Delay Mitigation for Backchannel Prediction in Spoken Dialog System

Amalia Istiqlali Adiba, Takeshi Homma, Dario Bertero, Takashi Sumiyoshi, and Kenji Nagamatsu

Abstract To provide natural dialogues between spoken dialog systems and users, backchannel feedback can be used to make the interaction more sophisticated. Many related studies have combined acoustic and lexical features into a model to achieve better prediction. However, extracting lexical features leads to a delay caused by the automatic speech recognition (ASR) process. The systems should respond with no delay, since delays reduce the naturalness of the conversation and make the user feel dissatisfied. In this work, we present a prior prediction model for reducing response delay in backchannel prediction. We first train both acoustic- and lexical-based backchannel prediction models independently. In the lexical-based model, prior prediction is necessary to consider the ASR delay. The prior prediction model is trained with a weighting value that gradually increases when a sequence is closer to a suitable response timing. The backchannel probability is calculated based on the outputs from both acoustic- and lexical-based models. Evaluation results show that the prior prediction model can predict backchannel with an improvement rate on the F1 score 8% better than the current state-of-the-art algorithm under a 2.0-s delay condition.

A. I. Adiba · T. Homma (✉) · D. Bertero · T. Sumiyoshi · K. Nagamatsu
Hitachi, Ltd., Tokyo, Japan
e-mail: takeshi.homma.ps@hitachi.com

A. I. Adiba
e-mail: amalia.adiba.dw@hitachi.com

D. Bertero
e-mail: dario.bertero.zt@hitachi.com

T. Sumiyoshi
e-mail: takashi.sumiyoshi.bf@hitachi.com

K. Nagamatsu
e-mail: kenji.nagamatsu.dm@hitachi.com

L. F. D'Haro et al. (eds.), *Conversational Dialogue Systems for the Next Decade*, Lecture
Notes in Electrical Engineering 704, https://doi.org/10.1007/978-981-15-8395-7_10

1 Introduction

Spoken dialog systems (SDS) are increasingly deployed in our daily lives in the form of smart phones, smart speakers, car navigation systems, and more. These systems can conduct simple tasks or answer questions. For this type of domain, the systems generally assume that a user makes one utterance per turn. This assumption is obviously far from natural, however, as humans often make several utterances in one turn during a conversation, the listener usually provides feedback, known as a backchannel. In SDS, a backchannel is defined as any kind of feedback a system provides to a user, including short phrases ('uh-huh,' 'right,' etc.) and nods. Backchannels can express sympathy and acknowledge, and they can encourage users to keep talking and let them know that the system has received the information. Without backchannels, users would be concerned about whether the system is still listening to the conversation or not. Therefore, if an SDS can predict the backchannel and generate a backchannel naturally, as humans do, it will be better able to converse with users.

When designing an SDS with backchannel functions, the backchannel models must accurately detect the backchannel and predict the suitable time to respond. The inputs of backchannel detection models come from user speech that has been converted into acoustic features such as loudness, pitch, and MFCC. The input could also include lexical features, e.g., the output of an automatic speech recognition (ASR) system. According to previous work, lexical features are the most informative features and provide better accuracy than acoustic features [11, 18]. However, accurate ASR needs time for processing, which delays the system responses. In addition, many SDS applications use cloud-based ASR, and in these cases, the delay of system responses will be caused not only by ASR processing but also by the latency of network communication. Figure 1 shows an example of this delay problem. Response delays of a few milliseconds might be acceptable as 'thinking time'; after all, response delays sometimes also occur in human-human conversation, such as during question-answering or reservation domain tasks. According to an earlier report [19], in human-robot conversations, the maximum user preference for a system to generate a response is one second. However, backchannel feedback is different depending on turn response. Delayed feedback might cause an unsophisticated response, as backchannel is a kind of a reflex action. If the SDS does not make a backchannel response at an appropriate time, it will produce unnatural conversation, possibly making the user feel dissatisfied. Thus, the SDS must start to give feedback as soon as the user has finished speaking.

To resolve these issues, we propose a modelling method that can mitigate the response delay problem in backchannel detection. Our contributions in this work are summarized as follows.

- We propose a backchannel prior prediction model. Prior prediction means that the model predicts backchannel events within only the available ASR output, which consists of the words derived from the beginning of the utterance to the several seconds before the final ASR outputs occur.

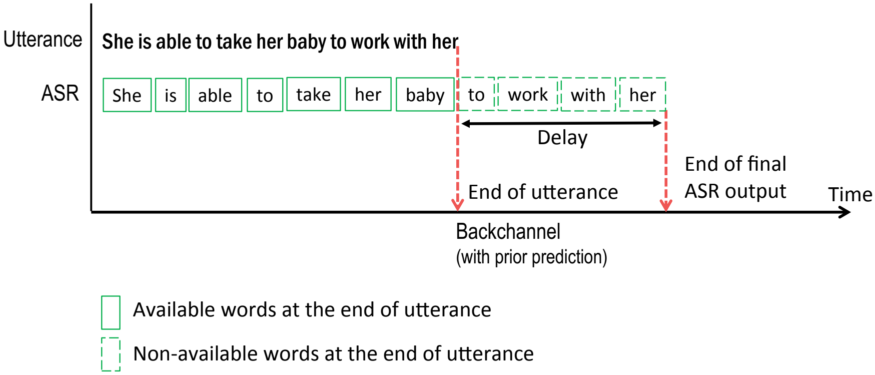

Fig. 1 Response delay caused by ASR in the spoken dialog system. Our proposed prior prediction mitigates the delay and predicts backchannel earlier than the end of the final ASR output.

- We propose an early loss (EL) function, in which a weighting on the loss value gradually increases when a word is closer to the final ASR outputs.
- We show that the prior detection model successfully predicts the backchannel events before the final ASR outputs.
- We show that our proposed prior detection model achieves better performance in the prediction of suitable backchannel timing than the baseline method in which the model is trained without the EL function.
- We show it is not necessary to train the model in various delay conditions independently, since even under different conditions, our EL-based model delivered better predictions.

2 Related Work

Prior work on backchannels has used different types of prediction methods and a variety of input features. These input features include acoustic, lexical, and visual features such as gaze movement.

One main task is to predict backchannel feedback from user's speech. Most related work has taken a rule-based approach. Ward and Tsukahara [22] proposed a method with acoustic features to predict a backchannel and reported that a few millisecond regions of low pitch are a good predictor of backchannel feedback. Truong et al. [21] developed a rule-based backchannel prediction model using power and pitch information. Other work has used a classifier to train the model. Morency et al. [14] proposed a multimodal approach in which acoustic, lexical, and eye gaze data are used as the features and the model is trained with a hidden Markov model (HMM) or conditional random fields (CRFs). Training a prediction model in a continuous manner using the LSTM neural network has also been proposed by [17, 18], who evaluated the method with various input features and reported that the performance with both lexical (e.g., part of speech, word) and acoustic (e.g., power, pitch) features was significantly better than with just acoustic features. They also proposed a prediction point within a predefined window length and reported that the performance peaks at 1.5 s of the window. By using a predefined window as a prediction point, they could also predict backchannels that appear in the middle of an utterance. However, choosing a predefined area might produce unnatural backchannel feedback, as the time gap between the detection point and the backchannel timing can vary significantly.

The other task is to detect speaking acts (including backchannel) [15]. Predicting different types of backchannel has also been proposed [11]. Skantze [20] proposed a continuous model to predict upcoming speech activity in a future time window. They categorized onset prediction into SHORT and LONG, where SHORT utterances can be considered as backchannel. The prediction of turn-taking with multitask learning by combining backchannel and filler prediction has also been proposed [7]. A general study on the correlations or synchrony effect in the prosody of backchannels also exists [10].

In a practical spoken dialog system, the ASR needs a certain length of processing time to output lexical features from speech data. Many researchers proposed backchannel prediction methods using the lexical features. However, most of the researchers did not discuss the influence of ASR processing time to obtain the lexical features. A few researchers conducted research on backchannel prediction using lexical features under the assumptions; ASR gives the lexical features with no delay [17, 18], or ASR takes a constant and short 100ms processing time to output the lexical features [20].

To predict backchannels using lexical features, the system must wait until ASR outputs the lexical features. Hence, the timing of backchannel feedbacks is delayed by the ASR processing time. Moreover, the ASR processing time will be varied depending on characteristics of input speech signals. To generate backchannels at an appropriate timing, the backchannel prediction method must take variable ASR processing time into account.

To the best of our knowledge, our paper is the first to investigate how to accurately predict backchannel based on lexical features considering the variable delay caused by ASR. Our work contributes to reducing the delay of the backchannel responses of the spoken dialog system.

3 Proposed Method

Backchannel prediction is a task that predicts whether the current user's utterance is a backchannel point or not. We extract acoustic and lexical features from an input utterance and feed them to the model to predict the output. The proposed method introduces prior prediction in which the backchannel will be detected several seconds before the end of an utterance. The model is trained based on LSTM-RNN for the

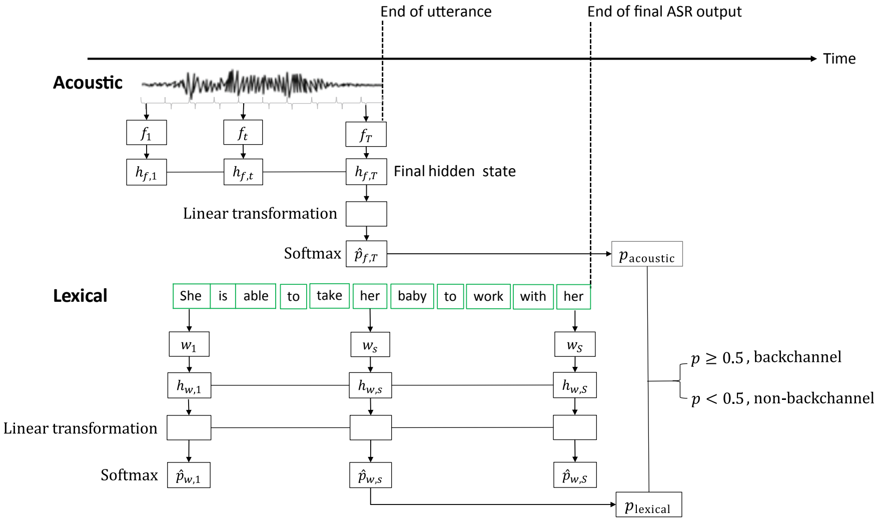

Fig. 2 Neural network architecture for backchannel prediction.

sequential information. We feed acoustic and lexical features to the network and train both models independently. In this section, we explain the overall architecture of the system for the backchannel prediction. We then explain the early loss (EL) function, which is the primary contribution of this study.

3.1 Neural Network for Classification

In a real-life scenario, a system receives both audio channel and ASR outputs. However, the corresponding words come several seconds after the end of an utterance. This means that the audio and the words do not always correspond to each other, and the difference between the time at the end of an utterance and the time at the final ASR output results is a response delay. Let us assume that the acoustic extraction process is not producing any delay. Therefore, prior prediction is not necessary in the acoustic case. In the lexical case, prior prediction is necessary to mitigate the response delay. Because the treatment of the acoustic and lexical cases is different, we train both models independently.

In this study, we trained the backchannel prediction model using an LSTM-RNN-based model [8]. The overview of our system is depicted in Fig. 2. In LSTM, a memory cell is used to store information over time. For lexical features, a token represents word outputs from ASR (w_1, w_2, ..., w_S), where S is the token number at the final ASR output. For acoustic features, a token represents a feature vector of each frame (f_1, f_2, ..., f_T), where T is the token number at the end of the utterance. Standard LSTM takes a sequence of tokens as input and then outputs a sequence of hidden representations $(h_{f,1}, h_{f,2}, ..., h_{f,T})$ for the acoustic-based model and $(h_{w,1}, h_{w,2}, ..., h_{w,S})$ for the lexical-based model. The input layer consists of the extracted features for a corresponding utterance. The hidden layer $h_{f,T}$ or $h_{w,s}$ is used to predict the backchannel probability outputs $\hat{p}_T$ or $\hat{p}_s$, respectively, where s is the latest token number within the words outputted from ASR at the timing of the end of utterance. In our case, the target output is the probability of backchannel. The softmax function is used to compute the probability at the i-th utterance. Finally, the backchannel's probability (p) is calculated as

$$p = \lambda \hat{p}_T + (1 - \lambda)\hat{p}_s, \tag{1}$$

where $\hat{p}_T$ is the backchannel probability from the acoustic-based model, $\hat{p}_s$ is the probability from the lexical-based model, and $\lambda \in [0, 1]$ is a hyper-parameter to control the weight for each model's output.

The loss is calculated only at the last token with a cross-entropy loss function for the acoustic-based model. In the other case, for the lexical-based model, the loss is calculated with an EL function. To clarify the effect of our EL function, we simplify the network and train the model with unidirectional LSTM for both the acoustic and lexical models. The label used for training the model is a one-dimensional binary label, which represents whether the corresponding utterance is a true backchannel (label = 1) or a false one (label = 0). The output of the model is a probability score for backchannel within 0 to 1.

3.2 *Early Loss (EL) Function for Prior Prediction*

Our prior prediction model plays two main roles. The first role is to mitigate delays in the extraction of input features. Let us consider a case where backchannel prediction is done using the lexical features, i.e., ASR outputs. After a user says the final word of an utterance, the ASR outputs this word several seconds after the user has said it. This "several seconds" is the delay we focus on. If the SDS must generate a backchannel response to the user as soon as the end-of-utterance has occurred, the backchannel prediction must be done using only the available input features derived from the beginning of the utterance to the "several seconds" before the end-of-utterance occurs. The prior prediction model can accomplish this functionality.

The second role of the prior detection model is to emphasize the end part of a sequence. We assume the SDS makes a backchannel only when the user stops speaking, even though in human-human dialogue a backchannel can happen anytime. The reason for this assumption is that we want to focus on backchannels as a confirmation function in which users usually take a brief pause to get confirmation about whether the system is still listening to the conversation or not. Therefore, we define the ground truth of a backchannel to be the end of the sequence. A simple way to predict backchannel using only the available input features is to calculate the loss for each token in a sequence with the cross-entropy loss function which has uniform weighting. However, this method leads to a higher chance of a false positive prediction, since the loss of the first token has an equal weight with a token number in the final ASR output. Intuitively, failing to predict a backchannel at a token very close to token number S should receive a higher weight than failing to predict a backchannel at a token far from token number S. Therefore, our strategy is to adaptively modify the weight value depending on how early the model predicts the backchannel. To overcome this challenge, we set the loss at a token very close to token number S to be higher than the loss at a token far from token number S in a sequence. This weighting loss idea originally comes from an earlier method for driver activity anticipation [9]. Chan et al. [3] applied the same idea for anticipating accidents in a dashcam video, which achieved accident anticipation about two seconds before it occurred.

In this work, we consider two types of utterance: those that do not encourage a backchannel response (non-BC utterance), and those that do (BC utterance). For a non-BC utterance, the system should not generate a backchannel feedback in which prior prediction is not necessary. Conversely, for a BC utterance, the system should generate a backchannel several seconds before the end of the final ASR outputs to mitigate response delay. Following [9], we use different loss calculation methods for the BC utterance (**positive**) and the non-BC utterance (**negative**). The loss for a negative utterance is a standard cross-entropy loss. In contrast, to calculate the loss function of a positive utterance, we multiply a weighting function by the standard cross-entropy loss; the weighting function gradually increases when a token is closer to token number S. Finally, the loss function is calculated as follows:

For the **positive** case:

$$L_p = \sum_{s=1}^{S} -e^{-(S-s)} \log(\hat{p}_s), \tag{2}$$

For the **negative** case:

$$L_n = \sum_{s=1}^{S} - \log(1 - \hat{p}_s), \tag{3}$$

where $\hat{p}_s$ is the backchannel probability at token s.

4 Experiments

4.1 Dataset

We use the Switchboard dataset [6] to model backchannel prediction. This dataset consists of English conversations between participants who were asked to discuss a specific topic that was chosen randomly from 70 possibilities. We use annotation from the Switchboard Dialog Act Corpus (SwDA),[1] as we require dialogue act labels (e.g., backchannel, statement) for the annotation. However, the SwDA corpus only maps the dialogue acts with lexical and turn information; it does not map dialogue acts to timing information in the original data of the Switchboard dataset. Because we are interested in delay mitigation, we combine the SwDA corpus with the NXT Switchboard corpus [2] to obtain utterance timing information. With the NXT Switchboard corpus, we can access turn, dialogue act, lexical, utterance, and even word timestamp information within one framework. We exclude utterances in the dataset that do not have timing information.

Previous works have defined an utterance on the basis of pauses [12] or turn shift (speaker change) [1, 13]. We follow the work of [13] and represent a conversation between two speakers as a sequence of utterances that has been sorted based on the start-talk time information available in the corpus $(u_1, u_2, \cdots, u_I)$, where u_i is the i-th utterance in the conversation. We also use dialogue act in corpus $(da_1, da_2, ..., da_I)$, where da_i is the dialogue act for the i-th utterance in the conversation. The backchannel label will be defined as true if the following two conditions are met:

- da_{i+1} is a backchannel
- u_i and u_{i+1} have a different speaker

The final dataset that we use contains 114,315 utterances, with 13.62% of them labelled as true backchannels.

4.2 Features

Following the work in [16], we extract 21 types of acoustic features with 88 dimensions in total using a eGeMAPs [4] configuration with the OpenSmile toolkit [5]. These features include power, pitch, and MFCC. Instead of part-of-speech (POS), we use words in the manual transcripts as lexical features. Note that this is not the ideal approach, as the ASR system might provide different outputs than the transcripts. Lexical features always start from [silb], which is a special word at the beginning of sentence, followed by the first word. We convert words into word indices, where each of the index numbers corresponds to an individual word. The word embedding is then used as input for our model architecture. We found that our dataset in the fastText model trained with the Common Crawl dataset[2] had the smallest number of unknown words. Thus, the word embedding function was conducted using the fastText model with 300 dimensions.

[1] https://github.com/cgpotts/swda.

[2] https://fasttext.cc/.

Table 1 Simulated example of available words in an utterance for various lengths of delay.

Delay (s)	Timestamp for each word											
	0.16	0.31	0.69	0.85	1.08	1.38	1.57	1.60	1.83	2.02	2.63	2.95
0	we	were	I	was	lucky	too	that	I	only	have	one	brother
0.5	we	were	I	was	lucky	too	that	I	only	have		
1	we	were	I	was	lucky	too	that	I	only			
1.5	we	were	I	was	lucky	too						
2	we	were	I	was								

4.3 Experimental Setup

We evaluated the model using 5-fold cross-validation: one fold for the test dataset, 70% of the remaining folds for the training dataset, and 30% for the validation dataset. Each dialogue conversation was split to ensure that the same speaker did not appear in the training, test, and validation sets. For training, we increased the number of positive utterances up to the same as negative utterances by using random oversampling in the training set. We optimized the loss function using an Adam optimizer with a learning rate of 0.0001. At the end of each epoch, we computed the macro F1 score on the validation set. We ran for a maximum of 100 epochs and stopped training if there was no improvement in the validation macro F1 scores for five consecutive epochs. The minibatch size was 10. We took the highest validation performance and applied it to the test data for evaluation. We trained a one-layer LSTM-RNN model using the data. The size of hidden layer is 64 units in the acoustic-based model, and 128 units in the lexical-based model. Afterwards, we evaluated the models with a manipulated utterance for each length of delay. We set λ to 0.5. All parameters were determined on the basis of the best macro F1 score on the validation dataset.

4.4 Simulation Under Various Delay Condition

In real-time conditions, the available ASR output will differ depending on the length of delay and on the duration time for each word in an utterance. For example, the word ‘conversation’ might have a longer duration than ‘yes’. To evaluate the model under various delay conditions, we manipulated the utterance text in the test dataset. The timestamp for each word in the i-th utterance in the conversation is $(d^i_1, d^i_2, ..., d^i_S)$, where S is the number of words from the beginning of the utterance up to the end of the final ASR output (see Fig. 1). Given that the ASR delay is α (in seconds), the available words in the i-th utterance at the end of the utterance are the words in which the timestamp is lower than or equal to $d^i_S - \alpha$. A manipulated utterance for each length of delay is shown in Table 1 as an example.

4.5 Baseline Models

For comparison to our proposed model, we use two baselines configuration from related state-of-the-art studies.

4.5.1 LSTM-RNN Prediction Using Acoustic and POS Combination Features

We re-implement LSTM-RNN originally proposed by Roddy et al. [16], where they used LSTM-RNN for three prediction tasks: prediction at pauses, prediction at onsets, and prediction at overlap. Here, we re-implement only prediction at onsets, which represents a prediction of the length of the next utterance. Roddy et al. stated that SHORT utterances can be considered as backchannels. They used various features (e.g., acoustic, word, and part-of-speech (POS)) and then compared the performance for each feature type. They reported that the best performance was achieved when they combined acoustic and POS features to train the model. Given an input utterance, we implement their method to predict whether the length of the next utterance will be SHORT or not. We follow their setup parameters and implement them for our dataset.

4.5.2 Time-Asynchronous Sequential Networks (TASN)

We re-implement TASN from [12]. In TASN, each feature (word and acoustic) is individually fed into a sequential network. Afterwards, both final hidden representations are concatenated and used to directly predict the output. The original task in their work was for end-of-turn prediction. However, we implement their architecture for backchannel prediction. In this method, the loss is calculated only at the last token for each feature with a cross-entropy loss function.

4.5.3 Lexical Without EL Function

In this method, the loss was calculated as follows:

For the **positive** case:

$$L_p = \sum_{s=1}^{S} -\log(\hat{p}_s), \tag{4}$$

For the **negative** case: same as Eq. (3).

We trained the model with the available word data for five lengths of delay (0.0, 0.5, 1.0, 1.5, and 2.0 s). All models were trained independently.

Table 2 Backchannel prediction performance under different length delay conditions. Results shown are macro-averaged values across positive and negative classes. All the models were trained with a 0.0-s delay condition.

(a) Evaluation under 0.0-s delay condition

Features	Loss function	Precision	Recall	F1
Acoustic	w/o EL	56.36	**63.43**	49.26
Lexical	w/o EL	56.02	57.34	56.01
Lexical	with EL	57.19	57.20	57.17
Acoustic + POS	w/o EL	**62.98**	56.50	53.23
TASN	w/o EL	56.78	56.55	56.55
Acoustic + lexical	w/o EL	57.05	54.94	55.47
Acoustic + lexical	with EL	59.58	57.43	**58.06**

(b) Evaluation under 0.5-s delay condition

Features	Loss function	Precision	Recall	F1
Lexical	w/o EL	56.77	55.54	55.74
Lexical	with EL	56.03	56.76	55.97
Acoustic + POS	w/o EL	53.62	54.66	52.10
TASN	w/o EL	51.88	53.03	51.19
Acoustic + lexical	w/o EL	**57.65**	53.25	53.42
Acoustic + lexical	with EL	57.60	**56.93**	**57.18**

(c) Evaluation under 1.0-s delay condition

Features	Loss function	Precision	Recall	F1
Lexical	w/o EL	56.43	53.98	52.90
Lexical	with EL	55.42	55.13	55.26
Acoustic + POS	w/o EL	54.40	56.53	51.84
TASN	w/o EL	50.43	50.76	48.88
Acoustic + lexical	w/o EL	**57.95**	52.32	51.86
Acoustic + lexical	with EL	55.68	**56.71**	**55.96**

(d) Evaluation under 1.5-s delay condition

Features	Loss function	Precision	Recall	F1
Lexical	w/o EL	55.82	53.04	50.66
Lexical	with EL	55.25	54.30	**54.47**
Acoustic + POS	w/o EL	54.70	**58.93**	50.62
TASN	w/o EL	50.11	50.17	48.10
Acoustic + lexical	w/o EL	**58.87**	52.72	51.01
Acoustic + lexical	with EL	56.29	53.77	52.96

(e) Evaluation under 2.0-s delay condition

Features	Loss function	Precision	Recall	F1
Lexical	w/o EL	**58.04**	51.20	49.50
Lexical	with EL	56.30	52.79	50.02
Acoustic + POS	w/o EL	51.86	**54.92**	48.08
TASN	w/o EL	49.90	49.84	47.46
Acoustic + lexical	w/o EL	58.00	52.02	49.41
Acoustic + lexical	with EL	56.53	53.47	**52.01**

Table 3 Backchannel prediction performance using lexical features without the EL function. Each model is trained independently under various delay conditions. Results shown are macro-averaged values across positive and negative classes.

Delay condition in model training (sec)	Delay condition in evaluation (sec)	Precision	Recall	F1
0.5	0.5	56.07	59.77	**56.16**
1.0	1.0	**56.20**	**61.27**	55.42
1.5	1.5	55.45	58.86	55.31
2.0	2.0	55.62	58.75	55.77

5 Results and Discussion

The evaluation results are shown in Table 2. All the metrics shown in Table 2 (i.e., precision, recall, and F1) are macro averaged values.[3] The best result for each metric is indicated in bold. Under a 0.0-s delay condition, the combination of acoustic and lexical features with the EL function outperformed all the baselines, with an F1 score of 58.06%. The result for the TASN model was basically the same as the lexical-based model without EL (around 56%). We also evaluated the models for different lengths of the delay condition. If we assume that the acoustic extraction process does not produce any delay, the performance of the acoustic-based model should be the same under all delay conditions, with an F1 score of 49.26%. Our combination of acoustic and lexical features with the EL function achieved an F1 score of 58.06% under the 0.0-s and 57.18% under the 0.5-s delay conditions. Afterwards, its performance dropped to 52.01% under the 2.0-s delay condition, which is still a better result than the acoustic-based model. Moreover, our method is better than TASN [12] (47.46%) and the "Acoustic + POS" model [16] (48.08%) under the same delay condition. It means our method can predict backchannel with an improvement rate on the F1 score 8% better than these state-of-the-art algorithms. Based on these results, our prior detection model achieves better performance in the prediction of suitable backchannel timing than the baseline method in which the model is trained without the EL function, under all delay conditions.

Next, we compare the acoustic- and lexical-based models independently. According to the F1 score, the lexical features were the most informative features and provided a better performance than the acoustic features. We also evaluated the models using simulated test data in which only the available words were fed to the models (see a simulated example in Table 1). As shown in Table 2, whether or not the EL function is applied, F1 scores for the lexical-based model in the 0.5-s delay condition were almost the same (around 56%). On the other hand, we can see the effect of the

[3] Each metric (precision, recall, or F1) is calculated for both positive and negative classes. Each macro-averaged metric is calculated by averaging the corresponding metrics for the positive class and the negative class.

EL function when the models are evaluated in longer delay conditions (1.0–1.5 s). The F1 scores of the lexical-based model trained with the EL function in these delay conditions were still higher than 54%, while the scores of the models trained without the EL function dropped significantly. However, even when the EL function was applied, the F1 scores dropped to 50.02% under the 2.0-s delay condition. This is a reasonable result, the average utterance duration in the dataset is 1.93 s. Therefore, under the 2.0-s delay condition, the model was evaluated mostly with a [silb] word as the input feature. However, it still achieves better performance than the acoustic-based model. According to these results, our prior detection model could successfully predict the backchannel events before the final ASR outputs.

Compared to the other combined methods, our combination (acoustic and lexical) model with the EL function had a better performance even under the 2.0-s delay condition. A continuous neural network that combines all the features (e.g., the TASN architecture) works well under zero latency, but if the model is used with an available input in the delay condition, it will likely fail to predict the backchannel accurately. This is because, under the delay condition, acoustic and lexical features do not correspond to each other. Without the EL function, the model could not predict backchannel as well as it could under the 0.0-s delay condition.

Another idea to keep the performance high under a longer delay condition is by training the model with the available words. The results in Table 3 show that, even without the EL function, the model achieved an F1 score higher than 55% for all delay conditions. However, building a model for each delay condition is difficult because it depends on the spoken dialog or the ASR system. We therefore suggest combining the lexical model and acoustic model for a better prediction.

6 Conclusion

In this paper, we have proposed a prior prediction model with an EL function for backchannel detection that can mitigate the response delay problem caused by ASR. The model was evaluated under various delay lengths since the delay might differ depending on the conditions. Results showed that the proposed prior prediction model can successfully predict backchannel even when only the available words are input as features. In future work, we will implement our proposed model in a robot system and perform a subjective evaluation.

References

1. Aldeneh Z, Dimitriadis D, Provost EM (2018) Improving end-of-turn detection in spoken dialogues by detecting speaker intentions as a secondary task. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP), pp 6159–6163 (2018)

2. Calhoun S, Carletta J, Brenier JM, Mayo N, Jurafsky D, Steedman M, Beaver D (2010) The NXT-format switchboard sorpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. Lang Resources Eval 44(4):387–419
3. Chan FH, Chen YT, Xiang Y, Sun M (2016) Anticipating accidents in dashcam videos. In: Proceedings of the computer vision-asian conference on computer vision (ACCV). Springer, pp 136–153
4. Eyben F, Scherer KR, Schuller BW, Sundberg J, André E, Busso C, Devillers LY, Epps J, Laukka P, Narayanan SS et al (2015) The geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. IEEE Trans Affect Comput 7(2):190–202
5. Eyben F, Wullmer M, Schuller (2018) OpenSMILE – the Munich versatile and fast open-source audio feature extractor. In: Proceedings of the ACM international conference on multimedia (ACM Multimedia), pp 1459–1462
6. Godfrey J, Holliman E, McDaniel J (1992) Telephone speech corpus for research and development. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP), pp 517–520
7. Hara K, Inoue K, Takanashi K, Kawahara T (2018) Prediction of turn-taking using multitask learning with prediction of backchannels and fillers. In: Proceedings of the annual conference of the international speech communication association (INTERSPEECH), pp 991–995
8. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
9. Jain, A., Singh, A., Koppula, H.S., Soh, S., Saxena, A.: Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 3118–3125. IEEE (2016)
10. Kawahara T, Uesato M, Yoshino K, Takanashi K (2015) Toward adaptive generation of backchannels for attentive listening agents. In: Proceedings of the international workshop on spoken dialogue systems technology (IWSDS), pp 1–10
11. Kawahara T, Yamaguchi T, Inoue K, Takanashi K, Ward NG (2016) Prediction and generation of backchannel form for attentive listening systems. In: Proceedings of the annual conference of the international speech communication association (INTERSPEECH), pp 2890–2894
12. Masumura R, Asami T, Masataki H, Ishii R, Higashinaka R (2017) Online end-of-turn detection from speech based on stacked time-asynchronous sequential networks. In: Proceedings of the annual conference of the international speech communication association (INTERSPEECH), pp 1661–1665
13. Meshorer T, Heeman PA (2016) Using past speaker behavior to better predict turn transitions. In: Proceedings of the annual conference of the international speech communication association (INTERSPEECH), pp 2900–2904
14. Morency LP, de Kok I, Gratch J (2010) A probabilistic multimodal approach for predicting listener backchannels. Auton Agent Multi-Agent Syst 20(1):70–84
15. Ries K (1999) HMM and neural network based speech act detection. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP), pp 497–500
16. Roddy M, Skantze G, Harte N (2018) Investigating speech features for continuous turn-taking prediction using LSTMs. In: Proceedings of the annual conference of the international speech communication association (INTERSPEECH), pp 586–590
17. Ruede R, Müller M, Stüker S, Waibel A (2017) Enhancing backchannel prediction using word embeddings. In: Proceedings of the annual conference of the international speech communication association (INTERSPEECH), pp 879–883 (2017)
18. Ruede R, Müller M, Stüker S, Waibel A (2017) Yeah, right, uh-huh: a deep learning backchannel predictor. In: Proceedings of the international workshop on spoken dialogue systems technology (IWSDS), pp 247–258
19. Shiwa T, Kanda T, Imai M, Ishiguro H, Hagita N (2008) How quickly should communication robots respond? In: Proceedings of the ACM/IEEE international conference on human-robot interaction (HRI), pp. 153–160 (2008)
20. Skantze G (2017) Towards a general, continuous model of turn-taking in spoken dialogue using LSTM recurrent neural networks. In: Proceedings of the annual SIGdial meeting on discourse and dialogue (SIGDIAL), pp 220–230 (2017)

21. Truong KP, Poppe R, Heylen D (2010) A rule-based backchannel prediction model using pitch and pause information. In: Proceedings of the annual conference of the international speech communication association (INTERSPEECH), pp 3058–3061 (2010)
22. Ward N, Tsukahara W (2000) Prosodic features which cue back-channel responses in English and Japanese. J Pragmat 32(8):1177–1207