

User Adaptive Restoration for Incorrectly Segmented Utterances in Spoken Dialogue Systems

Kazunori Komatani[†], Naoki Hotta[‡], Satoshi Sato[‡], Mikio Nakano[¶]

[†] The Institute of Scientific and Industrial Research (ISIR), Osaka University
Ibaraki, Osaka 567-0047, Japan

[‡] Graduate School of Engineering, Nagoya University
Nagoya, Aichi 464-8603, Japan

[¶] Honda Research Institute Japan Co., Ltd.
Wako, Saitama 351-0188, Japan

komatani@sanken.osaka-u.ac.jp

Abstract

Ideally, the users of spoken dialogue systems should be able to speak at their own tempo. The systems thus need to correctly interpret utterances from various users, even when these utterances contain disfluency. 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 classify whether restoration is required or not. We improve the accuracy by adapting the classifier to each user. We focus on the dialogue tempo of each user, which can be obtained during dialogues, and determine the correlation between each user's tempo and the appropriate thresholds for the classification. A linear regression function used to convert the tempos into thresholds is also derived. Experimental results showed that the proposed user adaptation for two classifiers, thresholding and decision tree, improved the classification accuracies by 3.0% and 7.4%, respectively, in ten-fold cross validation.

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 with disfluency, conventional systems basically assume that a user says one utterance with no pause. Systems need to handle utterances by both novice users who speak slowly and experienced users who want the systems to reply quickly.

We propose a method for spoken dialogue systems to interpret user utterances adaptively in terms of utterance units. We adopt an approach based on our a posteriori restoration for incorrectly segmented utterances (Komatani et al., 2014). The proposed system responds quickly while also interpreting utterance fragments by concatenating them when a user speaks with disfluency or speaks slowly with pauses. Another approach for this issue is to adaptively change the parameters of voice activity detection (VAD) for each user during dialogues, but 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, and the system can restore interpretation of user utterances after ASR results are obtained.

Our a posteriori restoration approach needs to classify whether two utterance fragments close in time need to be interpreted together or not, i.e., whether these are two different utterances or a single utterance incorrectly segmented by VAD. If these need to be interpreted separately, the system normally responds to the two fragments 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 it.

Misclassification causes erroneous system responses. If the system incorrectly classifies the restoration as not being required, its response often becomes erroneous because the original user utterance is interrupted in its middle. If the system classifies the restoration as being required even though it 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.

We adapt the classification to each user and show through experiments that the adaptation improves classification accuracy. We focus on the tempo of each user and use it to adapt the classifier. Since 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 the following 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.

These three do not agree, however, in cases of real user utterances. Since the first item is the input information, 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 those studies was to incrementally understand fragmented utterances and determine whether each fragment forms a DA with another. The disagreement 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 incorporating partial ASR results into their model (Raux and Eskenazi, 2009).

Our a posteriori restoration framework mainly considers the former disagreement 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 latter disagreement is partially considered in our framework by

classifying whether to respond to a fragmented utterance or not. Our problem setting relates in part to the one tackled by the above-mentioned studies, in which the system determines more precise timing to respond. Our approach can thus be used together with these studies to improve turn-taking (Kitaoka et al., 2005; Raux and Eskenazi, 2008; Raux and Eskenazi, 2009).

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 contents of the system utterances (Jokinen and Kanto, 2004) and dialogue management (Komatani et al., 2005), pause and gaze duration (Dohsaka et al., 2010), how to respond to a user (e.g., head nods or short vocalization like “uh-huh”) (de Kok et al., 2013), etc. On the other hand, there have been only a few studies 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 both of the above types; its result changes turn-taking, i.e., whether the system responds to fragments or not, and input interpretation, i.e., in which unit the system interprets user utterances. As far as we know, this is the first user adaptation method proposed for the restoration of utterance units.

3 Posteriori Restoration for Incorrectly Segmented Utterances

We first explain how conventional systems respond to an incorrectly segmented utterance. Here, a user utterance is segmented into a pair of utterance fragments denoted as first and second fragments. Given such a pair, one type of conventional system that does not allow barge-ins keeps responding to the first fragment, ignoring the second fragment of the user utterance that follows. Another type of conventional system that allows barge-ins can terminate its response for the first fragment but responds on the basis of an ASR result for the second fragment only.

An outline of our a posteriori restoration process is shown in Fig. 1. When a pair of utterance fragments is close in time, this process is invoked at the timing when the second fragment starts. The process consists of two steps:

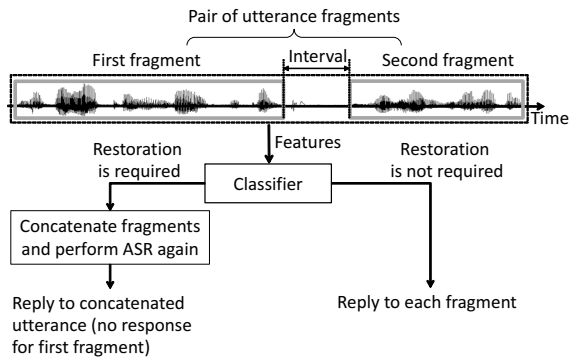


Figure 1: Overview of proposed restoration process.

1. Classify whether a pair of utterance fragments resulted from an incorrect segmentation or not, i.e., whether restoration is required or not.
2. Restore the utterance if it has been incorrectly segmented. The system also restores turn-taking, i.e., terminates its response to the first fragment and waits until the second fragment ends. The aim here is to avoid the system speaking during a user utterance.

If restoration is required, the system performs ASR again after concatenating the fragments to restore the ASR results, which may be erroneous due to incorrect segmentation. The system then responds on the basis of the ASR result for the concatenated fragments after the second fragment ends.

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.

There is a trade-off between the occurrences of erroneous system responses caused by incorrect segmentation and response delay resulting from the restoration. Our approach gives weight to preventing the erroneous responses at the expense of a small delay of system responses. We endeavor to reduce damage stemming from the delay: by producing fillers such as “Well” to prevent unnatural silences (Komatani et al., 2014) and improving implementation to reduce the delay itself.

4 Obtaining Appropriate Thresholds from Dialogue Tempos

The threshold for the temporal interval between a pair of utterance fragments plays a dominant role

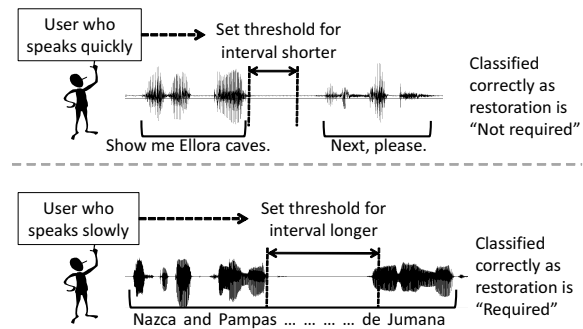


Figure 2: Examples of user-adapted restoration.

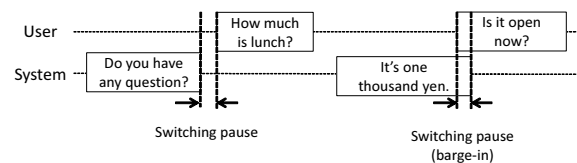


Figure 3: Examples of switching pauses.

in the classification of whether the pair is required to be restored. We assume that appropriate thresholds depend on the way each user speaks. Examples of how the thresholds need to change are given in Fig. 2.

It is assumed that brisk users speak with less disfluency and with shorter pauses. Thus, the threshold needs to be set shorter, which avoids 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).

In contrast, “slow” users often speak with long pauses during their utterances. In this case, the threshold needs to be set longer, which enables 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 times between when a system finishes speaking and a user starts speaking, as depicted in Fig. 3. We calculate this per user from the beginning of the dialogue. The duration of a switching pause becomes negative when the user barges in, i.e., the user starts speaking during a system utterance. Although 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 here, it can be accumulated per user when a user ID can be obtained (e.g., mobile phones, in-car interfaces, etc.).

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” here we mean that the threshold can classify whether the restoration is required or not with high accuracy. The restoration for a 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 the restoration is required or not. We used the SMO module in Weka (version 3.6.9) (Hall et al., 2009) as an SVM implementation. The parameters were set to its default values, e.g., its kernel function was polynomial. The SVM is able to set the 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 labels in 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 system that introduces the world heritage sites (Nakano et al., 2011). In total, speech data of 35 participants were recorded. Each participant engaged in 8-minute dialogues four times. Participants were not given any special instructions prior to or during the dialogues.

We used 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), with each fragment longer than 0.80 seconds. This was because our target is originally a single utterance, and we regard pairs whose intervals are greater than 2.00

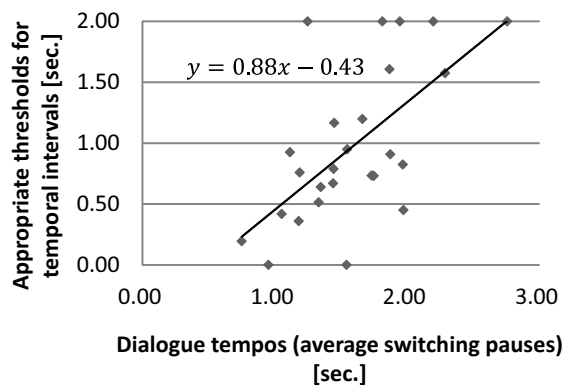


Figure 4: Correlations between appropriate thresholds and dialogue tempos per participant.

seconds and which are very short as not such an utterance (Komatani et al., 2014).

We obtained 3,099 utterances from the 26 participants. The data included 390 utterance pairs that satisfy the above conditions to possibly be a single utterance. We manually assigned the labels of whether the pair is a single utterance in accordance with the procedure in (Hotta et al., 2014). Since 240 pairs were originally single utterances and 150 pairs were not, the classification accuracy by the majority baseline was 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 3,099 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, since such large values simply indicate that the participant was thinking deeply. These values were determined experimentally.

Figure 4 plots the correlation, where the x-axis denotes the dialogue tempos and the y-axis denotes the appropriate thresholds, both in seconds. The correlation coefficient was 0.63. The linear regression function is derived as

$$y = 0.88x - 0.43. \quad (1)$$

This function is used in the next section for obtaining appropriate thresholds from the dialogue tempos per participant.

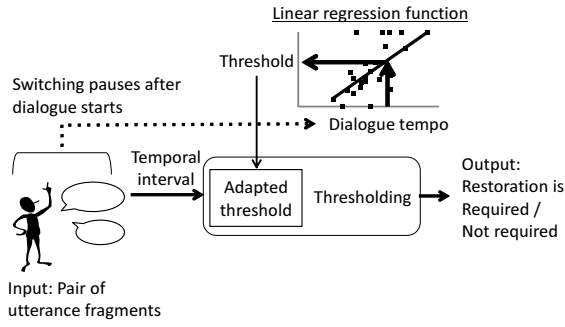


Figure 5: User-adapted classification in thresholding.

5 Adapting Classifiers for Restoration to Users

We investigate whether the correlation between dialogue tempos and appropriate thresholds is helpful or not. The correlation is used to derive the user-adaptive threshold from the user’s dialogue tempo and thus to improve classification accuracy for whether restoration is required or not. First, the system obtains the appropriate thresholds 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 examine user adaptation for two classification methods: thresholding and decision tree.

5.1 Thresholding

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

The process flow of thresholding with user adaptation is shown in Fig. 5. 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 obtains a threshold value corresponding to the tempo by the linear regression function. The system then classifies whether the restoration is required or not by 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 Tree

We also use a decision tree, which is a more complicated classifier than thresholding. We show that

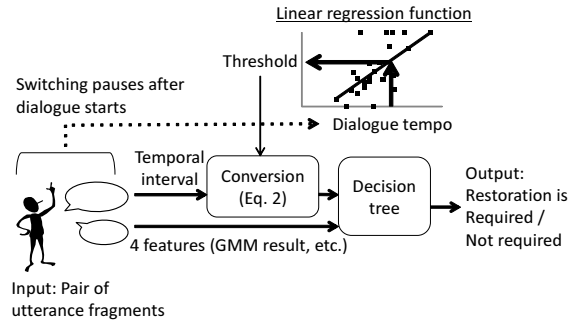


Figure 6: User-adapted classification in decision tree.

user adaptation is also effective in this case.

The process flow of the decision tree with user adaptation is depicted in Fig. 6. 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 by a Gaussian mixture model (GMM), F0 range of the first fragment, and maximum loudness in the first fragment.

The user adaptation is performed by converting the temporal interval only out of these five features. The interval is converted in both the training and classification phases in the decision tree learning. Instead of adapting the thresholds 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 relatively convert the feature values for the interval in accordance with each user, and thus enabling the system to classify adaptively to users with a constant threshold. Specifically, we use ratios between the threshold values of a 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} :

$$\hat{I}_{ij} = I_{ij} \times \frac{T_0}{T_j}, \quad (2)$$

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, which was the average interval of all users.

Our aim with this conversion is as follows. The correlation depicted in Fig. 4 shows that thresh-

Table 1: Deviation of parameters in linear regression function.

	a	b
Avg.	0.883	-0.431
Std. dev.	0.034	0.057

olds 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 used in decision tree learning.

6 Experimental Evaluation

We investigated whether the 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 is assumed to be always available.

6.1 Performance of User Adaptation

We investigated the classification accuracy for the two methods, thresholding and decision tree, as discussed in Section 5.

Experiments were conducted under two conditions: closed test and cross validation. In the closed tests, we used the same data in both adaptation and test phases, and under the cross-validation condition, we set each user as a unit.

Specifically, in thresholding, we extracted the data of one user from the data of all 26 participants, derived linear regression functions from the data of the 25 participants, and calculated the classification accuracy using the data of the one separated user. This process was repeated 26 times. During this cross validation, we investigated the deviations of the two parameters of the linear regression function $y = ax + b$, shown in Eq. (1). The results 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 data. This indicates that the linear regression function only depends only a little on the training sets and thus has more generality than the decision tree. This is because the number of parameters is small (only two).

As a result of this stability of the parameters, for simplicity of experimentation, we assumed that the linear regression function was known under the decision tree learning condition, that is, that the dialogue tempos of each user can be converted to the intervals, which are used in Eq. (2).

6.1.1 Thresholding Adapted to Users

Classification accuracies in thresholding are listed in the left column of Table 2. The condition “no adaptation” denotes the case where 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 manner as discussed in Section 4.2.

The results show that the user adaptation improved classification accuracies by 3.3 and 3.0 percentage points for the closed test and cross-validation conditions, respectively. We can also see that the accuracies of the closed test and cross-validation conditions were almost equivalent under both adaptation conditions (“no” and “online”). This suggests that no overfitting occurred in these cases and thus a similar performance will 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 condition “no adaptation” denotes normal decision tree learning, that is, no feature values were converted using Eq. (2). These results show that the user adaptation improved the accuracies by 2.1 and 7.4 percentage points for the closed test and cross-validation conditions, respectively. The difference in the cross-validation condition was statistically significant by the McNemar test ($p = 3.2 \times 10^{-4}$).

We can see that the accuracies of the cross-validation conditions were lower than those in the closed test. This is because a decision tree has many more parameters to be trained than thresholding, and thus 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 the cross validation was lower than that of the thresholding. This means that the complicated classifier makes the accuracy worse.

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%)

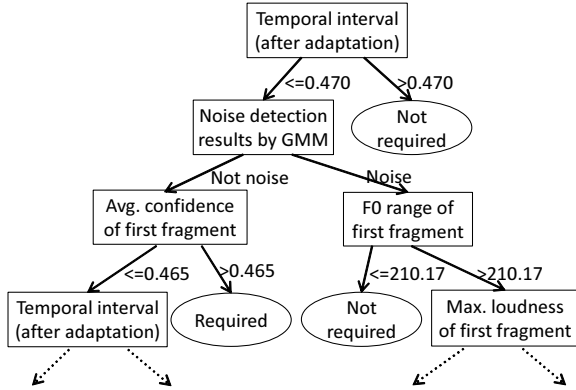


Figure 7: Obtained decision tree (depth < 4).

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

Figure 7 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 confirms that the feature was effective in the decision tree.

6.2 Comparison with Batch Adaptation

In all experiments discussed thus far, each user’s dialogue tempo was calculated by using the duration of switching pauses from the beginning of the dialogue until the target utterance. We call this “online adaptation”.

We also virtually calculated dialogue tempos by using the whole dialogue containing the target utterance. This condition, called “batch adaptation”, virtually assumes that the dialogue data of a target user has been sufficiently obtained beforehand. It thus corresponds to a case where the target user’s characteristics have already been obtained. We discuss its performance under this condition, since this can be regarded as an upper limit of user adap-

Table 3: Classification accuracies by adaptation methods.

	Thresholding	Decision tree
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%)

tation. Since performances of the batch adaptation were calculated as the closed tests, those of the online adaptation were calculated also as the closed tests.

Table 3 shows the classification accuracies under the no adaptation and two adaptation conditions. Here, for simplicity of experiments under the decision tree condition, we assume that the shapes of decision trees used in the online adaptation were the same as the batch adaptation; 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 online adaptation conditions by 3.1 and 2.8 percentage points for thresholding and decision tree, respectively. This implies that the classification performance is unstable in online adaptation when the number of available utterances of the target user is small.

6.3 Convergence Speed of Adaptation

We further investigated the convergence speed of the online adaptation. We conducted the following experiments only for thresholding because of the simplicity of implementation. It is natural that the classification accuracy of the online adaptation converges into that of batch adaptation when the number of a target user’s available utterances increases, as batch adaptation assumes that all utterances are obtained beforehand. We plot the classification performance when the number of a target user’s available utterance increased to analyze its convergence speed. Here, the performances were calculated as the closed tests, similarly with the previous section.

Figure 8 shows the number of correct classifica-

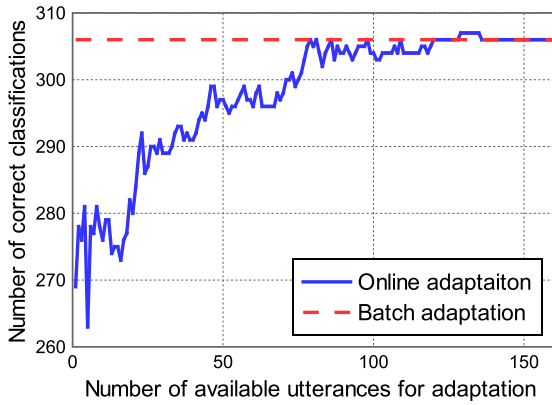


Figure 8: Convergence speed of adaptation (in thresholding)

tions when the number of available utterances for the online adaptation increased. 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 by using data from the beginning of the dialogue to the x -th utterance. The dashed line at the upper part of the graph denotes the case of batch adaptation, i.e., $y = 306$, as 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). The 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 with 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, it is better not to adapt the classifier because the performances were lower than under the “no adaptation” condition, whose number of correct classifications was 281, as shown in Table 3. Performance does not degrade if we adapt the classifier after about 10 utterances are obtained from the target user. Second, although it is unlikely that a one-shot user will make 80 utterances at once, it is possible to obtain such a number of utterances when user IDs are available and 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 by using techniques such as speaker identification.

7 Conclusion

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 adapting two classifiers: thresholding and decision tree. 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 intend to exploit aspects other than the dialogue tempos based on switching pauses to represent each user’s style of speaking, such as 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 together. Second, we want to adapt features other than the temporal interval between two utterance fragments used in this paper. For example, the maximum loudness of the first fragment can be adapted to each user. In addition, since some users have habitual intonation at the end of utterances, this can also be a target of adaptation. Third, the experiments in this paper were conducted using already recorded dialogue data between a human and a system. It is possible that the user behaviors in this data were influenced by the system performance when the data was collected. We therefore 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 verify the performance. Finally, variations of speaking styles 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 starts using the system, i.e., novice users. We need much more data per user to analyze this, but it is possible that such a consideration can improve the classification accuracy.

Acknowledgments

This work was partly supported by the Casio Science Promotion Foundation.

References

- Linda Bell, Johan Boye, and Joakim Gustafson. 2001. Real-time handling of fragmented utterances. In *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, pages 2–8.
- Iwan de Kok, Dirk Heylen, and Louis-Philippe Morency. 2013. Speaker-adaptive multimodal prediction model for listener responses. In *Proc. International Conference on Multimodal Interaction (ICMI)*, pages 51–58.
- Kohji Dohsaka, Atsushi Kanemoto, Ryuichiro Higashinaka, Yasuhiro Minami, and Eisaku Maeda. 2010. 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.
- Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. 2003. 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.
- Kotaro Funakoshi, Mikio Nakano, Kazuki Kobayashi, Takanori Komatsu, and Seiji Yamada. 2010. Non-humanlike spoken dialogue: A design perspective. In *Proc. Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 176–184.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November.
- Naoki Hotta, Kazunori Komatani, Satoshi Sato, and Mikio Nakano. 2014. 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.
- Kristiina Jokinen and Kari Kanto. 2004. 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.
- Norihide Kitaoka, Masashi Takeuchi, Ryota Nishimura, and Seiichi Nakagawa. 2005. Response timing detection using prosodic and linguistic information for human-friendly spoken dialog systems. *Journal of The Japanese Society for Artificial Intelligence*, 20(3):220–228.
- Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. 2005. User modeling in spoken dialogue systems to generate flexible guidance. *User Modeling and User-Adapted Interaction*, 15(1):169–183.
- Kazunori Komatani, Tatsuya Kawahara, and Hiroshi G. Okuno. 2009. 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.
- Kazunori Komatani, Naoki Hotta, and Satoshi Sato. 2014. Restoring incorrectly segmented keywords and turn-taking caused by short pauses. In *Proc. International Workshop on Spoken Dialogue Systems (IWSDS)*, pages 27–38.
- Mikio Nakano, Noboru Miyazaki, Jun ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. 1999. Understanding unsegmented user utterances in real-time spoken dialogue systems. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 200–207.
- Mikio Nakano, Shun Sato, Kazunori Komatani, Kyoko Matsuyama, Kotaro Funakoshi, and Hiroshi G. Okuno. 2011. 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, June.
- Tim Paek and David Maxwell Chickering. 2007. 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.
- Antoine Raux and Maxine Eskenazi. 2008. Optimizing Endpointing Thresholds using Dialogue Features in a Spoken Dialogue System. In *Proc. SIGdial Workshop on Discourse and Dialogue*, pages 1–10.
- Antoine Raux and Maxine Eskenazi. 2009. 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.
- Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyooki Aikawa. 2002. Learning decision trees to determine turn-taking by spoken dialogue systems. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 861–864.