PAPER

Posteriori Restoration of Turn-Taking and ASR Results for Incorrectly Segmented Utterances*

Kazunori KOMATANI†a), *Member*, Naoki HOTTA††, Satoshi SATO††, *Nonmembers*, *and* Mikio NAKANO†††, *Member*

SUMMARY Appropriate turn-taking is important in spoken dialogue systems as well as generating correct responses. Especially if the dialogue features quick responses, a user utterance is often incorrectly segmented due to short pauses within it by voice activity detection (VAD). Incorrectly segmented utterances cause problems both in the automatic speech recognition (ASR) results and turn-taking: i.e., an incorrect VAD result leads to ASR errors and causes the system to start responding though the user is still speaking. We develop a method that performs a posteriori restoration for incorrectly segmented utterances and implement it as a plug-in for the MMDAgent open-source software. A crucial part of the method is to classify whether the restoration is required or not. We cast it as a binary classification problem of detecting originally single utterances from pairs of utterance fragments. Various features are used representing timing, prosody, and ASR result information. Experiments show that the proposed method outperformed a baseline with manually-selected features by 4.8% and 3.9% in cross-domain evaluations with two domains. More detailed analysis revealed that the dominant and domain-independent features were utterance intervals and results from the Gaussian mixture model (GMM).

key words: *spoken dialogue system, VAD error, turn taking, a posteriori restoration*

1. Introduction

Appropriate turn-taking as well as generating correct responses is imperative in spoken dialogue systems. Turn-taking generally denotes that two people are talking alternatively. A spoken dialogue system needs to know when to start responding and when to terminate the response. When we assume that the user has initiative to control the system, the system should

1. reply as quickly as possible when the user finishes speaking,
2. terminate its response if the user starts speaking, and
3. not start speaking while the user is speaking.

Simple examples of inappropriate turn-taking are depicted in Fig. 1. The upper part depicts an example in which the system does not reply after the user finishes speaking. Since appropriate feedback is very important from the viewpoint of human-computer interface (HCI) design [3], the system should start responding as soon as each user utterance ends. Meanwhile, the lower part depicts an example in which the system erroneously starts responding during a user utterance ("Please tell me about the Carneros Inn in Napa"). This is called a *false cut-in* and typically occurs when a short pause exists due to disfluency, stammering, etc., within a user utterance.

The error depicted in the lower part of Fig. 1 is caused when a voice activity detection (VAD) error occurs: the user utterance is divided into two fragments by the short pause in the middle and the system accordingly starts responding to the first fragment. This phenomenon, called the *incorrect segmentation* of user utterances, causes two problems:

- the system starts speaking while a user is still speaking, and
- automatic speech recognition (ASR) fails for the wrong VAD results.

Here, we assume VAD results as equivalent to the results of end-point detection, which determines when a user utterance starts and ends. This is because end-pointing is basically performed on the basis of VAD results.

We develop a method that performs *a posteriori restoration*. "A posteriori" means the restoration is performed after an incorrect segmentation is detected, while most existing studies have tried to improve the segmentation accuracy itself (explained in Sect. 5). For the former problem, we add rules on the MMDAgent toolkit [4] to terminate the system utterance as soon as the user starts speaking, which was presented in our previous report [1]. For the latter, we try to restore such incorrectly segmented utterances. This approach can reduce wrong system responses caused by incorrect ASR results, but will delay the start of system

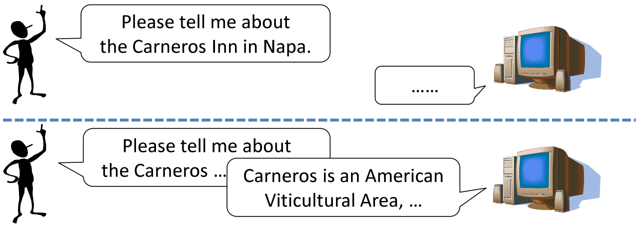
Please tell me about the Carneros Inn in Napa.
……
Please tell me about the Carneros …
Carneros is an American Viticultural Area, …

Fig. 1 Examples of inappropriate turn-taking.

Manuscript received January 19, 2015.
Manuscript revised June 6, 2015.
Manuscript publicized July 14, 2015.

[†]The author is with The Institute of Scientific and Industrial Research, Osaka University, Ibaraki-shi, 567–0047 Japan.
[††]The authors are with Graduate School of Engineering, Nagoya University, Nagoya-shi, 464–8603 Japan.
[†††]The author is with Honda Research Institute Japan, Co., Ltd., Wako-shi, 351–0188 Japan.

[*]This paper is a modified and extended version of our earlier reports [1], [2].

a) E-mail: komatani@sanken.osaka-u.ac.jp

DOI: 10.1587/transinf.2015EDP7014

responses during the restoration process. To compensate it, our demo system produces fillers during the delay [1].

The crucial part of this method is to detect incorrect segmentation; that is, to classify whether or not the restoration is required. In this paper, we improve the accuracy for pairs of utterance fragments. We cast this as a binary classification problem and perform decision tree learning with various features. The features are extracted from pairs of utterance fragments and represent timing, prosody, and ASR result information. To ensure use across various domains, the features should not be dependent on any specific domain. We thus perform two kinds of feature selection to obtain effective and domain-independent features for improving the classification accuracy.

2. Problems Caused by Incorrect Utterance Segmentation

The timing when the system starts responding is determined on the basis of VAD (and resulting end-pointing) results. A VAD module generally distinguishes voices from silences on the basis of the amplitude of the target speech signals and zero crossing rates [5]. User utterances are regarded as having ended when duration of the silence exceeds a threshold. This threshold needs to be set smaller to ensure that the system can respond quickly enough. Responses with latency make users think their utterance has been rejected, which may make them repeat it again. This should be avoided from the viewpoint of user interface design [3].

VAD errors occur often, especially when users make short pauses within utterances due to breathing or thinking about what to say next. The threshold for the silence duration is a parameter in the VAD module. When the threshold is set smaller, it becomes more difficult to determine whether the user has actually finished an utterance or intends to continue it. As a result, a user utterance is more likely to be incorrectly segmented into fragments. This leads to ASR errors because ASR is performed for such incomplete fragments.

An example of incorrect segmentation is illustrated in Fig. 2. The user intends to say "I want to go to Yagoto Nisseki station", but there is a short pause between "Yagoto" and "Nisseki". "Yagoto Nisseki" ("Yagoto Red Cross Hospital") is a station name in Nagoya, Japan. The system should respond with "I will show you the way to Yagoto Nisseki" after the whole user utterance. However, the ASR result is incorrect because ASR is performed separately for two fragments: "Yagoto" and "Nisseki Eki ni ikitai". Here, the keyword "Yagoto Nisseki" is divided into two fragments. Consequently, the system responds with "I will show you the way to Nisshin station" (Nisshin is another station) on the basis of the incorrect ASR result for the second fragment. This does not match the user's request. The ASR result is incorrect because the word fragment "Nisseki" is not in the system's dictionary. Although another approach to dealing with such ASR errors is to add shorter subwords corresponding to utterance fragments into the ASR dictionary, as Jan et al. [6] and Katsumaru et al. [7] have done, this would degrade ASR accuracy because too many subwords would be added into the ASR dictionary. We therefore adopt an approach to restore incorrect segmentation.

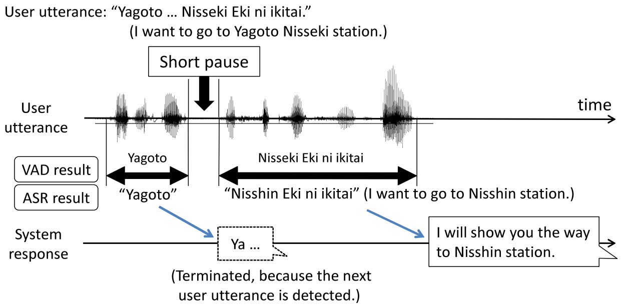
User utterance: "Yagoto … Nisseki Eki ni ikitai."
(I want to go to Yagoto Nisseki station.)
Short pause
User utterance
time
VAD result
ASR result
Yagoto
Nisseki Eki ni ikitai
"Yagoto"
"Nisshin Eki ni ikitai" (I want to go to Nisshin station.)
System response
Ya …
I will show you the way to Nisshin station.
(Terminated, because the next user utterance is detected.)

Fig. 2 Incorrect segmentation of user utterance due to short pause.

The other problem caused by the incorrect segmentation is inappropriate turn taking. This means that the system starts responding while the user is still speaking. Basically, spoken dialogue systems start responding when they receive an ASR result. ASR is performed for speech segments, which are obtained from VAD. If the VAD module incorrectly segments a user utterance into fragments, ASR results are obtained for each fragment, and the system starts responding on the basis of the ASR results for each fragment. Since the first fragment is only a part of the original user utterance, the system erroneously starts responding during the user utterance.

3. Restoring Incorrect Segmentation

Our posteriori restoration process consists of two steps:

1. Determine whether a pair of utterance fragments resulted from an incorrect segmentation or not.
2. Concatenate the utterance fragments if they are incorrectly segmented.

An outline of the proposed method is shown in Fig. 3. Here, a user utterance is segmented into a pair of utterance fragments, denoted hereafter as the first and second fragments. Given a pair of utterance fragments, the system de-

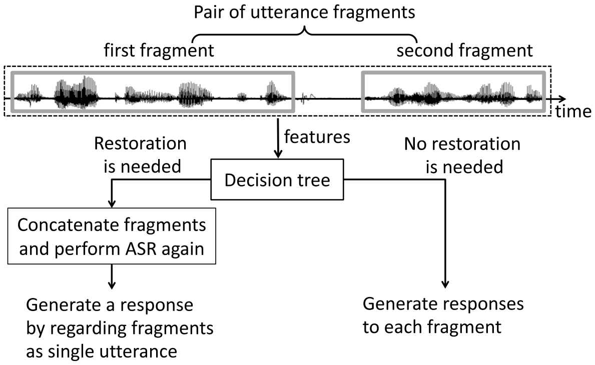
Pair of utterance fragments
first fragment
second fragment
time
features
Restoration is needed
Decision tree
No restoration is needed
Concatenate fragments and perform ASR again
Generate a response by regarding fragments as single utterance
Generate responses to each fragment

Fig. 3 Overview of proposed method.

termines whether the fragments should be interpreted after concatenating them or separately. This is equal to classifying whether the fragment pair was originally a single utterance or not. If the fragments are deemed to be parts of one utterance, the system does not start speaking and performs ASR again after concatenating the fragments in order to restore turn-taking and the ASR results, which have been erroneous due to incorrect segmentation. If the fragments are deemed to be two utterances, the system responds normally; that is, it generates responses based on the ASR results for each fragment. For example, since our current system [1] simply responds to each user utterance independently and it allows barge-ins, its response for the first fragment is immediately terminated and that for the second one is only generated, as illustrated in Fig. 2.

In this problem setting, a VAD parameter that may cause incorrect segmentation is used. This is to secure quick responses after user utterances, as already stated. A safer VAD parameter can reduce incorrect segmentation, but will make the start of system responses slower, accordingly.

3.1 Target Binary Labels

The system needs to determine whether the restoration is required or not. That is, when given a pair of fragments, the system determines whether the pair should be interpreted by concatenating them or separately. Restoration is required when the fragments were originally a single utterance.

We manually annotated each fragment pair with labels indicating if it was originally a single utterance or not. Since the pairs were automatically obtained from VAD results, the data set contains various sounds that are not actually user utterances, such as coughs, wind noise, the system's synthesized voices, etc.

Figure 4 shows examples in which fragment pairs are originally single utterances. At the top is an example of a user wanting to say the long keyword (a point-of-interest (POI) name) "Santa Maria delle Grazie", which is a world heritage site in Italy. However, the user pauses slightly in the middle of the word and the utterance is thus segmented. In this case, ASR always fails because such word fragments are not in the system's dictionary. At the bottom is a user saying "I'd like to know how much lunch costs". These fragment pairs should be concatenated.

Figure 5 shows examples in which fragment pairs are not single utterances. At the top is a fragment pair where the first fragment is a filler. This pair does not have to be concatenated because the first fragment has no content to be conveyed to the system. This is the same for a fragment pair including either noise or the system's synthesized voice. At the bottom, the user's intentions (dialogue acts) are different for each fragment. Those of the first and second fragments are to delete search conditions for stations and foods, respectively. These fragments should not be interpreted by concatenating them[†].

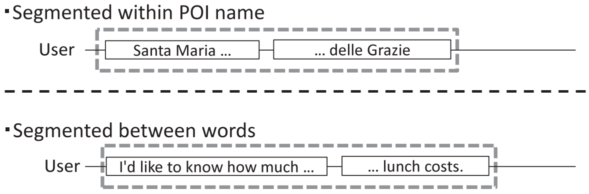
・Segmented within POI name
User
Santa Maria ...
... delle Grazie
・Segmented between words
User
I'd like to know how much ...
... lunch costs.

Fig. 4 Examples of pairs that are originally single utterances.

3.2 Classification by Decision Trees

3.2.1 Features

We perform decision tree learning for this binary classification problem. Here, decision tree learning is used because of interpretability of the obtained results and behaviors of the features. Use of other classifiers such as SVM is in our future work. Decision trees are built by J48 with its default parameter in machine learning software Weka[††].

In total, 18 features are used: eight from the ASR engine, five of timing, and five of prosody (Table 1). These are explained below with a focus on the five features, marked in bold, that were effective in our experimental results described in Sect. 4.3. The numbering before each feature name corresponds to that in Table 1.

- Features from ASR engine: (1)–(8)
  We use (1) the confidence score of the ASR result for the first fragment. The confidence scores are obtained per word from ASR engine Julius. The idea here is that an incorrectly-segmented utterance tends to have a low confidence score, especially when a word is incorrectly segmented within it. We also use (5) noise detection results by Gaussian mixture models (GMMs) constructed by Lee et al. [8]. This model classifies utterances into five classes: "adult speech", "child speech", "laughter", "coughing", and "others". Feature (5) has a binary value, "user utterance" and "noise" by summarizing the GMM results; it is "user utterance" if the GMM-based noise detection results of both fragments are "adult speech" or "child speech".
- Timing features: (9)–(13)
  We define (9) an interval between fragments as the time between the end of the first fragment and the start of the

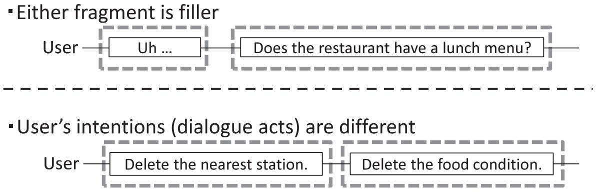
・Either fragment is filler
User
Uh ...
Does the restaurant have a lunch menu?
・User's intentions (dialogue acts) are different
User
Delete the nearest station.
Delete the food condition.

Fig. 5 Examples of pairs that are not single utterances.

[†] In this case, the system needs to respond to the two utterances. We here focus on whether these need to be restored or not, but the way the system should respond in this case is a remaining issue.

[††] http://www.cs.waikato.ac.nz/ml/weka/

Table 1 Eighteen features used for decision tree learning.

Features from ASR engine	Timing features	Prosodic features
(1) Average confidence score of first fragment	**(9) Interval between fragments**	(14) Volume change in final part of first fragment
(2) Confidence score of last word of first fragment	(10) Duration of tail silence in first fragment	(15) Frequency gradient in first vowel of first fragment
(3) Language model (LM) score of first fragment	(11) Duration of head silence in second fragment	**(16) F0 range of first fragment**
(4) Acoustic model (AM) score of first fragment	(12) Duration of first fragment	**(17) Maximum loudness in first fragment**
(5) Noise detection results by GMM	(13) Duration of final syllable of first fragment	(18) Maximum loudness in second fragment
(6) Overlap ratio of phoneme bigrams		
(7) Number of fillers in first fragment		
(8) Number of fillers in second fragment		

Bold: Effective features

second. The idea here is that an originally-single utterance tends to have a shorter interval, as short pauses within utterances due to disfluency are shorter than intervals when a user's utterances actually end. This tendency will be confirmed in Sect. 4.2.1.

- Prosodic features: (14)–(18)
  We exploited five features that may be effective for our task by referring to previous work on turn-taking decision [10]–[12]. There might be other features that are effective, but exploring such features is among the future work.
  We used openSMILE[†] to obtain the prosodic features. We use (16) the F0 range of the first fragment for detecting noises with no harmonic structure. We also use (17) the maximum loudness in the first fragment to help detect the system's synthesized voices, which are unintentionally mixed into the microphones and tend to be low loudness because the microphone was placed near the users.

3.2.2 Two Kinds of Feature Selection

As stated earlier, the features used in the decision tree need to be effective also in other domains. We thus perform two kinds of feature selection:

1. Excluding features having a negative influence
2. Selection of domain-independent features

We first exclude features having a negative influence on classification accuracy. More specifically, we build a decision tree by removing a feature one by one and compare its classification accuracy with the original one with all features. If the accuracy does not degrade without the feature, it is removed because it does not contribute to the accuracy.

Features that are independent of domains are also selected with dialogue data in two domains: restaurant and world heritage. Their details will be explained in Sect. 4.1. We first build decision trees for both domains by ten-fold cross validation. If a feature is used in both the decision trees, that means it is effective in both domains and we regard it as not being dependent on either domain. We select such features as domain-independent ones.

3.3 Restoring ASR Results

After the system detects that a pair of utterances was incorrectly segmented, their wav files are concatenated and ASR is performed again for the concatenated wav file. Specifically, the system saves the segments corresponding to every VAD result to wav files. The margins attached to the speech segments during VAD are then removed. The margins can cause a mismatch with the corresponding entry in the system's dictionary because a short pause may occur within a long keyword. Thus, the speech segments corresponding to the end silence (denoted as phoneme `silE`) of the previous utterance and the beginning silence (`silB`) of the following utterance are removed. The duration of the silences is obtained from phoneme alignment results in ASR. Then, the system concatenates the wav files and performs ASR for it.

4. Experimental Evaluation

4.1 Target Data

We used dialogue data in two domains: restaurants and world heritage [9]. Data were collected by our spoken dialogue systems that search databases of the two domains. In the both domains, recruited participants used the systems to obtain information. The numbers of participants were 30 and 40 respectively for the restaurants and world heritage domains. Each participant talked with either system for eight minutes at most and repeated it four times. We first obtained VAD results by Julius for the wav files recorded for the whole dialogue sessions. The VAD module of Julius detects a speech segment when its frames whose amplitude levels exceed its threshold have a higher zero-crossing rate (ZCR) than its threshold. The module then attaches margins at the start and end of speech segments. We set the parameters for VAD to be almost the same conditions as those of the data collection. The ZCR was set to its default value (60) and the level thresholds of speech input detection (`-lv`) were set to 500 and 1,200 in range of 16 bits for the restaurant and world heritage domains, respectively. The margin lengths were set to 300 and 240 milliseconds at the start and end of speech segments, which were specified by `-headmargin` and `-tailmargin`, respectively. The latter corresponds to the threshold for pause duration in VAD. We obtained 6,615 and 6,593 VAD results in the two domains, parts of which

[†] http://opensmile.sourceforge.net/

Table 2 Target data.

Domains	Restaurant	World Heritage
No. of dialogues	120	156
No. of VAD results	6,615	6,593
No. of target fragment pairs	376	444
No. of target fragment pairs (excluding self-repairs)	255	354

(1,564 and 1,905) were short noise segments.

Our target is pairs of utterance fragments likely to require the restoration because our method does nothing for the pairs that require no restoration. Thus, we selected pairs of VAD results (possible utterance fragments) close in time. We specifically selected fragment pairs whose intervals are shorter than 2,000 milliseconds and each fragment is longer than 800 milliseconds; the latter is to exclude short noises. This condition reflects the fact that we had rejected VAD results shorter than 800 milliseconds when the data were collected[†]. We also make data sets in which self-repairs are manually excluded in advance, as we think self-repairs are different phenomena from our target and should be detected by other features[††]. We have actually found in a preliminary experiment that self-repairs can be automatically excluded with a precision of 70%–90% by using the overlap ratio of phoneme bigrams between the fragments, i.e., how many phonemes are commonly included.

Overall, we obtained 376 and 444 pairs of VAD results in the restaurant and world heritage domains, respectively. After excluding self-repair utterances, there are 255 and 354 pairs. These are summarized in Table 2. After the manual annotation, the numbers of originally single utterances are 156 (61.2%) and 270 (76.3%) out of the 255 and 354 pairs in the two domains, respectively.

4.2 Preliminary Analysis

We preliminarily checked the distribution of intervals between pairs of VAD results. We also checked the effect of ASR restoration to confirm whether utterance understanding accuracy improves or not when incorrectly segmented pairs are restored. In this section, we used the data set before excluding self-repair utterances (376 fragment pairs) in the restaurant domain.

4.2.1 Distribution of Utterance Interval

We analyzed the utterance intervals of the 376 pairs of VAD results in the restaurant domain. We manually annotated whether each pair was originally a single utterance after listening to the wav files, and classify them according to their intervals.

The results are shown in Fig. 6. The vertical axis shows the occurrence of pairs of VAD results, and the horizontal axis shows the millisecond groupings. The figure shows that the shorter interval groupings included more pairs that were originally single utterances. The longer interval groupings included more independent utterances and noise segments. This result shows the utterance interval is an important indicator of whether an utterance fragment pair is originally a single utterance or not.

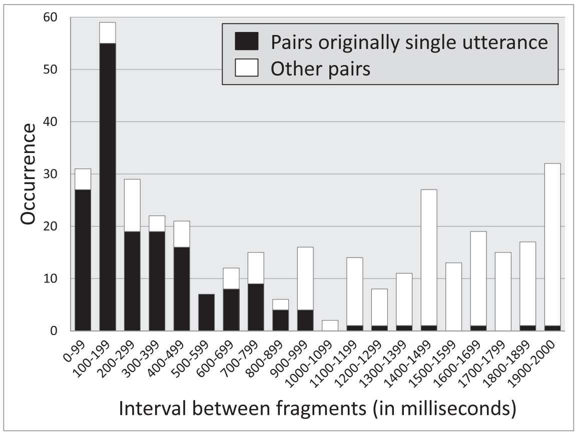

Fig. 6 Results of manual determination.

4.2.2 Effect of ASR Restoration

We checked whether the ASR restoration can actually improve utterance understanding accuracy. We further extracted 153 of the 376 pairs of VAD results in the restaurant domain that satisfied two conditions: the interval of the pairs was shorter than 900 milliseconds and their manual transcriptions contained at least one keyword. The former condition assumes a case when we determine whether the restoration is performed or not by a threshold for the interval. The latter is set because we here use utterance understanding accuracy, defined by using keywords, and utterances without keywords do not affect the accuracy. More specifically, we regarded an utterance pair as correctly understood only when all keywords in its manual transcription were correctly contained in its ASR result. The keyword set contained 2,789 POI names such as places, stores, and stations that are important in this domain, that is, for searching restaurant information.

The 153 pairs contained 124 originally single utterances that had been incorrectly segmented, 17 self-repairs, and 12 other pairs. The last category consisted of pairs of two independent utterances close in time, either of those was a filler or a noise, etc.

We set the following three conditions. Under Conditions 1 and 2, segmented utterances are not concatenated and a result from either fragment is selected, instead. Under Conditions 3 and 4, the fragments are restored in different ways.

Condition 1 Use the ASR result for the first fragment
Condition 2 Use the ASR result for the second fragment

[†] The option `-rejectshort` of Julius was used for this purpose.

[††] In a normal system without restoration, it will respond to self-repairs separately. These should not be simply connected, but another restoration method for them is required.

Table 3 Utterance understanding accuracy for three conditions.

Conditions	Accuracy
1. Use only first fragment	43/153 (28%)
2. Use only second fragment	31/153 (20%)
3. Connect two ASR results	103/153 (67%)
4. Concatenate two wav files	114/153 (75%)

Table 4 Changes in the number of correct results when each feature was removed.

Removed feature	Cross	All
(1) Average confidence score of first fragment	**− 5**	**−6**
(2) Confidence score of last word of first fragment	0	0
(3) LM score of first fragment	**1**	**−1**
(4) AM score of first fragment	−3	6
(5) Noise detection results by GMM	**−12**	**−3**
(6) Overlap ratio of phoneme bigrams	0	1
(7) Number of fillers in first fragment	0	5
(8) Number of fillers in second fragment	0	1
(9) Interval between fragments	**−130**	**−175**
(10) Duration of tail silence in first fragment	0	1
(11) Duration of head silence in second fragment	4	10
(12) Duration of first fragment	**−8**	**−21**
(13) Duration of final syllable of first fragment	13	17
(14) Volume change in final part of first fragment	0	1
(15) Frequency gradient in first vowel	0	5
(16) F0 range of first fragment	**−4**	**−4**
(17) Maximum loudness in first fragment	**−10**	**−10**
(18) Maximum loudness in second fragment	4	9

Bold: Features improving accuracy

Table 5 Number of occurrences of each feature in decision trees.

Features \ Domains	Restaurant	W.H.
(1) Ave. confidence score of first fragment	**4**	**1**
(3) LM score of first fragment	4	0
(5) Noise detection results by GMM	**9**	**10**
(9) Interval between fragments	**10**	**10**
(12) Duration of first fragment	5	0
(16) F0 range of first fragment	**4**	**1**
(17) Maximum loudness in first fragment	**8**	**9**

Bold: Effective features in both domains
W.H. denotes the world heritage domain.

Condition 3 Simply connect the two ASR results

Condition 4 Concatenate the two wav files and then perform ASR again for the concatenated file

Condition 1 corresponds to a system that ignores a user's barge-in. That is, the second fragment is ignored because it occurs during the system response for the first fragment. Condition 2 corresponds to a system that simply accepts a user's barge-in. That is, the system response for the first fragment is immediately terminated when the second fragment is detected. Thus, only the system response for the second fragment is shown to the user. An example of this case is shown in Fig. 2. Conditions 3 and 4 are to confirm whether the restoration is required or not.

The results are shown in Table 3. The accuracies of Conditions 3 and 4 were higher than those of Conditions 1 and 2 where no restoration is performed. This result may be self-evident in a sense that the selected target data contained many pairs of VAD results that were originally single utterances, 124 pairs out of 153, which can be correctly interpreted only when the pairs are considered to form one dialogue act. Nevertheless, it explicitly shows that the restoration can improve the utterance understanding accuracy if the system correctly identifies whether the restoration is required or not.

4.3 Classification Accuracy of Whether Restoration is Required or Not

We evaluate the crucial part of our method, i.e., whether the restoration is required or not, which is formulated as a binary classification problem. Hereafter, we used the data set after excluding self-repair utterances (255 and 354 fragment pairs) in the two domains.

When evaluating the classification accuracy, we performed cross-domain tests in addition to in-domain tests. All the in-domain tests were performed by ten-fold cross validation within one domain data. The cross-domain test indicates that the decision tree is trained on one domain data and its accuracy is evaluated on the other domain data. This is to verify whether or not the obtained decision trees are dependent on any specific domain. We performed four tests – two cross-domain tests and two in-domain tests – since we had two domains (restaurant and world heritage). Hereafter, "Cross" denotes results from the cross-domain test and "All" denotes total results from both the cross-domain and in-domain tests.

4.3.1 Results of Feature Selection

First, we identify features that had a negative influence on decision trees for all the 18 features. Table 4 shows the change in the number of correct classification results when each feature was removed from all 18 features. The negative values in the table mean that the accuracy of the decision tree degraded when the corresponding feature was removed. From these results, we selected seven features ((1), (3), (5), (9), (12), (16), and (17)) that had negative values for the "All" condition in the table.

Next, the results of selecting domain-independent features are shown in Table 5. The numbers in the table indicate how many times each feature was used in each of the 10 decision trees. They thus correspond to the importance of each feature in the domains. Five features, marked in bold in the table ((1), (5), (9), (16), and (17)), appeared in both domains and were regarded as domain-independent. We used these five as the selection result.

4.3.2 Classification Accuracy of Decision Trees

We compared the classification accuracies for the following three conditions: a baseline, "without feature selection", and "with feature selection". The baseline only used (9) the interval between fragments, which corresponds to a simple rule using optimal thresholds for the interval. The "without feature selection" condition used all 18 features listed in Table 1. The "with feature selection" condition used the five

Table 6 Classification accuracies of decision trees.

	Restaurant	W.H.	Restaurant→W.H.	W.H. → Restaurant
Baseline	215/255 (84.3%)	288/354 (81.4%)	285/354 (80.5%)	209/255 (82.0%)
Without feature selection	219/255 (85.9%)	291/354 (82.2%)	289/354 (81.6%)	214/255 (83.9%)
With feature selection	230/255 (90.2%)*	305/354 (86.2%)*	302/354 (85.3%)*	219/255 (85.9%)

W.H. denotes the world heritage domain.
* denotes difference from "Without feature selection" was statistically significant at 5% level.

Table 7 Changes in the number of correct results when each feature was removed from final feature set.

Removed features	Cross	All
(1) Average confidence score of first fragment	−9	−12
(5) Noise detection results by GMM	−30	−49
(9) Interval between fragments	−52	−124
(16) F0 range of first fragment	−9	−11
(17) Maximum loudness in first fragment	−3	−14

features obtained by the feature selection process, i.e., (1), (5), (9), (16), and (17).

Table 6 summarizes the classification accuracies of decision trees. "Restaurant" and "W.H." are the results of 10-fold cross validation in each domain. "Restaurant → W.H." and "W.H. → Restaurant" are the results of the cross-domain tests. For example, the former shows the result when the decision tree was trained on the restaurant domain data and its accuracy was calculated on the world heritage domain data. Our main objective is to improve the classification accuracy in the cross-domain tests, which are shown in the right half of Table 6, because the obtained decision tree should be domain-independent.

Under all conditions, the accuracies of "without feature selection" were slightly higher than those of the baseline. This indicates that the incorporated features were able to be helpful for the classification. Furthermore, the accuracies of "with feature selection" were also higher than those of "without feature selection". The differences were statistically significant at 5% level by McNemar test in the both in-domain conditions: $p = 0.012$ for "Restaurant" and $p = 0.023$ for "W.H." That was also statistically significant ($p = 7.9 \times 10^{-4}$) in condition "Restaurant → W.H." of the cross-domain test. These results suggest that the two kinds of feature selections successfully select effective features. But that was not statistically significant in the other cross-domain condition ($p = 0.38$). Insufficient test data can be a reason of this result and further investigation is required to verify the significance in the condition.

4.3.3 Analysis for Obtained Features

We performed an additional feature selection for the final five features to confirm their effectiveness. Table 7 summarizes the result. The numbers in the table indicate the change in the number of correct classification results when each feature was removed. Here, no features had positive values, indicating that no features had a negative influence. The classification accuracies significantly decreased under both the "Cross" and "All" conditions when the (5) and (9) features were removed. This indicates that (5) the noise detection results by GMM and (9) the interval between fragments were dominant. The next important feature was (17) the maximum loudness in the first fragment. This was effective to eliminate fillers and the system's synthesized voices, which tend to have lower loudness, from being incorrectly classified to be restored.

5. Related Work

There have been several studies on VAD and end-pointing with richer features. Their performance needs to be improved for better turn-taking. Sato et al. proposed a method for determining whether or not the system should respond by using decision tree learning [10]. They used various features such as the final word of the ASR results, the system's state, and prosodic information. Ohsuga et al. identified prosodic features that are helpful for determining ends of turns with decision tree learning on the Japanese Map Task Corpus [11]. Kitaoka et al. also used both prosodic and linguistic information to determine timing of system response generation [12]. Edlund et al. developed a prosodic analysis tool to augment end-point detection [13]. Raux and Eskenazi proposed dynamic adaptation of the threshold for the silence duration in a VAD module to improve the performance of end-pointing [14] and incorporated partial ASR results into their model [15]. Studies of incremental understanding [16]–[19] also inherently use partial ASR results to determine utterance ends. There have been various studies to improve VAD performance itself such as by using GMMs [20].

Even if the performance of end-pointing is further improved, a mechanism for restoring incorrect segmentation is required because such errors are unavoidable. We focus on a posteriori restoration of incorrect segmentation and develop a method with a normal VAD, specifically, that of the Julius ASR engine [21], which is based on the amplitude of the speech signal and the zero crossing rate [5]. The proposed method relies on neither a special ASR engine nor a specific end-pointing method; that is, it is complementary to other approaches. Integration with more sophisticated VAD and end-pointing methods remains for future work.

There have also been studies that tried to understand segmented utterances. Nakano et al. proposed a method for incrementally understanding segmented utterances and responding in real time [22]. Bell et al. proposed a method for handling fragmented utterances and controlling the system's behaviors [23]. It determines whether the system waits for remaining user input or starts to respond. These two studies assumed that a short pause occurs at word or clause boundaries and that ASR results are correctly obtained. The main difference between these studies and ours is that we assume a short pause may also occur within keywords such as POI names. Thus, ASR results are unreliable because fragments of keywords or keyphrases are not necessarily contained in the system's ASR dictionary and thus need to be restored after incorrect segmentation.

6. Conclusion

We determine whether or not a posteriori restoration is required in order to restore mistakenly segmented utterances caused by VAD errors. We formulated this as a binary classification problem that determines whether a fragment pair was originally a single utterance or not. We used decision tree learning with various features for which two kinds of feature selection were performed. Results demonstrated that the obtained decision trees outperformed the baseline in terms of classification accuracy.

In this paper we dealt with only cases where a user utterance is incorrectly segmented into two segments. There can be cases, however, where a user utterance is segmented into three or more segments. Our current implementation cannot deal with those cases since they are very rare in the corpora, but we think our proposed method is potentially able to restore them by sequentially determine the necessity of restoration.

Several issues remain as future work. First, it should be verified whether and how much the improvement of the classification accuracy affects the ASR accuracy of user utterances and more global metrics such as the task success rate. Second, since more sophisticated VAD and end-pointing methods would also be helpful, as explained in Sect. 5, integration with such methods should be also investigated. Finally, we have only considered "self-repaired utterances" in a preliminary experiment. There are several studies that treat such self-repairs including repetition, correction, and so on [24]–[27]. These findings can be used to determine how the segmented utterances should be handled.

Acknowledgments

This work was partly supported by JST PRESTO and the Naito Science & Engineering Foundation.

References

[1] K. Komatani, N. Hotta, and S. Sato, "Restoring incorrectly segmented keywords and turn-taking caused by short pauses," Proc. International Workshop on Spoken Dialogue Systems (IWSDS), pp.27–38, 2014.

[2] N. Hotta, K. Komatani, S. Sato, and M. Nakano, "Detecting incorrectly-segmented utterances for posteriori restoration of turn-taking and asr results," Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), pp.313–317, 2014.

[3] B. Shneiderman, Designing the User Interface, 3rd Edition, Addison-Wesley, 1997.

[4] A. Lee, K. Oura, and K. Tokuda, "MMDAgent — A fully open-source toolkit for voice interaction systems," Proc. IEEE International Conference on Acoustics, Speech & Signal Processing (ICASSP), pp.8382–8385, 2013.

[5] A. Benyassine, E. Shlomot, H.-Y. Su, D. Massaloux, C. Lamblin, and J.-P. Petit, "ITU-T Recommendation G.729 Annex B: A silence compression scheme for use with G.729 optimized for v.70 digital simultaneous voice and data applications," IEEE Commun. Mag., vol.35, no.9, pp.64–73, 1997.

[6] E.E. Jan, B. Maison, L. Mangu, and G. Zweig, "Automatic construction of unique signatures and confusable sets for natural language directory assistance application," Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH), pp.1249–1252, 2003.

[7] M. Katsumaru, K. Komatani, T. Ogata, and H.G. Okuno, "Adjusting occurrence probabilities of automatically-generated abbreviated words in spoken dialogue systems," Next-Generation Applied Intelligence, Lecture Notes in Computer Science, vol.5579, pp.481–490, Springer, Berlin, Heidelberg, 2009.

[8] A. Lee, K. Nakamura, R. Nisimura, H. Saruwatari, and K. Shikano, "Noise robust real world spoken dialogue system using GMM based rejection of unintended inputs," Proc. Int'l Conf. Spoken Language Processing (ICSLP), pp.173–176, 2004.

[9] M. Nakano, S. Sato, K. Komatani, K. Matsuyama, K. Funakoshi, and H.G. Okuno, "A two-stage domain selection framework for extensible multi-domain spoken dialogue systems," Proc. Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL), pp.18–29, June 2011.

[10] R. Sato, R. Higashinaka, M. Tamoto, M. Nakano, and K. Aikawa, "Learning decision trees to determine turn-taking by spoken dialogue systems," Proc. Int'l Conf. Spoken Language Processing (ICSLP), pp.861–864, 2002.

[11] T. Ohsuga, M. Nishida, Y. Horiuchi, and A. Ichikawa, "Investigation of the relationship between turn-taking and prosodic features in spontaneous dialogue," Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH), 2005.

[12] N. Kitaoka, M. Takeuchi, R. Nishimura, and S. Nakagawa, "Response timing detection using prosodic and linguistic information for human-friendly spoken dialog systems," Transactions of the Japanese Society for Artificial Intelligece, vol.20, no.3, pp.220–228, 2005.

[13] J. Edlund, M. Heldner, and J. Gustafson, "Utterance segmentation and turn-taking in spoken dialogue systems," Computer Studies in Language and Speech, pp.576–587, 2005.

[14] A. Raux and M. Eskenazi, "Optimizing endpointing thresholds using dialogue features in a spoken dialogue system," Proc. SIGdial Workshop on Discourse and Dialogue, pp.1–10, 2008.

[15] A. Raux and M. Eskenazi, "A finite-state turn-taking model for spoken dialog systems," Proc. Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT NAACL), pp.629–637, 2009.

[16] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," Proc. Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL), pp.1–8, 2010.

[17] T. Baumann and D. Schlangen, "Predicting the micro-timing of user input for an incremental spoken dialogue system that completes a user's ongoing turn," Proc. Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL), pp.120–129, 2011.

[18] E. Selfridge, I. Arizmendi, P.A. Heeman, and J.D. Williams, "Stability and accuracy in incremental speech recognition," Proc. Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL), pp.110–119, 2011.

[19] D. Traum, D. DeVault, J. Lee, Z. Wang, and S. Marsella, "Incremental dialogue understanding and feedback for multiparty, multimodal conversation," Intelligent Virtual Agents, Lecture Notes in Computer Science, vol.7502, pp.275–288, Springer, Berlin, Heidelberg, 2012.

[20] H. Sakai, T. Cincarek, H. Kawanami, H. Saruwatari, K. Shikano, and A. Lee, "Voice activity detection applied to hands-free spoken dialogue robot based on decoding using acoustic and language model," Proc. 1st International Conference on Robot Communication and Coordination (ROBOCOMM), 2007.

[21] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine Julius," Proc. APSIPA ASC: Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference, pp.131–137, 2009.

[22] M. Nakano, N. Miyazaki, J.-I. Hirasawa, K. Dohsaka, and T. Kawabata, "Understanding unsegmented user utterances in real-time spoken dialogue systems," Proc. 37th Annual Meeting of the Association for Computational Linguistics (ACL), pp.200–207, 1999.

[23] L. Bell, J. Boye, and J. Gustafson, "Real-time handling of fragmented utterances," Proc. NAACL Workshop on Adaption in Dialogue Systems, pp.2–8, 2001.

[24] M.G. Core and L.K. Schubert, "A syntactic framework for speech repairs and other disruptions," Proc. 37th Annual Meeting of the Association for Computational Linguistics (ACL), pp.413–420, 1999.

[25] P.A. Heeman and J.F. Allen, "Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialogue," Computational Linguistics, vol.25, pp.527–571, 1999.

[26] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," IEEE Trans. Audio Speech Language Process., vol.14, no.5, pp.1526–1540, Sept. 2006.

[27] K. Georgila, N. Wang, and J. Gratch, "Cross-domain speech disfluency detection," Proc. Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL), pp.237–240, 2010.

Kazunori Komatani received B.E., M.S., and Ph.D degrees in Informatics in 1998, 2000, and 2002 from Kyoto University, Japan. He is currently a professor of the Institute of Scientific and Industrial Research, Osaka University. He has received several awards including the 2002 FIT Young Researcher Award and the 2004 IPSJ Yamashita SIG Research Award, both from the Information Processing Society of Japan (IPSJ). He is currently a SIGDIAL Scientific Advisory Committee member and a JSAI executive board member. He is also a member of IPSJ, NLP, and ISCA.

Naoki Hotta received the B.E. and M.E. degrees from Nagoya University in 2013 and 2015, respectively. His research theme was utterance timing in spoken dialogue systems while he was in the university.

Satoshi Sato is a professor of Graduate School of Engineering in Nagoya University. His research is centered on natural language processing, artificial intelligence, and automatic information compilation. He received B.E., M.E., and Doctor of Engineering from Kyoto University in 1983, 1985, and 1992, respectively.

Mikio Nakano is a principal researcher at Honda Research Institute Japan Co., Ltd. He received his M.S. degree in Coordinated Sciences and Sc.D. degree in Information Science from the University of Tokyo, respectively in 1990 and 1998. From 1990 to 2004, he worked for Nippon Telegraph and Telephone Corporation. He was a visiting scientist at MIT Laboratory for Computer Science from 2000 to 2002. In 2004, he joined Honda Research Institute Japan Co., Ltd. His research interests include spoken dialogue systems, speech understanding, and conversational robots. He is a member of ACM, ACL, IEEE, ISCA, JSAI, IPSJ, RSJ, and ANLP.