# THE USE OF ACOUSTICALLY DETECTED FILLED AND SILENT PAUSES IN SPONTANEOUS SPEECH RECOGNITION

*Jun Ogata*[1], *Masataka Goto*[1], *and Katunobu Itou*[2]

[1]National Institute of Advanced Industrial Science and Technology (AIST).
Ibaraki 305-8568, Japan
[2]Hosei University. Tokyo 102-8160, Japan

## ABSTRACT

In recognizing spontaneous speech, the performance of typical speech recognizers tends to be degraded by filled and silent pauses, which are hesitation phenomena frequently occurred in such speech. In this paper, we present a method for improving the performance of a speech recognizer by detecting and handling both filled pauses (lengthened vowels) and silent (unfilled) pauses. Our method automatically detects these pauses by using a bottom-up acoustical analysis in parallel with a typical speech decoding process, and then incorporates the detected results into the decoding process. From the results of experiments conducted using the CIAIR spontaneous speech corpus, the effectiveness of the proposed method was confirmed.

**Index Terms**: spontaneous speech, filled pause, silent pause, acoustic model, language model

## 1. INTRODUCTION

Current state-of-the-art speech recognition systems can achieve high recognition accuracy for read speech or spoken dialogue in limited domains. However, the recognition of spontaneous speech still remains problematic because such speech includes various phenomena such as filled and silent pauses, repairs, hesitations, repetitions, and partial words. To improve the accuracy for spontaneous speech, novel improvement techniques to reduce the recognition errors caused by such phenomena are required.

As a first step toward dealing with such natural phenomena in speech recognizers, we focus on two important phenomena, namely, filled pauses (lengthened vowels) and silent (unfilled) pauses. While filled and silent pauses play an important role in spoken language, for example, in helping a speaker hold a conversational turn and express mental and thinking states [1][2], these pauses tend to cause recognition errors in typical speech recognition systems. For example, a Japanese sentence "*ee- washoku- no resuto-ran' e ...*" (English translation: "er, to a Japanese food restaurant ...") includes three filled pauses (lengthened vowels), which are represented by "-" and underlines. Note that in this paper we use the term "filled pause" for a vowel-lengthening phenomenon such as a filler (e.g., "*ee-*", "*maa-*", and "*ano-*" in Japanese) and a lengthened vowel during a word. In the above Japanese sentence, three types of filled pauses appear (1) as a filler "*ee-*" ("er" in English) (2) at the end of a word "*washoku-*" ("Japanese food" in English), and (3) within a word "*resuto-ran*" ("restaurant" in English). Although acoustical properties of filled pauses are common in most languages, their probable positions in an utterance depend on languages — for example, the latter two types, (2) and (3), are quite popular in Japanese spontaneous speech. On the other hand, the term "silent pause" basically means a temporal region in which a speaker does not utter during a word, phrase, or sentence in spontaneous speech. In particular, during spontaneous conversations (at least in Japanese), relatively long silent pauses tend to be inserted at any positions in an utterance, both within a word as well as between words.

Several previous methods have dealt with filled pauses in speech recognition. In [3], a language model that considers speech disfluency was presented. In [4], filled pauses were incorporated into pronunciation variations for lexical modeling. Most of these approaches assumed that filled pauses can be dealt with as words in a system vocabulary and that a speech recognizer can reliably generate hypotheses containing such pauses as words. However, Stouten *et al.* [5] reported that this assumption is often wrong because it is difficult to predict the occurrences of filled pauses by using acoustic, language, and pronunciation models (lexicons). Therefore, they proposed a method that uses an independent filled-pause detector to reduce speech recognition errors caused by disfluencies. This method [5] first segments the input speech into phoneme-like segments, and then judges whether each segment is a filled pause or not (i.e., detects a filled pause) by using a neural network before speech recognition. All the frames of the detected filled-pause segments are then ignored when decoding, resulting in the avoidance of recognition errors caused by filled pauses. Although this method is effective for fillers such as "uh" and "uhm", it did not take into account a filled pause at the end of a word or within a word, which we deal with by using a bottom-up acoustical analysis without segmenting the input speech. Moreover, this method did not deal with silent pauses at all.

In this paper, we present a robust recognition method that can deal with both filled and silent pauses in spontaneous speech in Japanese. First, two detectors, one for filled pauses and the other for silent pauses, identify the beginning and end times of each pause in the input speech. Then the detected pauses are utilized to control the decoding process in our speech recognizer. Since both pause detectors are based on a bottom-up acoustical signal analysis, our method has two advantages:

- It can handle all types of filled pauses, regardless of where a filled pause occurs in utterances — i.e., as a filler (between words), at the end of a word, or within a word.
- It does not depend on training data, task domains, and languages.

In the following sections, we first present an overview of our method and explain the filled and silent pause detectors. We then present a decoding method using the detection results. Finally, we present the results of several experiments using a large-scale spontaneous speech database and confirm the effectiveness of our method.

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

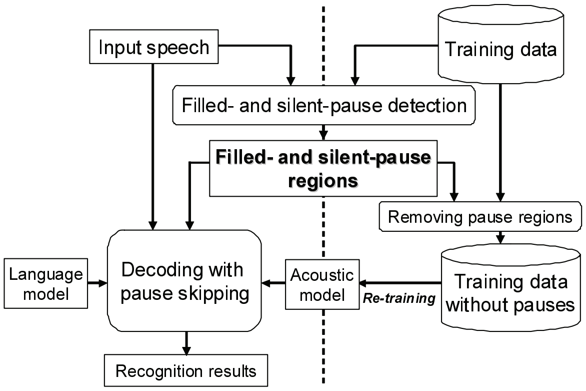

Input speech
Training data
Filled- and silent-pause detection
Filled- and silent-pause regions
Removing pause regions
Language model
Decoding with pause skipping
Acoustic model
Re-training
Training data without pauses
Recognition results

**Fig. 1**. *System overview.*

## 2. SPEECH RECOGNITION BASED ON FILLED AND SILENT PAUSE DETECTION

The basic concept of our method is to control a decoding process on the basis of the output of filled and silent pause detectors. In our method, this modified decoding based on detected pauses is carried out only for utterances including detected pauses, and normal decoding is carried out for the other normal utterances. In this manner, for normal utterances, we can minimize the recognition errors that might be caused by our modified decoding.

### 2.1. System overview

Figure 1 shows the overview of our recognizer. The first step is to use the filled and silent pause detectors to identify the beginning and end times of filled and silent pauses in the audio signals of the input speech. Then, a modified decoding by skipping the detected pauses is conducted to obtain speech recognition results. However, since such a skipping process implies the intentinal manipulation (shortening) of input utterances, it might cause an acoustic mismatch between the testing and training data. In order to reduce the acoustic mismatch and incorporate the acoustic changes caused by the skipping process, we introduce the re-training of acoustic models by using the pause detection. Even in the training data of the acoustic models, our system automatically detects filled and silent pauses in a similar manner. By using the beginning and end times of these pauses, it removes these temporal regions from the acoustic feature sequences (e.g., MFCCs) of the training data. The acoustic model is then re-trained on these sequences without the detected filled and silent pauses.

### 2.2. Filled pause detection method

To detect filled pauses, we use a real-time detection method proposed by Goto *et al.* [2]. The basic idea of this method is to find acoustical features of filled pauses in speech signals by using frequency analysis. If filled pauses (lengthened vowels) are uttered while the speaking process is waiting for the next speech content from the thinking process, a speaker cannot change the articulator parameters during the filled pauses because subsequent utterances have not yet been prepared. Hence, the method assumes that a filled pause contains a continuous voiced sound of an unvaried phoneme, because such a sound is uttered when the vocal cords are vibrated with almost constant articulator parameters (i.e., with a constant vocal-tract shape). This method accordingly detects filled pauses on the basis of the following two features:

1. Small F0 (fundamental frequency) transition
   When the tension of the vocal cords is unvaried under constant articulator parameters, the F0 of the voice remains almost constant.
2. Small spectral envelope deformation
   When the vocal tract shape is unvaried under constant articulator parameters, the spectral envelope forming the formants remains almost constant. When the deformation of the envelope is evaluated, it is necessary to eliminate the air flow's amplitude modulation, since the air flow from the lungs may vary.

The method determines the beginning and end times of each filled pause by finding the above two acoustical features of filled pauses. Experimental results for a Japanese spontaneous speech corpus showed that this method can detect, in real time, filled pauses with a recall rate of 84.9% and a precision rate of 91.5%. For more details, see [2].

### 2.3. Silent pause detection method

For silent pause detection, we can apply several techniques that were originally developed for audio classification and segmentation tasks [6]. In this work, we employ an energy-based speech/silence classifier, i.e., silent pauses are detected based solely on the log energy of the input speech signals.

### 2.4. Decoding method with pause skipping

We introduce a pause skipping mechanism into a typical time-synchronous Viterbi search (baseline recognizer), as shown in Figure 2. When the search process arrives at the beginning time (frame) of a detected pause, all the nodes (HMM states) at this frame are maintained (stored). The search process does not decode anything at each frame during the pause region — i.e., each detected pause is skipped in decoding. Finally the search process is resumed from the end time of the pause region by using the maintained node information.

Note that in the case of silent pauses, an extra operation is required in the decoding process. If all the frames of a silent pause between words are skipped (discarded) completely, the identification of the word boundary in the decoder becomes very difficult, and this may lead to more recognition errors than the baseline recognizer. Therefore, in this work, some frames (i.e., a short silent region) are not discarded but retained for the decoding process. In more detail, the length of each detected silent pause is shortened to a fixed length (0.1 sec. in this paper).

This decoding method might be considered almost equivalent to a method that first removes the temporal regions of pauses directly from the acoustic feature sequences (MFCCs) and then performes recognition by using a regular recognizer without any extension. However, direct handling of the pause information during the decoding process has the advantage of affording a variety of possibilities for further improvements and extensions such as incorporating linguistic context information around each type of pauses into the decoding process, and dynamically controlling decoding parameters (language model scaling factors, word insertion penalties, etc.) in the decoding process.

## 3. EXPERIMENTS

To investigate the effectiveness of our method, we conducted recognition experiments with actual spontaneous speech data.

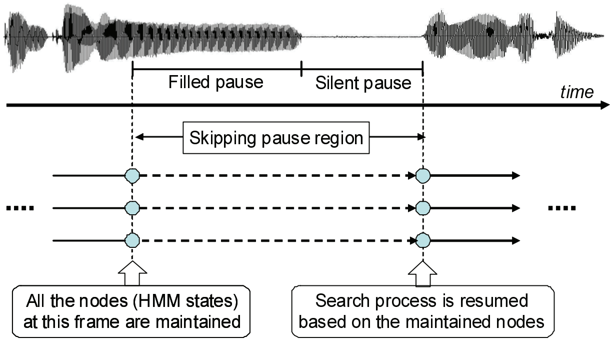

Filled pause
Silent pause
time
Skipping pause region
....
....
All the nodes (HMM states) at this frame are maintained
Search process is resumed based on the maintained nodes

**Fig. 2**. *Decoding with pause skipping.*

### 3.1. Database

We evaluated our method on the CIAIR in-car speech database [7], which includes a large amount of speech data collected in the form of real-world spoken dialogues in a car. This database contains multichannel data recorded from 16 microphones that are placed in various positions, and we used the speech data recorded at the "driver headset" position. Note that this speech data is relatively clean as compared to that from the other channels. Filled and silent pauses occur frequently in the speech data because the CIAIR database consists of utterances of car-navigation dialogues that were recorded while each speaker was driving an actual car in a town. In our evaluation, since utterances in which our filled-pause detector does not detect filled pauses are simply recognized by a regular recognizer without any extension, we excerpted test-set utterances including filled and silent pauses from the CIAIR database (11190 utterances by 101 speakers) as follows. First, the filled-pause detection was performed with all the utterances in the data set. By doing so, filled pauses were detected for 1658 utterances, and therefore, we set aside these utterances as the test set. Note that these 1658 utterances also include silent pauses.

### 3.2. Baseline recognizer

The baseline recognizer was constructed on the CIAIR database. For the training data of acoustic modeling, we used 79093 utterances by 401 drivers. The speech recognizer uses a Japanese syllable-based HMM [9] that consists of 245 acoustic units. This acoustic model has no context dependencies across the acoustic units (i.e., mono-syllable). For the training data of the language model, we used the transcriptions of the spoken dialogue (94306 sentences). A bigram language model was trained on this text data. The vocabulary size was 6528.

The conventional way of handling pauses was incorporated in the baseline language model. With regard to handling filled pauses, Japanese fillers were modeled in the language model by using transcriptions from the CIAIR. On the other hand, word-end silent pauses were modeled in the lexicon by adding a short-pause (SP) phone at the end of every pronunciation.

### 3.3. Experimental results and discussion

The experimental results of our proposed method are summarized in Figure 3. In this figure, "Baseline" indicates the word accuracy of the baseline recognizer, "FP" (filled pause) and "SP" (silent pause) indicate the word accuracy of the decoding method wherein each type of the detected pauses was incorporated independently, and "FP+SP" (filled and silent pauses) indicates the word accuracy of the decoding method wherein both types of the detected pauses were incorporated. We also show the recognition performance in which the acoustic model was re-trained by using the training data after skipping the detected pauses. In the following sections, we show the evaluation of each method individually and discuss the performance.

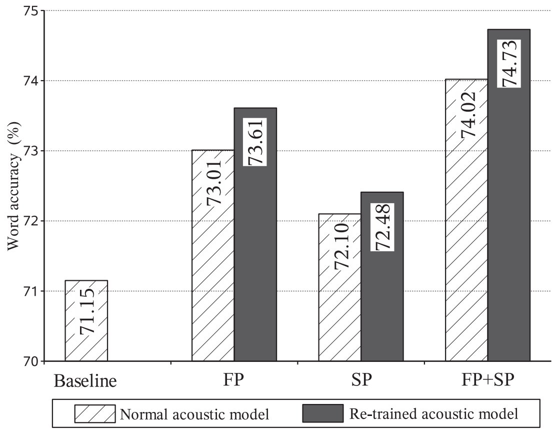


**Fig. 3**. *Performance improvement of our proposed method (FP + SP).*

#### 3.3.1. Evaluation of recognition with filled-pause detection

First, we compare the recognition performance using the filled-pause detector (FP) with the baseline recognition (Baseline). As shown in Figure 3, the word accuracy was improved (71.15% ⇒ 73.01%), and it was found that filled-pause detection is helpful for spontaneous speech recognition. In addition, acoustic-model re-training achieved a further improvement (73.01% ⇒ 73.61%). By investigating the recognition results in detail, we found that larger improvements were obtained for word-end filled pauses (e.g., "*washoku-*") and word-internal filled pauses (e.g., "*resuto-ran*").

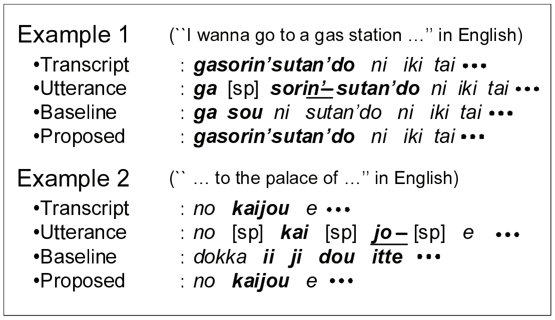
In spoken dialogue tasks such as those in this experiment, filled pauses tend to occur in the case of any proper nouns (e.g., shop name, place name, and product name) uttered by a speaker. Therefore, it is very difficult to expect the occurrences of such filled pauses even if we train a language model or a lexical model. Furthermore, our method could suppress insertion errors for filler words that have a longer duration (e.g., "*ee-*" and "*ano-*").

#### 3.3.2. Evaluation of recognition with silent-pause detection

Next, we compare the performance of the recognition using the silent-pause detector (SP) with the baseline recognition (Baseline). As shown in Figure 3, the word accuracy was improved (71.15% ⇒ 72.48%) by the silent-pause detection. By using the proposed method, we confirmed larger improvements, especially for within-word silent pauses. The following are examples of utterances including a silent pause (denoted as "[sp]") that were not correctly recognized by the baseline recognizer and resulted in two or three incorrect word fragments:

- Utterance where "*hoteru*" ("hotel" in English) was uttered as "*ho* [sp] *teru*"
- Utterance where "*imaikeshiten'*" ("Imaike branch" in English) was uttered as "*imai* [sp] *keshiten'*"

Our method was able to correctly recognize both of these examples (i.e., the performance was improved in these cases). In a real



Example 1 (``I wanna go to a gas station …'' in English)

- Transcript : ***gasorin'sutan'do*** *ni iki tai* •••
- Utterance : ***ga*** [sp] ***sorin'-sutan'do*** *ni iki tai* •••
- Baseline : ***ga sou*** *ni sutan'do ni iki tai* •••
- Proposed : ***gasorin'sutan'do*** *ni iki tai* •••

Example 2 (`` … to the palace of …'' in English)

- Transcript : *no* ***kaijou*** *e* •••
- Utterance : *no* [sp] ***kai*** [sp] ***jo –*** [sp] *e* •••
- Baseline : *dokka* ***ii ji dou itte*** •••
- Proposed : *no* ***kaijou*** *e* •••

**Fig. 4**. *Examples of improvements obtained by our proposed method. "Transcript" shows a correct word sequence, "Utterance" shows an actual utterance with filled pauses ("-" with underlines) and silent pauses [sp], "Baseline" shows a recognition result by the baseline method without any extension, and "Proposed" shows a recognition result obtained by our proposed method by skipping both filled and silent pauses.*

world spoken dialogue, since a speaker tends to utter while thinking, such silent pauses occur even within a word. In particular, "syllable-stressed speech" [8] that occurs in making error recovery utterances tends to include a silent pause within a word. It is therefore important to improve the performance for such utterances.

Note that it was generally difficult to improve the performance on a silent pause between words because the lexicon used in the baseline language model has already dealt with such a silent pause. When noise sounds such as breath, tongue clicking, and small in-car noise were observed during long silent pauses between words, however, they caused some insertion errors with the baseline recognition. Our method with the energy-based detector correctly ignored most of these small noise sounds below a threshold and improved the performance.

### *3.3.3. Evaluation of recognition with filled and silent-pause detectors*

Finally, we describe the recognition results of the filled and silent pause detectors (FP+SP). As shown in Figure 3, the improvements obtained by using both detectors together were better than those obtained by using each detector alone. As compared to the baseline recognition (Baseline), it eventually increased the word accuracy from 71.15% to 74.73%. This suggests that both filled and silent pauses actually occurred in spontaneous speech or dialogues such as those in the CIAIR database. Both filled and silent pauses can sometimes occur within just one word. Figure 4 shows examples of such utterances that were actually observed in our experiment. In Example 1, an utterance of "*gasorin'stan'do*"("gas station" in English) includes a silent pause "*ga* [sp] *sori*" and a filled pause "*n'-*" within this word. In Example 2, an utterance of "*kaijou*" ("venue" in English) includes a silent pause "*kai* [sp] *jo-*" within this word and a filled pause "*jo-*" at the end of this word. In addition, two other silent pauses also occurred before and after the word "*kaijou*". As shown in this figure, the results of the baseline recognition included insertion errors, but those of our proposed method showed significant improvements for these difficult utterances. We thus confirmed the effectiveness of dealing with both filled and silent pauses in the decoding process.

## 4. CONCLUSIONS

In this paper we presented a decoding method that can reduce recognition errors caused by filled and silent pauses in spontaneous speech. These two types of pauses are detected independently by using their acoustical features, and the detection results are used to control the decoding process so that it can skip such pauses. Our proposed method achieved significant improvements on a spontaneous dialogue speech corpus.

Future work will include improvements in our decoding method and further evaluations using other tasks or corpora. Since we used a context-independent acoustic model (syllable model) in the experiments, we will extend the decoding method to handle a context-dependent acoustic model, which would result in a higher performance. Furthermore, we plan to incorporate the decoding method into our public web service, *"PodCastle"* [10][11], that provides full-text searching of podcasts on the basis of automatic speech recognition. Since spontaneous speech in podcasts covers a wider variety of speech data and often includes filled and silent pauses, the decoding method proposed in this paper must be effective.

## 5. ACKNOWLEDGEMENTS

This work was partially supported by Grant-in-Aid for Scientific Research (B) (KAKENHI 19300065).

## 6. REFERENCES

[1] E. Sheriberg: "To 'errrr' is human: ecology and acoustics of speech disfluencies", *Journal of the International Phonetic Association*, vol.31, no.1,pp.153-169, 2001.

[2] M. Goto, K. Itou, and S. Hayamizu: "A real-time filled pause detection system for Spontaneous Speech Recognition", In *Proc. of Eurospeech 1999*, pp.227-230, 1999.

[3] A. Stolcke and E. Shriberg: "Statistical language modeling for speech disfluency", In *Proc. of ICASSP '96*, pp.405-408, 1996.

[4] H. Schamm, *et al.*: "Filled pause modeling for medical transcriptions", In *Proc. of SSPR*, pp.143-146, 2003.

[5] F. Stouten, *et al.*: "Coping with disfluencies in spontaneous speech recognition: Acoustic detection and linguistic context manipulation", *Speech Communication*, 48, 1590-1606, 2006.

[6] D. Wang, *et al.*: "Speech segmentation without speech recognition", In *Proc. of ICASSP 2003*, pp.468-471, 2003.

[7] K. Takeda, *et al.*: "Construction and Evaluation of a Large in-car Speech Corpus," *IEICE Transactions on Information and Systems*, Vol. E88-D, No.3, pp.553-561, 2005.

[8] K. Okuda, T. Matsui, S. Nakamura: "Toward the creation of acoustic models for stressed Japanese speech", In *Proc. of* Eurospeech 2001, pp.1653-1656, 2001.

[9] J. Ogata and Y. Ariki: "Syllable-based acoustic modeling for Japanese spontaneous speech recognition", In *Proc. of Eurospeech 2003*, pp.2513-2516, 2003.

[10] M. Goto, J. Ogata, and K. Eto: "PodCastle: A web 2.0 approach to speech recognition research", In *Proc. of Interspeech 2007*, pp.2397-2400, 2007.

[11] J. Ogata, M. Goto, and K. Eto: "Automatic transcription for a web 2.0 service to search podcasts", In *Proc. of Interspeech 2007*, pp.2617-2620, 2007.