Dialogue & Discourse 8(2) (2017) 206–224 doi: 10.5087/dad.2017.209

User-Adaptive A Posteriori Restoration for Incorrectly Segmented Utterances in Spoken Dialogue Systems[*]

Kazunori Komatani KOMATANI@SANKEN.OSAKA-U.AC.JP
The Institute of Scientific and Industrial Research, Osaka University

Naoki Hotta
Graduate School of Engineering, Nagoya University

Satoshi Sato SSATO@NUEE.NAGOYA-U.AC.JP
Graduate School of Engineering, Nagoya University

Mikio Nakano NAKANO@JP.HONDA-RI.COM
Honda Research Institute Japan Co., Ltd.

Editor: Amanda Stent

Submitted 05/2016; Accepted 07/2017; Published online 12/2017

Abstract

Ideally, the users of spoken dialogue systems should be able to speak at their own tempo. Thus, the systems needs to interpret utterances from various users correctly, even when the utterances contain pauses. In response to this issue, we propose an approach based on a posteriori restoration for incorrectly segmented utterances. A crucial part of this approach is to determine whether restoration is required. We use a classification-based approach, adapted to each user. We focus on each user's dialogue tempo, which can be obtained during the dialogue, and determine the correlation between each user's tempo and the appropriate thresholds for classification. A linear regression function used to convert the tempos into thresholds is also derived. Experimental results show that the proposed user adaptation approach applied to two restoration classification methods, thresholding and decision trees, improves classification accuracies by 3.0% and 7.4%, respectively, in cross validation.

Keywords: spoken dialogue system, turn taking, user adaptation, a posteriori restoration

1. Introduction

To make spoken dialogue systems more user-friendly, users need to be able to speak at their own tempo. Even though not all users speak fluently, i.e., some speak slowly and put some pauses within their utterances, conventional systems basically assume that users say every utterance with no pauses. Systems need to handle utterances both by novice users who speak slowly and by experienced users who want the system to reply quickly.

We propose a method for spoken dialogue systems to interpret user utterances adaptively in terms of utterance units. We utilize an approach based on our *a posteriori restoration* method for incorrectly segmented utterances (Komatani et al., 2014). The proposed method allows the system

[*] This paper is a modified and extended version of our earlier report (Komatani et al., 2015).

to respond quickly while also interpreting utterance fragments by concatenating them when a user speaks with pauses.

Another approach to this issue is to change the parameters of voice activity detection (VAD) adaptively for each user during dialogue. However, automatic speech recognition (ASR) engines with such adaptive control are uncommon, and implementing an online-adaptive VAD module is difficult. Our a posteriori restoration approach does not require changing ASR engines but uses VAD and ASR results as they are, and it restores the interpretation of user utterances.

A crucial part of our approach is to determine whether or not two utterance fragments that are close in time need to be interpreted together, i.e., whether these are two different utterances or a single utterance incorrectly segmented by VAD. If the fragments need to be interpreted separately, the system normally responds to each on the basis of their ASR results. If they need to be interpreted together, the system immediately stops its response to the first fragment, concatenates the two segments, and then interprets the combined segments.

Misclassification of whether restoration is needed causes erroneous system responses. If the system incorrectly classifies the restoration as not being required, its response may become erroneous because the original user utterance is interrupted in the middle. If the system classifies the restoration as being required even though the restoration is actually not, the system takes an unnecessarily long time before it starts responding, and its response tends to be erroneous because an unnecessary part is attached to the actual utterance.

In this work, we present a method for adapting the restoration classifier to each user, and we show through experiments that user adaptation improves classification accuracy. We focus on the tempo of each user and use it to adapt the classifier. Because the temporal interval between two utterance fragments is an important parameter in the classifier (Komatani et al., 2014), we adapt its threshold to user behaviors obtained during the dialogue.

2. Related Work

The aim of our restoration is to resolve a problem with utterance units. Spoken dialogue systems that do not consider the problem naively assume that these three items are always in agreement:

1. Results of voice activity detection (VAD)
2. Units of dialogue acts (DAs)
3. Units of user turns

The second item is used to update dialogue states in the system, and the third determines when the system starts responding.

However, these three do not agree in cases of real user utterances. Because the first item is the input received by the system, existing studies on the problem can be categorized into two: handling disagreements between 1 and 2 and between 1 and 3. The disagreement between 1 and 2 was tackled by Nakano et al. (1999) and Bell et al. (2001). The purpose of these studies was to understand fragmented utterances incrementally and to determine whether or not each fragment forms a DA with another. These approaches treated each word as a unit of the fragments. Methods for incremental understanding (Skantze and Hjalmarsson, 2010; Baumann and Schlangen, 2011; Selfridge et al., 2011; Traum et al., 2012) treat each input frame as a unit of the fragments and determined whether or not each input frame forms a DA with previous ones. The disagreement

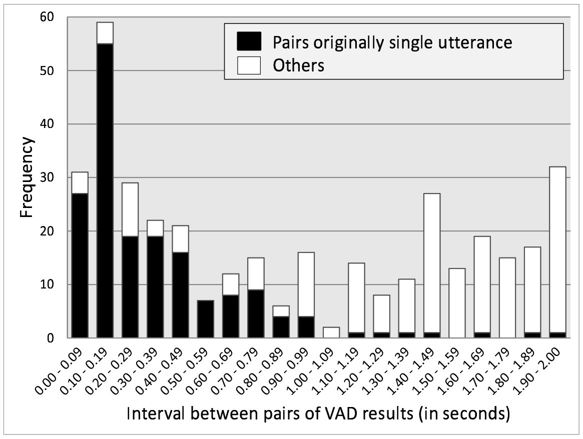

Figure 1: Distributions of intervals between utterance fragments (Komatani et al., 2014)

between 1 and 3 was tackled by Sato et al. (2002), Ferrer et al. (2003), and Kitaoka et al. (2005), who determined the timing at which a system needs to start responding. Raux and Eskenazi also tackled this problem by changing the thresholds for silence duration in a VAD module (Raux and Eskenazi, 2008) and by incorporating partial ASR results into their model (Raux and Eskenazi, 2009). Some current systems allow the system to start responding even during user utterances (Paetzel et al., 2015; Zhao et al., 2015).

We define a correct utterance unit as a dialogue act (DA). Our a posteriori restoration framework primarily considers the disagreement between 1 and 2, by restoring fragmented ASR results. Unlike previous studies, such as (Nakano et al., 1999) and (Bell et al., 2001), which are based on syntactic parsing, our method assumes that the DA boundaries are a subset of the VAD boundaries. The disagreement between 1 and 3 is partially considered in our framework, which can decide whether to respond to a fragmented utterance. Our problem setting relates in part to the one tackled by (Kitaoka et al., 2005; Raux and Eskenazi, 2008, 2009), in which the system determines more precise timing to respond. Thus, our approach can be used together with these methods to improve turn-taking.

User-adaptive spoken dialogue systems can be categorized into two types: adaptation of the system's output and adaptation during input interpretation. Several previous studies have adapted the system output to users by changing behaviors such as the content of system utterances (Jokinen and Kanto, 2004) and dialogue management (Komatani et al., 2005), pause and gaze duration (Dohsaka et al., 2010), and backchannel behaviors (e.g., head nods or short vocalization like "uh-huh") (de Kok et al., 2013). However, only a few studies have been conducted on adaptation during input interpretation. As one example, Paek and Chickering (2007) exploited the history of a user's commands and adapted the system's ASR language model to the user.

Our adaptation is concerned with input interpretation, i.e., in which unit the system interprets user utterances, thereby aiming at improving utterance understanding accuracy. As far as we know, this is the first method of user adaptation proposed for the restoration of utterance units.

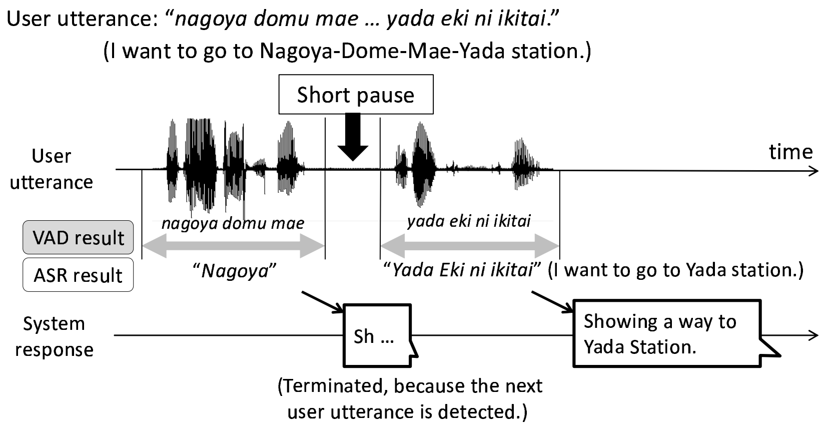
User utterance: "nagoya domu mae … yada eki ni ikitai."
(I want to go to Nagoya-Dome-Mae-Yada station.)
Short pause
User utterance
time
VAD result
ASR result
nagoya domu mae
yada eki ni ikitai
"Nagoya"
"Yada Eki ni ikitai" (I want to go to Yada station.)
System response
Sh …
Showing a way to Yada Station.
(Terminated, because the next user utterance is detected.)

Figure 2: Example of incorrect utterance segmentation due to short pauses

3. Posteriori Restoration for Incorrectly Segmented Utterances

Most conventional dialog systems wait to respond until they have received an entire user utterance, to which a VAD result is assumed to correspond. That is, such a system responds when every VAD result (possibly an utterance segment) arrives on the basis of the ASR and understanding result for it. However, user utterances are not always obtained as single VAD results, and they are often incorrectly segmented into several fragments. Figure 1 shows the distributions of intervals between pairs of VAD results classified by whether or not they were originally single utterances (i.e., incorrectly segmented) (Komatani et al., 2014). The total number of pairs in this data set was 376. The pairs were manually annotated. We can see that many pairs were incorrectly segmented by VAD and that their intervals tended to be shorter than others. These pairs need to be restored to obtain correct interpretations of the original utterances.

We explain how conventional dialog systems respond to a pair of utterance fragments, denoted as first and second fragments. When a user utterance is incorrectly segmented by VAD, a problem called a false cut-in occurs, and the system incorrectly starts speaking even during a user utterance.

Because the pair is close in time, we can think of two types of basic system behavior concerning whether or not barge-in is allowed.

- System prohibits barge-in: the system responds to the first fragment and ignores the second fragment of the user utterance while it is speaking.

- System allows barge-in: the system terminates its response to the first fragment and responds on the basis of an ASR result for the second fragment only.

The latter strategy is better for the false cut-in problem because its damage is smaller. This is because the system allowing barge-in immediately stops speaking when the user starts speaking, while the system prohibiting barge-in continues to speak, thereby requiring the user to wait until the whole system utterance ends.

An example of incorrect segmentation is illustrated in Fig. 2. The system in this example adopts the latter strategy: allowing the user to barge in. The user intends to say "I want to go to Nagoya-Dome-Mae-Yada station," but a short pause occurs between "Nagoya-Dome-Mae" and "Yada." If

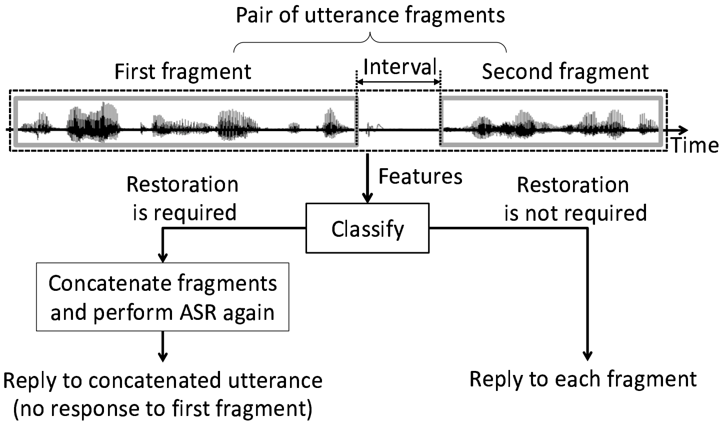
Pair of utterance fragments
First fragment
Interval
Second fragment
Time
Features
Restoration is required
Classify
Restoration is not required
Concatenate fragments and perform ASR again
Reply to concatenated utterance (no response to first fragment)
Reply to each fragment

Figure 3: Overview of a posteriori restoration (Komatani et al., 2014)

the system correctly recognizes the user utterance and understands that "Nagoya-Dome-Mae-Yada" is a station name, it should respond with "I will show you the way to Nagoya-Dome-Mae-Yada station." However, the ASR result is incorrect because ASR is performed separately for two fragments: "Nagoya-Dome-Mae" and "Yada eki ni ikitai." Consequently, the system responds with "Showing a way to Yada station" (Yada is another station) on the basis of the incorrect ASR result for the second fragment, because the response to the first fragment is immediately terminated when the second fragment starts. The resulting system response does not match the user's request.

An outline of our a posteriori restoration process (Komatani et al., 2014) is shown in Fig. 3. This approach is based on the latter type of conventional dialog system, which allows barge-in and thus aims at alleviating the false cut-in problem. When a pair of utterance fragments is close in time, this process is invoked when the second fragment starts. The process consists of two steps:

1. Classify whether or not a pair of utterance fragments resulted from an incorrect segmentation, i.e., whether or not restoration is required.

2. Restore the utterance if it has been incorrectly segmented; i.e., concatenate the two utterance fragments and perform ASR again for the resulting segment. Because the system allows barge-in, it also tries to reduce the damage caused by the false cut-ins by terminating its response to the first fragment and waiting until the second fragment ends to avoid the system speaking during a user utterance.

If restoration is required, the system performs ASR again after concatenating the fragments. The system then responds on the basis of the ASR result for the concatenated fragments. In particular, ASR for incorrectly segmented fragments tends to be erroneous when a long word is incorrectly segmented in it. Restoration improves utterance understanding accuracy. Over 153 segment pairs that were close in time and that contained at least one keyword, accuracy was 75% (114/153) when restoration was performed, while accuracy was 28% (43/153) and 20% (31/153) when the ASR result of either a first or second segment was only used for understanding. (Komatani et al., 2014).

If restoration is not required, i.e., 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.

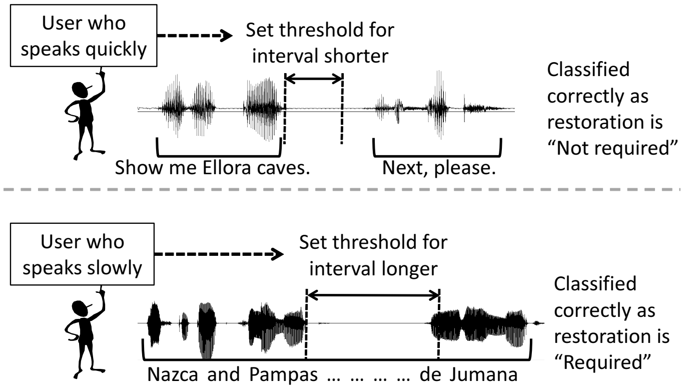
User who speaks quickly
Set threshold for interval shorter
Classified correctly as restoration is "Not required"
Show me Ellora caves.
Next, please.
User who speaks slowly
Set threshold for interval longer
Classified correctly as restoration is "Required"
Nazca and Pampas … … … … de Jumana

Figure 4: Examples of user-adapted restoration

When the system starts the second response also depends on the system's configuration: after the first response finishes or immediately by stopping the first response.

A trade-off exists between the occurrences of erroneous system responses caused by incorrect segmentation and response delay resulting from restoration. Our approach gives weight to preventing erroneous responses at the cost of a small delay in system responses. We have endeavored to reduce damage stemming from the delay, by producing fillers such as "Well" to prevent unnatural silences (Komatani et al., 2014) and by improving implementation to reduce the delay itself.

Even if the performance of end-pointing is further improved, a mechanism for restoring incorrect segmentation is required because such errors are unavoidable. Our a posteriori restoration of incorrect segmentation works with a normal VAD, specifically, that of the Julius ASR engine (Lee and Kawahara, 2009), which is based on the amplitude of the speech signal and the zero crossing rate (Benyassine et al., 1997). 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. For example, Sakai et al. (2007) showed that VAD performance improved by using GMMs. Prosodic features are also known to be helpful in determining ends of turns (Ohsuga et al., 2005; Edlund et al., 2005).

4. Obtaining Appropriate Thresholds from Dialogue Tempos

We analyze each user's dialogue tempo, which is a parameter representing the way a user speaks quantitatively. In improving the classification accuracy of whether or not an utterance fragment pair is required to be restored, the threshold for the temporal interval between fragments plays a dominant role. We assume that appropriate thresholds depend on the way each user speaks. Examples of how the thresholds need to change are given in Fig. 4.

Brisk users are assumed to speak fluently with shorter pauses. Thus, the threshold needs to be set shorter, thereby avoiding unnecessary restoration and subsequent late responses. We should point out here that such users often repeat their utterances when the system's response is not quick enough because they think the system has not heard their utterance, and this causes utterance collision (Funakoshi et al., 2010).

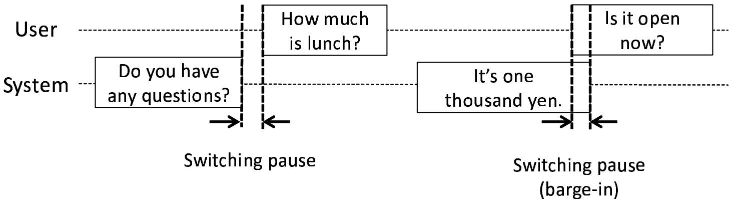

Figure 5: Examples of switching pauses

In contrast, "slow" users often speak with long pauses during their utterances. In this case, the threshold needs to be set longer, thereby enabling the system to restore utterances even when longer pauses exist in a single utterance.

4.1 Definition of Dialogue Tempo

We define dialogue tempo as a quantitative parameter showing how each user speaks. Specifically, it is defined as the average duration of switching pauses, which are the times between when a system finishes speaking and when a user starts speaking, as depicted in Fig. 5. We calculate this for each user from the beginning of the dialogue. The duration of a switching pause can be negative, as when the user barges in, i.e., when the user starts speaking during a system utterance.

Although the speaking rate can also be used for defining the tempo, we here use the duration of switching pauses. Although the tempo is calculated for each dialogue, it can be accumulated per user across dialogues when a user ID can be obtained (e.g., mobile phones and in-car interfaces).

4.2 Appropriate Threshold for Interval

We set appropriate thresholds for each user to investigate the relationship of the threshold to the dialogue tempo. By "appropriate," we mean that the threshold can classify whether or not restoration is required with high accuracy. Restoration for an utterance fragment pair is classified as "required" if its interval is shorter than the threshold and "not required" otherwise.

Here, we set the threshold as a discriminant plane (point) of a support vector machine (SVM) whose only feature is the temporal interval between two utterance fragments. A reference label was manually given, i.e., whether or not the restoration is required. We used the SMO module in Weka (version 3.6.9) (Hall et al., 2009) as an SVM implementation. The parameters were set to the default values, e.g., its kernel function was polynomial. The SVM can set a discriminant plane that maximizes distances between classes. If a user's training data did not contain both positive and negative labels, we set fixed values for the threshold as exceptions: large enough (2.00 seconds) when all the labels in the training data were "restoration is not required" and small enough (0.00 seconds) when they were all "restoration is required."

4.3 Target Data

Our target data were collected by our spoken dialog system that introduces world heritage sites (Nakano et al., 2011). In total, speech data from 35 participants was recorded. Each participant

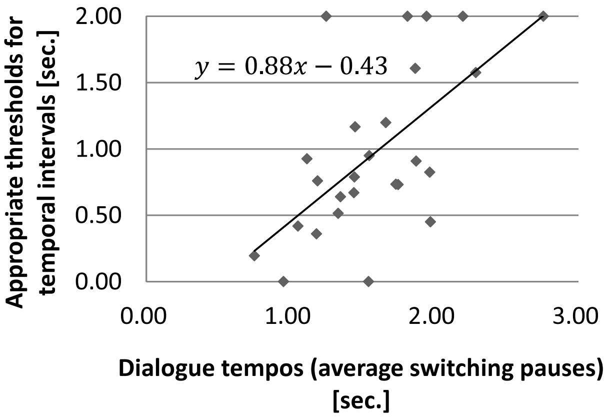

Figure 6: Correlation between appropriate thresholds and dialogue tempos per participant

engaged in four 8-minute dialogues. Participants were not given any special instructions prior to or during the dialogues.

We used the data of only 26 of the 35 participants because nine participants did not have sufficient utterance pairs. Specifically, we used the data only of participants who had more than six utterance pairs whose temporal intervals were close in time (less than 2.00 seconds), and whose fragments were longer than 0.80 seconds. This was because our target was originally a single utterance and because we regarded pairs that are very short and that have intervals greater than 2.00 seconds as not being such an utterance (Komatani et al., 2014).

We obtained 3099 utterances from the 26 participants. The data included 390 utterance pairs that satisfy the aforementioned conditions to possibly be a single utterance. We manually assigned the labels of whether or not the pair is a single utterance in accordance with the procedure in (Hotta et al., 2014). Because 240 pairs were originally single utterances and 150 pairs were not, the classification accuracy of a majority baseline is 61.5%.

4.4 Correlation between Dialogue Tempos and Appropriate Thresholds

We investigated the correlation between dialogue tempos and the appropriate thresholds for restoration for each of the 26 participants. All 3099 utterances were used to obtain the dialogue tempos of each participant. We excluded outliers: specifically, utterances whose switching pauses are less than -3.5 seconds and more than 6 seconds were excluded because such large values simply indicate that the participant was thinking deeply. These values were determined experimentally.

Figure 6 plots the data of the 26 participants, where the x-axis denotes the dialogue tempos and the y-axis denotes the appropriate thresholds, both in seconds. First, we can see that the appropriate thresholds (y-axis values) varied depending on the participant. This shows the distributions of within-speaker pauses are different across users, thereby indicating that adaptation of the thresholds can improve the classification accuracy. Second, the correlation coefficient between the two values was 0.63. The linear regression function is derived as

$$y = 0.88x - 0.43. \tag{1}$$

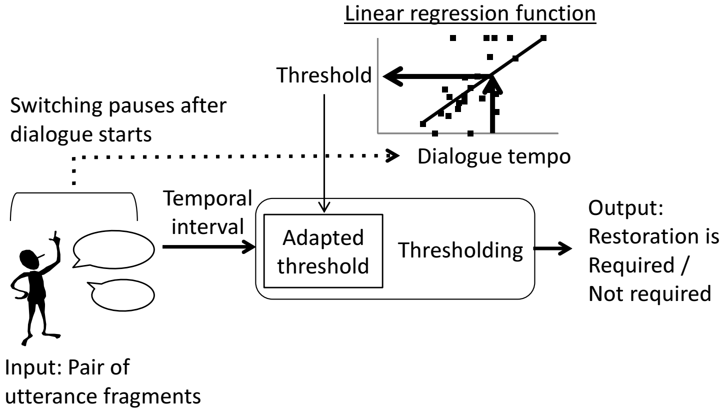
Linear regression function
Threshold
Switching pauses after dialogue starts
Dialogue tempo
Temporal interval
Adapted threshold
Thresholding
Output: Restoration is Required / Not required
Input: Pair of utterance fragments

Figure 7: User-adapted classification using thresholding

This function is used in the next section for estimating thresholds from the dialogue tempos per user.

Here, dialogue tempos are used as an approximation to represent how fluently a user speaks. The dialogue tempos and the thresholds correspond to between-speaker and within-speaker pauses, respectively. These two kinds of pauses are different, but we have empirically shown that these two pauses were correlated when analyzing them per user. Because the dialogue tempos can be obtained during dialogues, this empirical correlation is used to estimate the thresholds for the classification.

We can also see that the appropriate thresholds were generally smaller than the dialogue tempos, which can be seen in Eq. (1): the coefficient of the first-order term was less than one (0.88), and the constant term was also negative (-0.43). Because the thresholds reflect the upper bounds of durations of within-speaker pauses, the actual distributions of within-speaker pauses were shorter than the thresholds. This tendency is different from the results shown by Heldner and Edlund (2010), whose finding was that within-speaker pauses are generally longer than switching pauses in human-human conversation corpora. One of the reasons may be that our data were collected from human-computer dialogues, where users tend to speak more formally than in human-human conversation. More detailed analysis on such differences when users speak with humans and systems remains for future work.

5. Adapting Classifiers for Restoration to Users

We present our investigation on whether or not the correlation between dialogue tempos and appropriate thresholds is helpful. The correlation is used to derive the user-adaptive threshold from the user's dialogue tempo, thereby improving classification accuracy for whether or not restoration is required. First, the system derives threshold values for the temporal intervals from the user's dialogue tempos by using the linear regression function. It then adapts the classifier to each user. We examined user adaptation for two classification methods: thresholding and decision trees.

5.1 Thresholding

Thresholding is the simplest method for classification on the basis of the temporal interval between utterance fragments. We first discuss the effectiveness of user adaptation with this method.

The process flow of thresholding with user adaptation is shown in Fig. 7. Its input is a pair of utterance fragments (and the temporal interval between them). The system calculates the user's dialogue tempo on the basis of switching pauses from when the dialogue starts and derives a threshold value corresponding to the tempo using the linear regression function. The system then classifies whether or not the restoration is required using the adapted threshold. The restoration for a pair is classified as "required" if its temporal interval is shorter than the adapted threshold and is "not required" otherwise.

5.2 Decision Trees

We also use decision trees, which are a more complicated classifier than thresholding. We show that user adaptation is also effective in this case.

The process flow of the decision trees with user adaptation is depicted in Fig. 8. In addition to the temporal interval between a pair of utterance fragments, we use four features that were shown to be effective in our previous report (Hotta et al., 2014): an average confidence score of the first fragment, noise detection results using a Gaussian mixture model (GMM), the F0 range of the first fragment, and the maximum loudness in the first fragment.

User adaptation is performed by converting the temporal interval only; the other four features are not changed here. The interval is converted in both the training and classification phases in the decision tree learning. Instead of adapting the threshold to each user, we convert its feature values. This is because, in the normal training phase of decision tree learning, a single decision tree having fixed thresholds across different users is obtained. Our approach is to convert the feature values relatively for the interval in accordance with each user, thereby enabling the system to classify adaptively to users with a constant threshold. Specifically, we use ratios between the threshold values of the target user and the average one of all users. The feature value is converted using Eq. (2), where we denote an original interval i by a user j as I_{ij} and its converted value as $\hat{I_{ij}}$:

$$\hat{I_{ij}} = I_{ij} \times \frac{T_0}{T_j}, \tag{2}$$

where T_j is a threshold value adapted to user j, which is obtained from the user's dialogue tempo and the linear regression function, and T_0 is a constant set to 0.519 seconds[1], which was the average interval of all users.

Our aim with this conversion is as follows. The correlation depicted in Fig. 6 shows that thresholds need to be smaller for users with quicker dialogue tempos. This conversion makes the feature values of the interval relatively larger for such users (having smaller T_j) by multiplying the ratio T_0/T_j. This is equivalent to setting a relatively smaller threshold even though fixed and common thresholds are actually used in decision tree learning.

[1] The constant T_0 is not necessarily required. It is used to adjust the feature values within a similar value range as the original ones for making it easy to interpret learned trees.

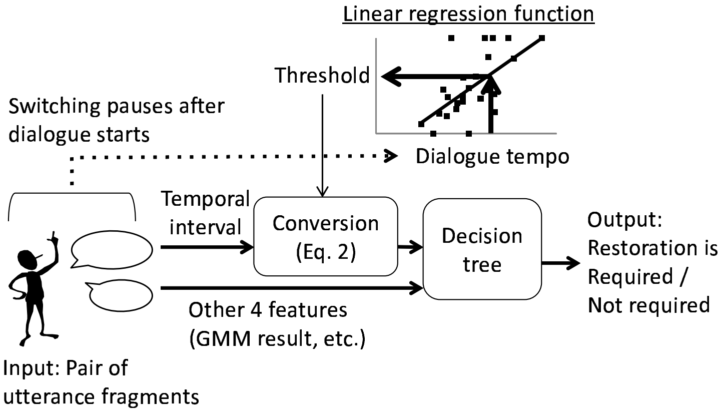
Linear regression function
Threshold
Switching pauses after dialogue starts
Dialogue tempo
Temporal interval
Conversion (Eq. 2)
Decision tree
Output: Restoration is Required / Not required
Other 4 features (GMM result, etc.)
Input: Pair of utterance fragments

Figure 8: User-adapted classification using a decision tree

6. Experimental Evaluation

We investigated whether user adaptation contributes to improving the classification accuracy. We also experimentally checked the upper limit and convergence speed of the proposed adaptation by comparing the accuracy with its batch version, in which all utterance data from a target user are assumed to be available.

6.1 Performance of User Adaptation

We report classification accuracy for the two methods, thresholding and decision trees, discussed in Section 5. Experiments were conducted under two conditions: closed test and cross validation.

Figure 9 illustrates the training and test data usage for the experimental evaluation. The background models to be trained were the linear regression function (i.e., coefficients a and b) and decision trees for the thresholding and decision tree methods, respectively. During the training of decision trees, the linear regression function (fixed here, as explained later) and each participant's dialogue tempo were used to convert feature values. The training data for the background models were those of all 26 participants in the closed test and those of 25 participants excluding one for the test and adaptation phase in cross validation. That is, cross validation was performed by leaving one participant out.

The data contained utterances of participants in chronological order. More specifically, for pairs of VAD results close in time, their features including temporal intervals and their reference labels (i.e., whether or not they should be restored) were given. Durations of switching pauses were also recorded for every utterance, by which a dialogue tempo was calculated at each point in time.

The adaptation and test phase was performed by loading each participant's data chronologically. Each user's dialogue tempo was calculated by using the duration of switching pauses from the beginning of the dialogue until the target utterance[2]. Adaptation was based on the target user's

[2] In Sections 6.2 and 6.3, batch adaptation is used; all switching pause durations of the target user are loaded at once in advance, meaning the user's dialogue tempo is already known from the beginning of the dialogue.

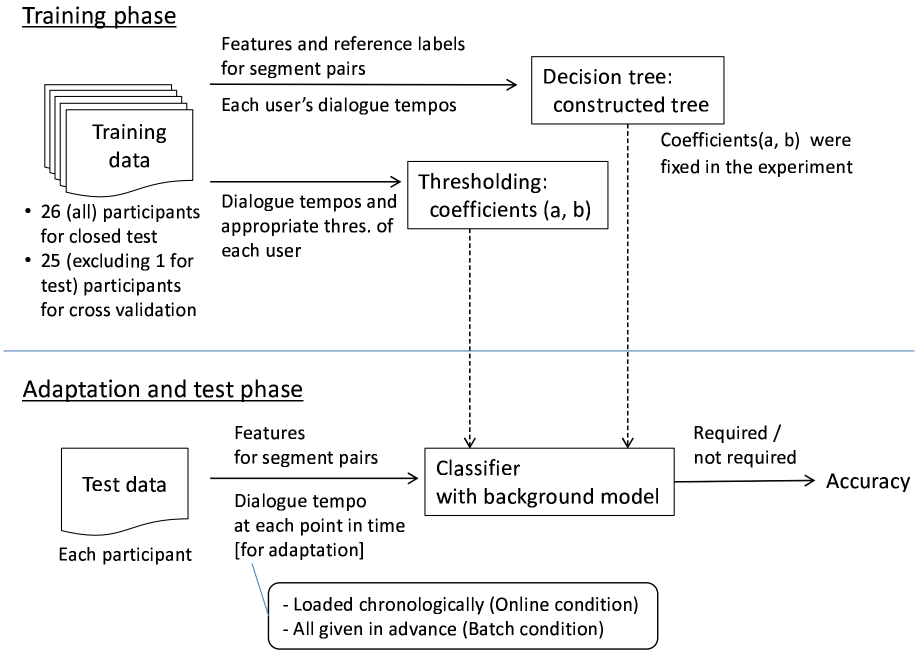
Training phase
Training data
• 26 (all) participants for closed test
• 25 (excluding 1 for test) participants for cross validation
Features and reference labels for segment pairs
Each user's dialogue tempos
Decision tree: constructed tree
Coefficients(a, b) were fixed in the experiment
Dialogue tempos and appropriate thres. of each user
Thresholding: coefficients (a, b)
Adaptation and test phase
Test data
Each participant
Features for segment pairs
Dialogue tempo at each point in time [for adaptation]
Classifier with background model
Required / not required
Accuracy
- Loaded chronologically (Online condition)
- All given in advance (Batch condition)

Figure 9: Training and test data usage for experimental evaluation

Table 1: Deviation of parameters in the linear regression function

	a	b
Average	0.883	−0.431
Std. dev.	0.034	0.057

dialogue tempo at each point in time using the linear regression function. More specifically, thresholds were changed by the function in the thresholding method, and feature values were converted by Eq. (2) in the decision tree method. Then classification accuracy was calculated by comparing the outputs of the classifier and reference labels for the pairs of VAD results.

The deviations of the two parameters of the linear regression function $y = ax + b$ during the cross validation for the thresholding method are listed in Table 1. The two parameter values, a and b, only changed slightly, and their averages were almost the same as the coefficients in Eq. (1), which were calculated using all the data. Thus, we used the same parameter values ($a = 0.88$ and $b = -0.43$) in the decision tree method for simplicity of experimentation.

6.1.1 THRESHOLDING ADAPTED TO USERS

Classification accuracies in thresholding are listed in the left column of Table 2. In the "no adaptation" condition for the closed test, a constant threshold (0.822 seconds) was used to classify all data. This threshold was determined optimally for all data by an SVM (SMO in Weka) in the same

Table 2: Classification accuracies with/without adaptation

	Thresholding		Decision tree	
	closed	cross validation	closed	cross validation
No adaptation	281/390 (72.1%)	281/390 (72.1%)	312/390 (80.0%)	271/390 (69.5%)
Online adaptation	294/390 (75.4%)	293/390 (75.1%)	320/390 (82.1%)	300/390 (76.9%)

manner as discussed in Section 4.2. In the "no adaptation" condition for cross validation, thresholds were determined for each fold; that is, data of 25 participants were used to determine the threshold for testing the excluded data of one participant. This was repeated 26 times.

User adaptation improved classification accuracies by 3.3 and 3.0 percentage points for the closed test and cross validation, respectively. We can also see that the accuracies were almost equivalent under both adaptation conditions ("no" and "online"). This suggests that no overfitting occurred in these cases, so a similar performance will also be obtained for unknown users. The number of parameters is small, which is why they are stable, as already shown in Table 1.

6.1.2 DECISION TREE LEARNING ADAPTED TO USERS

Classification accuracies for decision tree learning are listed in the right column of Table 2. The "no adaptation" condition denotes normal decision tree learning, that is, no feature values were converted using Eq. (2). User adaptation improved the accuracies by 2.1 and 7.4 percentage points for the closed test and the cross validation, respectively. The difference in cross validation was statistically significant according to the McNemar test ($p = 3.2 \times 10^{-4}$).

We can see that the accuracies in cross validation were lower than those in the closed test. This is because a decision tree has many more parameters to be trained than thresholding, so the obtained trees were overfitted to the training data. This means that the accuracies under the closed test condition were unreasonably high. Note that the accuracy under the "no adaptation" condition in cross validation was lower than that in the same condition for the thresholding. This means that the complicated classifier makes the accuracy worse.

In contrast, when user adaptation was performed, the accuracy the for decision tree under the "online adaptation" condition in cross validation outperformed that of thresholding. This implies that user adaptation makes the features more general and essential, so overfitting was avoided even when the more complicated classifier (decision tree) was used.

Figure 10 shows the top part of the obtained decision tree, whose depth did not exceed four. The feature at the top was the temporal interval after the user adaptation. This fact also demonstrates that the feature was effective in the decision tree.

6.2 Comparison with Batch Adaptation

We calculated dialogue tempos by using the whole dialogue containing the target utterance. This condition, called "batch adaptation," corresponds to a case where the target user's characteristics have already been obtained. We discuss accuracy under this condition because this can be regarded as an upper limit of user adaptation. Because the accuracies of the batch adaptation were calculated as closed tests, those of the on line adaptation were calculated also as closed tests.

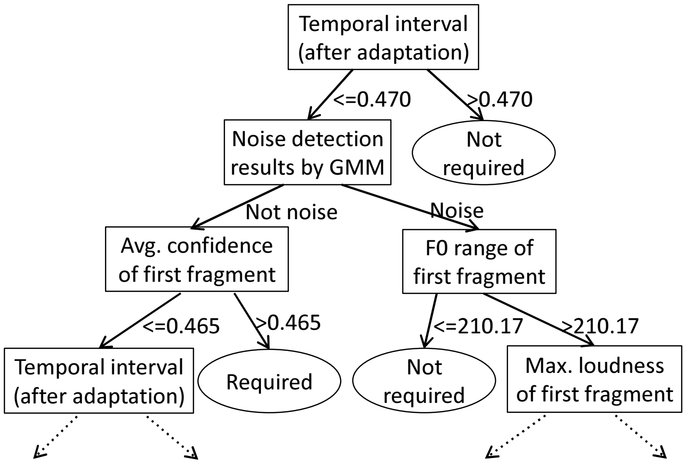
Temporal interval (after adaptation)
<=0.470
>0.470
Noise detection results by GMM
Not required
Not noise
Noise
Avg. confidence of first fragment
F0 range of first fragment
<=0.465
>0.465
<=210.17
>210.17
Temporal interval (after adaptation)
Required
Not required
Max. loudness of first fragment

Figure 10: Obtained decision tree (depth < 4)

Table 3: Classification accuracies by adaptation methods

	Thresholding	Decision trees
No	281/390 (72.1%)	312/390 (80.0%)
Online	294/390 (75.4%)	320/390 (82.1%)
Batch	306/390 (78.5%)	331/390 (84.9%)
Oracle	334/390 (85.7%)	–

Table 3 shows the classification accuracies under the no adaptation and two adaptation conditions. Here, for the simplicity of experiments in the decision tree method, we used the same decision tree (including branching conditions and tree structure) as the batch adaptation, but the temporal interval features were re-estimated over time; that is, the available number of switching pause durations to calculate dialogue tempos increased online. The results show that the accuracies of the batch adaptation were higher than those of online adaptation conditions by 3.1 and 2.8 percentage points for thresholding and decision trees, respectively. This implies that the classification accuracy improves when plenty of utterances from the target user are available.

Furthermore, we investigated an "oracle" condition, where the optimal thresholds were determined for each user by SVM as in Section 4.2. The accuracy was 85.7% (334/390), as also shown in Table 3. This result shows that adaption still has room for improving the accuracy more by capturing the detailed characteristics of each user.

6.3 Convergence Speed of Adaptation

We further investigated the convergence speed of the online adaptation. We conducted the following experiments only for thresholding for simplicity of implementation. The classification accuracy of online adaptation naturally converges into that of batch adaptation when the number of the target

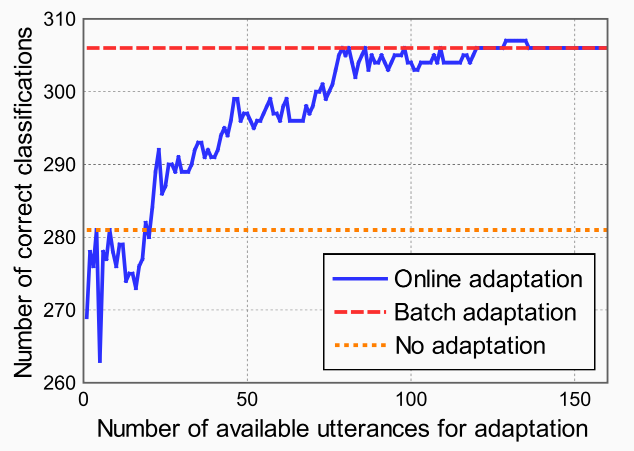

Figure 11: Convergence speed of adaptation (in thresholding)

user's available utterances increases because batch adaptation assumes that all utterances are obtained beforehand. We plotted the classification performance when the number of a target user's available utterances increased to analyze its convergence speed. Here, the levels of performance were calculated as closed tests, similarly with those of the previous section.

Figure 11 shows the number of correct classifications when the number of available utterances for online adaptation increased. The vertical and horizontal axes denote the number of correct classifications and available utterances for the adaptation, respectively. More specifically, the horizontal axis shows that the user's dialogue tempo was calculated using data from the beginning of the dialogue to the x-th utterance. The dashed line and the dotted line represent batch adaptation, i.e., $y = 306$ and no adaptation, i.e., $y = 281$, respectively. Both of these results are listed in Table 3.

We can see that when the number of available utterances was small ($x < 10$), the number of correct classifications was significantly varied and also small (about 275). Correct classification results increased when the available utterances increased and became equivalent to that of batch adaptation after $x = 80$. This shows that the performance converged at about 80 utterances.

These results lead us to the following conclusions. First, when the number of available utterances is small, i.e., less than 10, the classifier should not be adapted because the levels of performance were lower than those under the "no adaptation" condition. Performance does not degrade if we adapt the classifier after about 10 utterances are obtained from the target user. Second, although a one-shot user will probably not make 80 utterances at once, such a number of utterances can be obtained when user IDs are available and when a user's utterances are obtained through several sessions. User IDs can be obtained when the system is used through personal terminals (e.g., cell phones) or when using techniques such as speaker identification.

7. Conclusions and Future Work

We developed a user-adaptive method to classify whether restoration is required for incorrectly segmented utterances by focusing on each user's style of speaking. We empirically showed the correlation between dialogue tempo and appropriate thresholds for temporal intervals between utterance fragments, which are an important feature for the classification. We then investigated classification accuracies by online adaptation of two classifiers: thresholding and decision trees. Results showed that the accuracies improved in both classifiers more than in the baselines using a constant threshold for all users.

Several issues remain as future work to improve the classification accuracy even more. First, we can exploit aspects other than dialogue tempo to represent each user's style of speaking, such as the speaking rate and the frequency of self-repairs. Lexical or semantic features, which were used in previous studies such as (Nakano et al., 1999), can also be used.

Second, we want to adapt features in the feature set other than the temporal interval between two utterance fragments used in this paper. For example, the maximum loudness in the first fragment can be adapted to each user. The F0 range of the first fragment can also be a target of adaptation because some users have habitual intonation at the end of utterances.

Third, the experiments in this paper were conducted using already recorded dialogue data between a human and a system. The user behaviors in this data may have been influenced by the system performance when the data was collected. Therefore, we need to conduct another experiment where a system with the proposed method actually interacts with humans. Other metrics such as user satisfaction and completion time will be helpful to demonstrate the performance.

Finally, variations in speaking styles naturally exist within the same user as well as across users when the system is used repeatedly (Komatani et al., 2009). This occurs especially when the user first tries the system. If much more data per user are made available to analyze, the classification accuracy may improve.

References

Timo Baumann and David Schlangen. Predicting the micro-timing of user input for an incremental spoken dialogue system that completes a user's ongoing turn. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 120–129, 2011.

Linda Bell, Johan Boye, and Joakim Gustafson. Real-time handling of fragmented utterances. In *Proc. NAACL Workshop on Adaption in Dialogue Systems*, pages 2–8, 2001.

A. Benyassine, E. Shlomot, Huan yu 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 Communications Magazine*, 35(9):64–73, 1997.

Iwan de Kok, Dirk Heylen, and Louis-Philippe Morency. Speaker-adaptive multimodal prediction model for listener responses. In *Proc. International Conference on Multimodal Interaction (ICMI)*, pages 51–58, 2013.

Kohji Dohsaka, Atsushi Kanemoto, Ryuichiro Higashinaka, Yasuhiro Minami, and Eisaku Maeda. User-adaptive coordination of agent communicative behavior in spoken dialogue. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 314–321, 2010.

Jens Edlund, Mattias Heldner, and Joakim Gustafson. Utterance segmentation and turn-taking in spoken dialogue systems. In Bernhard Fisseni, Hans-Christian Schmitz, Bernhard Schröder, and

Petra Wagner, editors, *Computer Studies in Language and Speech*, volume 8, pages 576–587. Peter Lang, Frankfurt am Main, 2005.

Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. A prosody-based approach to end-of-utterance detection that does not require speech recognition. In *Proc. IEEE International Conference on Acoustics, Speech & Signal Processing (ICASSP)*, volume 1, pages 608–611, 2003.

Kotaro Funakoshi, Mikio Nakano, Kazuki Kobayashi, Takanori Komatsu, and Seiji Yamada. Non-humanlike spoken dialogue: A design perspective. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 176–184, 2010.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11: 10–18, November 2009.

Mattias Heldner and Jens Edlund. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555 – 568, 2010.

Naoki Hotta, Kazunori Komatani, Satoshi Sato, and Mikio Nakano. Detecting incorrectly-segmented utterances for posteriori restoration of turn-taking and ASR results. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 313–317, 2014.

Kristiina Jokinen and Kari Kanto. User expertise modeling and adaptivity in a speech-based e-mail system. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 87–94, 2004.

Norihide Kitaoka, Masashi Takeuchi, Ryota Nishimura, and Seiichi Nakagawa. Response timing detection using prosodic and linguistic information for human-friendly spoken dialog systems. *Journal of The Japanese Society for Artificial Intellignece*, 20(3):220–228, 2005.

Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. User modeling in spoken dialogue systems to generate flexible guidance. *User Modeling and User-Adapted Interaction*, 15(1):169–183, 2005.

Kazunori Komatani, Tatsuya Kawahara, and Hiroshi G. Okuno. A model of temporally changing user behaviors in a deployed spoken dialogue system. In *Proc. International Conference on User Modeling, Adaptation, and Personalization (UMAP)*, volume 5535 of *Lecture Notes in Computer Science*, pages 409–414. Springer, 2009. ISBN 978-3-642-02246-3.

Kazunori Komatani, Naoki Hotta, and Satoshi Sato. Restoring incorrectly segmented keywords and turn-taking caused by short pauses. In *Proc. International Workshop on Spoken Dialogue Systems (IWSDS)*, pages 27–38, 2014.

Kazunori Komatani, Naoki Hotta, Satoshi Sato, and Mikio Nakano. User adaptive restoration for incorrectly segmented utterances in spoken dialogue systems. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 393–401, 2015.

Akinobu Lee and Tatsuya Kawahara. Recent development of open-source speech recognition engine Julius. In *Proc. APSIPA ASC: Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference*, pages 131–137, 2009.

Mikio Nakano, Noboru Miyazaki, Jun ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. Understanding unsegmented user utterances in real-time spoken dialogue systems. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 200–207, 1999.

Mikio Nakano, Shun Sato, Kazunori Komatani, Kyoko Matsuyama, Kotaro Funakoshi, and Hiroshi G. Okuno. A two-stage domain selection framework for extensible multi-domain spoken dialogue systems. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 18–29, 2011.

Tomoko Ohsuga, Masafumi Nishida, Yasuo Horiuchi, and Akira Ichikawa. Investigation of the relationship between turn-taking and prosodic features in spontaneous dialogue. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*, 2005.

Tim Paek and David Maxwell Chickering. Improving command and control speech recognition on mobile devices: using predictive user models for language modeling. *User Modeling and User-Adapted Interaction*, 17(1-2):93–117, 2007.

Maike Paetzel, Ramesh Manuvinakurike, and David DeVault. ”so, which one is it?” the effect of alternative incremental architectures in a high-performance game-playing agent. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 77–86, 2015.

Antoine Raux and Maxine Eskenazi. Optimizing Endpointing Thresholds using Dialogue Features in a Spoken Dialogue System. In *Proc. SIGdial Workshop on Discourse and Dialogue*, pages 1–10, 2008.

Antoine Raux and Maxine Eskenazi. A finite-state turn-taking model for spoken dialog systems. In *Proc. Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT NAACL)*, pages 629–637, 2009.

Hiroyuki Sakai, Tobias Cincarek, Hiromichi Kawanami, Hiroshi Saruwatari, Kiyohiro Shikano, and Akinobu Lee. Voice activity detection applied to hands-free spoken dialogue robot based on decoding using acoustic and language model. In *Proc. International Conference on Robot Communication and Coordination (ROBOCOMM)*, 2007.

Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyoaki Aikawa. Learning decision trees to determine turn-taking by spoken dialogue systems. In *Proc. International Conference on Spoken Language Processing (ICSLP)*, pages 861–864, 2002.

Ethan Selfridge, Iker Arizmendi, Peter A. Heeman, and Jason D. Williams. Stability and accuracy in incremental speech recognition. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 110–119, 2011.

Gabriel Skantze and Anna Hjalmarsson. Towards incremental speech generation in dialogue systems. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 1–8, 2010.

David Traum, David DeVault, Jina Lee, Zhiyang Wang, and Stacy Marsella. Incremental dialogue understanding and feedback for multiparty, multimodal conversation. In *Intelligent Virtual Agents*, volume 7502 of *Lecture Notes in Computer Science*, pages 275–288. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33196-1.

Tiancheng Zhao, Alan W. Black, and Maxine Eskenazi. An incremental turn-taking model with active system barge-in for spoken dialog systems. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 42–50, 2015.