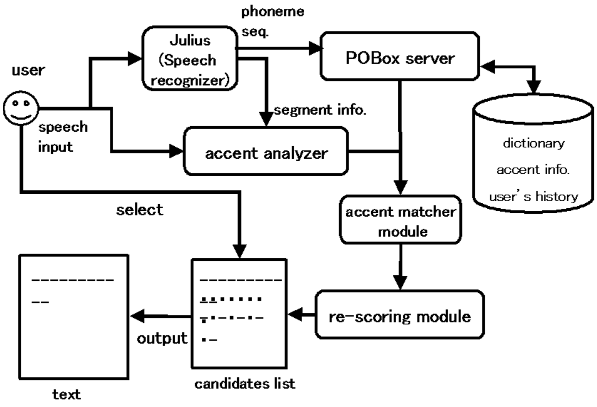


Figure 2: *Architecture of input prediction system.*

2.2. Usage of Japanese accent pattern

The Japanese accent is a pitch accent which has a value of high or low. The value is assigned to each mora by relative difference of fundamental frequency (F_0). Each mora corresponds to one Japanese character in principle (except for a contracted sound). It is widely known that the feature for accent information is observed in larger window size compared to phonetic information [4]. Therefore, we use variable window size, which divides a mora segment into three parts.

In the Japanese accent pattern, the following rules are observed [5]:

- The value of pitch is high or low.
- Each mora has a value of high or low.
- High pitch cannot appear twice in one word.
- The first mora and the second mora must have different pitch values.

Following these rules, Japanese accent type can be classified as type n (where, n is from 0 to N in the case of an N-mora word). "n" indicates a location of dominant downfall in the F_0 contour. For example, Type 0 begins low and changes to high. Type 1 begins high and changes to low. Type n (n>1) begins low and once it has risen high at the nth mora returns to low.

In our application described in the previous subsection, this accent type information can be used to narrow down the completion candidates on the condition that we can use accent type information for each word in the dictionary. However, these accent rules can be applied only to the standardized Japanese or the Tokyo-region dialect read speech. Even in the standardized Japanese corpus, certain utterances do not follow these accent rules. In addition, in western-region dialects, the above mentioned rule of "The first mora and the second mora must have different pitch values." cannot be applied. In these regions, both a /high-high/ pattern and a /low-low/ pattern exist.

Therefore, we use feature level prosodic pattern matching for registered words and accent pattern recognition for unregistered words. Registered words are words that have been uttered previously by a specific user. Each user has an individual dialect and accent pattern. Because of this individual variation, a standard accent pattern dictionary is useless. On the other hand, it is unrealistic for each user to record all the words' pronunciations in the dictionary in his/her manner. Therefore, we use an accent pattern dictionary and the result of accent type recognition for narrowing down the candidates of unregistered words. In the next section, we describe each usage method of accent information.

3. Narrowing down method of completion candidates using accent information

3.1. Accent pattern recognition

In the case of unregistered words, an accent pattern is extracted in order to narrow down candidates. The flow of our accent recognition method is shown in Figure 3.

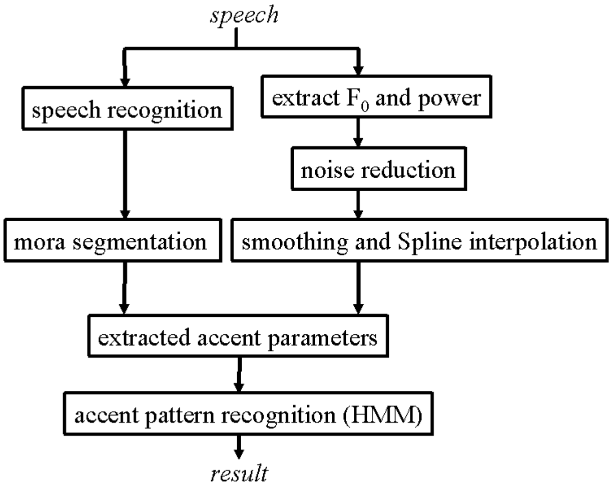

speech
speech recognition
extract F0 and power
noise reduction
mora segmentation
smoothing and Spline interpolation
extracted accent parameters
accent pattern recognition (HMM)
result

Figure 3: *Flow of accent recognition.*

Input speech is recognized by Julius [2], an automatic speech recognizer which outputs a recognized phoneme sequence with alignment information. As for prosodic information, the F_0 parameter is extracted by the auto-correlation method. After the process of noise filtering and modification of overtones, smoothing is performed by median filtering and the least squares method. In addition, in order to acquire a smooth F_0 contour, we use spline interpolation to a voiceless segment. The flow of this feature extraction method is shown in Figure 4.

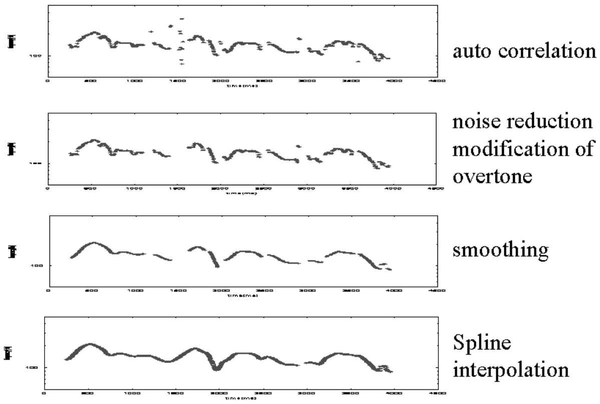


Figure 4: *Extraction of F_0 feature.*

Considering that the transit of prosodic features is somewhat slower than that of phonetic features, a larger unit is required for grasping prosodic features compared to the ordinary window size for phonetic features. However, if we simply widen the window size, the feature extraction can become unstable. Therefore, in our method [6], prosodic features are extracted by frame units, the centers of which are identical to the center of the division of each mora into three parts (Figure 5).

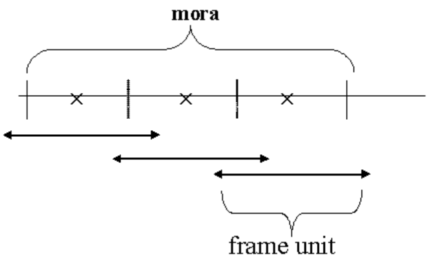


Figure 5: *Frame unit.*

We use HMM as a model of prosodic patterns. The numbers of states for type 0, type 1, and type n (n>1) accent patterns are 7, 7, and 10, respectively. Each state has three Gaussian mixtures. This HMM is trained using 506 sentences read by 40 male speakers and is tested using 50 sentences by 1 male speaker. We examined various lengths of frame units (from 20 ms to 50 ms) because we suspect that there is a tradeoff between feature extraction accuracy and stability of the parameter.

The results are shown in Figure 6. We obtained 66% accuracy of open test for the frame length of 30 ms (i.e. 33% error rate) for the type 0, 1, and n categories (also, 75% accuracy in closed test).

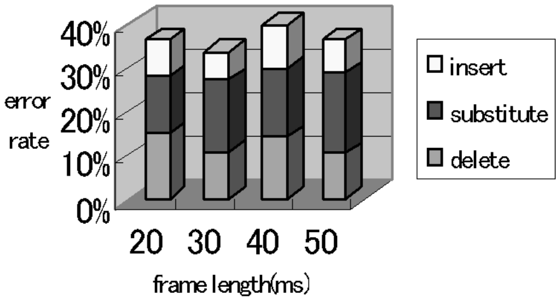


Figure 6: *Results of accent pattern recognition.*

3.2. Accent pattern matching using a neural network

In the case of registered words, accent information is recorded with each user's word dictionary along with frequency and recent usage information. Accent information is used to narrow down the completion candidates and usage information is used to rescore the candidates list.

For the purpose of reduction of dictionary entry size, the accent pattern category information described in the previous subsection is suitable for each entry of accent information. However, accuracy of accent type recognition is not so precise as to use to define a accent type for each word entry in user's dictionary. In addition, accent type recognition of input speech is also unreliable. Therefore, it is not appropriate to use accent type level information in order to filter the completion candidates in the case of registered words.

In order to avoid information loss via the recognition process, we use a representative value for the F_0 contour of each mora (the F_0 value at the end of each mora; it is based on the fact that the prosodic features are delayed with the phonetic features) and its linear regression coefficient (Figure 7). This is a rough approximation of the parameters described in the previous section. Filtering for making completion candidates is performed via neural network in order to deal with individual variations. The inputs of the network are representative value for F_0 contour and linear regression coefficient of the first five morae for registered and input words. The output is a confidence score of the consistency of the accent pattern. The network structure is a feed-forward three-layer type.

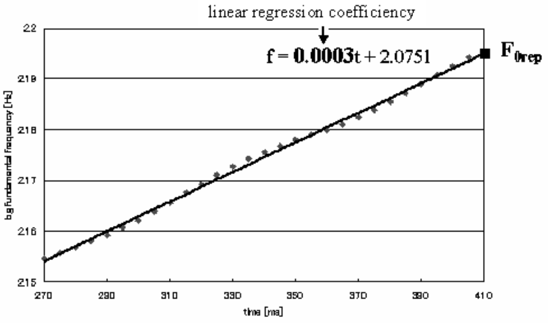


Figure 7: *Parameters for accent pattern matching.*

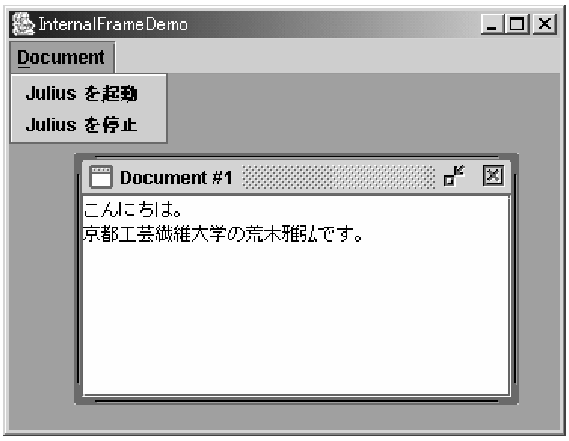
As an evaluation of this accent pattern matching method, we conducted a filtering experiment. We prepare compound nouns which begin with a place name. The words are all typical of the same-phonetic sequence and different-accent word group. For example, “*kyouto daigaku* (Kyoto University)” and “*kyouto* eki (Kyoto Station)” are type 1 accents. On the other hand, “*kyouto kougei sen’i daigaku* (Kyoto Institute of Technology)” is a type 2 accent. We prepares 160 such compound nouns as recorded by five male subjects.

Using this recorded data, we examined a pair-wise accent pattern matching test using the neural network. The threshold value of the output layer, which indicates confidence score of the consistency of the accent pattern for word pair, is optimized through the experiment.

As a result of this experiment, we obtained accent consistency accuracies of 91% for the speaker-closed test and 71% for speaker-open test. This method is assumed to be applicable for speaker dependent applications. Therefore, the accuracy is high enough for filtering completion candidates.

4. System description

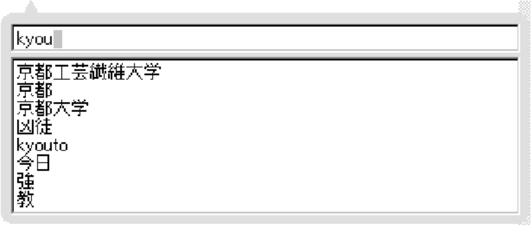
Based on the above methods, we implemented an input prediction front end processor for dictation. We used Julius [2] as a speech recognizer and POBox [3] as a completion server. The front end processor and client are implemented by Java. The input prediction system is shown in Figure 8.



こんにちは。
京都工芸繊維大学の荒木雅弘です。

Figure 8: *Input prediction dictation system.*

Following a user’s speech input, this system shows a completion candidates list (Figure 9). The candidates are ordered by accent information and recent usage (and frequency) information. If the user stops speaking and selects one of candidates, the word is inserted into the text area and the selection history is stored to the user dictionary. If the user ignores the completion information, it disappears as the user continues to input speech.



kyou

京都工芸繊維大学
京都
京都大学
凶徒
kyouto
今日
強
教

Figure 9: *Display of completion candidates list.*

5. Conclusion

In this paper, we propose an input prediction method for dictation using accent information, which can narrow down completion candidates using the results of accent recognition in unregistered words and accent pattern information in registered words. We examined each accuracy experimentally.

Full integration of the proposed method requires a machine readable accent pattern dictionary. However, such a dictionary is not yet publicly available. The developer group of the Japanese morpheme analyzer ChaSen announced a plan to add an accent entry to its dictionary. Our design of treatment of unregistered words assumes such a machine readable accent dictionary.

As future research, we plan to implement this input prediction dictation system on a PDA and examine its availability through e-mail input experiments.

6. References

[1] Goto, M.; Itou, K.; Hayamizu, S., 2002. Speech completion: On-demand completion assistance using filled pauses for speech input interfaces, In *Proc. of ICSLP 2002*, 1489-1492.

[2] Kawahara, T; Lee, A.; Kobayashi, T.; Takeda, K.; Minematsu, N.; Sagayama, S.; Itou, K.; Ito, A.; Yamamoto, M.; Yamada, A.; Utsuro, T.; Shikano, K., 2000. Free software toolkit for Japanese large vocabulary continuous speech recognition. In *Proc. of ICSLP 2000*, 476-479.

[3] Masui, T., 1999. POBox: An Efficient Text Input Method for Handheld and Ubiquitous Computers. In *Proc. of the International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, 289-300.

[4] Hirose, K.; Iwano, K., 2000. Detection of prosodic word boundaries by statistical modeling of mora transitions of fundamental frequency contours and its use for continuous speech recognition, In *Proc. of ICASSP 2000*, 1763-1766.

[5] NHK Broadcasting Culture Research Institute, 1999. *NHK Japanese accent dictionary*, NHK publishing.

[6] Kinoshita, I.; Nishimoto, T.; Araki, M.; Niimi, Y., 2001. Accent pattern recognition using HMM (*in Japanese*), *Technical report on IEICE*, SP2001-140, 37-42.