EasyChair Preprint

№ 15119

Predicting Utterance-Final Timing Considering Linguistic Features Using Wav2vec 2.0

Takanori Kanai, Yukoh Wakabayashi, Ryota Nishimura and Norihide Kitaoka

September 27, 2024

Predicting Utterance-final Timing Considering Linguistic Features Using Wav2vec 2.0

Takanori Kanai*, Yukoh Wakabayashi*, Ryota Nishimura†, Norihide Kitaoka*
*Toyohashi University of Technology
Toyohashi, Japan
Email: kanai.takanori.kq@tut.jp, wakayuko@cs.tut.ac.jp, kitaoka@tut.jp
†Tokushima University
Tokushima, Japan
Email: nishimura@is.tokushima-u.ac.jp

***Abstract*—Accurate turn-taking prediction is essential in spoken dialog systems, in order to determine whether the system or the user should make the next utterance. Previous research has significantly improved the accuracy of turn-taking prediction, allowing dialog systems to avoid unnatural pauses before responding. However, in human-to-human dialogs, responses do not always occur immediately after a speaker's utterance ends; sometimes there are deliberate pauses or responses made with overlap. Therefore, this study proposes a method to estimate in advance when the interlocutor's utterances will end, allowing the system to respond with more natural timing, including occasional overlaps. We utilized wav2vec 2.0, fine-tuned for automatic speech recognition, to estimate utterance end times by considering linguistic features, and compared these methods with prediction methods that use only acoustic features. The results of our comparison showed that considering linguistic features allows more accurate prediction of utterance-final timing. Additionally, we observed that when using the proposed method, the estimated time until the end of the utterance decreases as the utterance approaches its end.**

***Index Terms*—spoken dialog system, turn-taking, utterance-final timing prediction, wav2vec 2.0**

I. Introduction

Current spoken dialog systems are no longer simply task-oriented systems that answer user's questions, but there are also systems that perform non-task-oriented functions, such as chatting with users. As a result, rapid changes in dialog content have made it more difficult for these systems to determine turn-taking timing by identifying the end user utterances. Therefore, it has become necessary for dialog systems to make better turn-taking decisions in order to achieve smooth conversation.

A naive approach to turn-taking is for the system to assume that the conversation has been handed over only when the speaker stops talking and the duration of their silence exceeds a set threshold. The naturalness of conversations when using this approach depends on the selected threshold. For example, when using a small value, the system will interrupt the speaker's utterances more frequently, while larger values result in awkwardly long pauses. Therefore, to achieve natural conversations similar to those between people, a different approach for predicting turn-taking is necessary.

Recent research has focused on using machine learning to predict turn-taking, by determining whether the user has completed an utterance, or if the user intends to continue speaking. Many turn-taking prediction methods which use acoustic features are based on Inter-Pausal Units (IPUs), which are delimited by fixed lengths of silence. Annotations are made on these IPUs and on subsequent intervals of silence to predict turn-taking using acoustic features [1]. Jiudong et al. [2] have proposed using not only acoustic features, but also linguistic features, IPU length, and speaking rates.

Hara et al. [3] proposed using Transition Relevance Places (TRPs), intervals during which speaker changes would not seem unnatural, for turn-taking predictions apart from methods using IPUs. Other studies have proposed predicting whether a pause by the speaker in natural, non-fluent conversations is merely a pause or an indication that they have finished speaking [4]. More recently, methods leveraging self-supervised learning models (SSLs) such as wav2vec 2.0 [5] and HuBERT [6], as well as Large Language Models (LLMs) such as GPT2 [7], have also been proposed. SSLs are used in various tasks such as automatic speech recognition (ASR) and speaker classification, and have achieved accuracy rates comparable to or exceeding existing high-precision ASR models by performing fine-tuning (FT) with only a small amount of data. SSLs have been used for turn-taking prediction and have achieved higher accuracy than traditional acoustic feature-based methods, such as using Mel-Frequency Cepstrum Coefficients (MFCC) [8]. LLMs are considered to capture linguistic semantic relationships better. Research has been conducted using LLMs to predict turn-taking using only textual information [9]. When combined with acoustic models, these systems not only predict turn-taking but can also identify occurrences of backchannels, achieving higher accuracy than the use of textual information alone [10]. Researchers have also proposed using multimodal features to predict turn-taking, by combining acoustic, linguistic, and other feature [11]. All of these recently developed approaches can more accurately determine the end of utterances and turn-taking, allowing more natural system responses without unnatural pauses.

However, in human conversations, responses do not always occur immediately after the end of a conversation partner's utterance, because sometimes humans intentionally pause before responding, or their response may slightly overlap the

end of the previous speaker's utterance [12]. Achieving such complex, human-like interactions in dialogs with systems would be very challenging using existing methods, which predict turn-taking at the moment an utterance-final occurs. Therefore, we believe that in order to achieve more human-like interactions, it is necessary to know in advance when the other speaker's utterance will end. So in this study, rather than estimating whether the system can acquire the right to speak at the end of a user's utterance, as in conventional methods, we propose a method that estimates in advance when a speaker's utterance will end, enabling the system to achieve more natural timing, including overlaps. The most important key point of this study is the approach of how many seconds later the current speaker "will" finish speaking, which is essentially different from conventional methods of turn-taking estimation and methods like [11] that predict who the next speaker will be after an arbitrary period of time has elapsed. Furthermore, given the success of SSL models in turn-taking prediction tasks, we use wav2vec 2.0 for feature extraction and apply it to utterance end-time estimation. While some existing turn-taking prediction methods utilize linguistic information, most use pre-given texts, or transcripts obtained from separate ASR models. However, in actual systems, pre-given texts obviously do not exist, and transcripts obtained from ASR models may contain errors, which can significantly impact the accuracy of predictions. Therefore, we propose a method that leverages the hidden representations of wav2vec 2.0 with FT for ASR, utilizing linguistic features without the use of transcripts. This approach takes advantage of the ease with which representations obtained from a once-trained SSL model can be applied to different tasks.

II. Proposed Method

The objective of this study is to achieve natural interactions between users and spoken dialog systems by controlling the timing of the system's initiation of utterances. To achieve this, we propose a method for sequentially estimating the time at which the user's utterance, currently being input into the system, will end, in real time. Specifically, we construct a deep learning model capable of estimating the time remaining until the user's current utterance input ends, using acoustic information previously input into the system. An overview of our proposed method is shown in Fig. 1.

A. Wav2vec 2.0 for linguistic features utilization

Many previously proposed methods of turn-taking prediction have utilized linguistic information, suggesting that linguistic features such as contextual and semantic information are effective [2], [8], [11]. In this experiment, we use features obtained from wav2vec 2.0 with FT for ASR to consider linguistic features. First, we construct a wav2vec 2.0 for ASR using dialog speech. This model employs wav2vec 2.0, fine-tuned with dialog speech using end-to-end learning, as the encoder, and a fully-connected layer and Connectionist Temporal Classification (CTC) [14] as the decoder. Details of this method are provided in Section IV. This approach

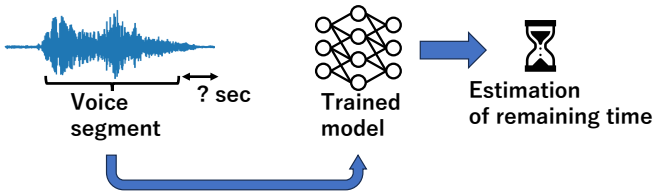

Fig. 1: Overview of the proposed method.

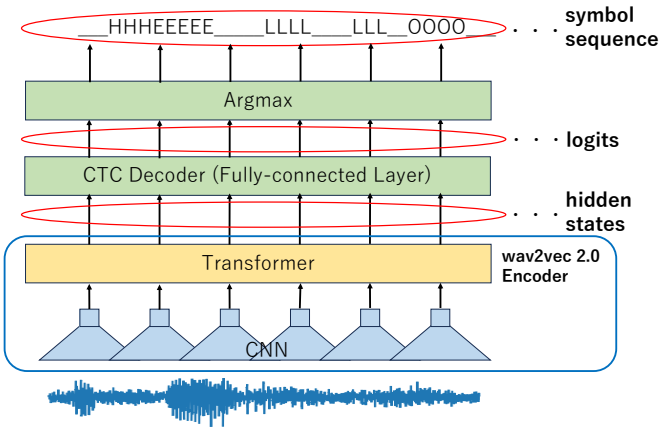

Fig. 2: Linguistic features obtained from wav2vec 2.0.

allows wav2vec 2.0 to be tuned into a model that can capture linguistic features, such as contextual information, so it can be utilized for predicting time when the speaker's utterance will end.

B. Linguistic features from wav2vec 2.0

In this experiment, we use wav2vec 2.0 with FT for ASR. To investigate if there are differences in prediction accuracy between features obtained from wav2vec 2.0, we use features from the final layer of the wav2vec 2.0 (hidden states), the probability state of each word token (logits), and the state in which a single token is determined with a mixture of blanks (symbol sequence). Additionally, to verify the effectiveness of considering linguistic features, we compare prediction accuracy using these features with that of using features obtained from the final layer of wav2vec 2.0 without FT for ASR (hidden states w/o FT). A graphical representation of the features actually used is shown in Fig. 2.

III. Dataset

The dataset used in this experiment is the Corpus of Everyday Japanese Conversation (CEJC) [13], which is composed of conversations which occur naturally in everyday life, such as conversations of families while dining at a restaurant, casual chats with acquaintances, etc. It includes 200 hours of audio, 577 conversations, and 461 recording sessions featuring the voices of 1,675 speakers. The corpus features audio and video recordings of natural conversations set in everyday situations, as well as transcripts with the start and end times of each utterance in each session. Verbal/non-verbal tags are also available.

IV. Training ASR model

It is necessary to fine-tune wav2vec 2.0 for ASR of the dialog system user's speech, in order to predict when their utterances will end considering the linguistic features. Here, we describe how wav2vec 2.0 was fine-tuned for ASR, and then present the results. The ASR model used wav2vec 2.0 as the encoder, a fully-connected layer as the decoder, and CTC as the loss function. The dataset used for fine-tuning the model was the CEJC, which will also be used in the subsequent utterance end-time prediction experiment. The verbal/non-verbal tags included in the CEJC transcripts were removed before training, and the experimental data was divided into training, validation, and test sets, which consisted of 138, 1, and 1 hours of data, respectively.

Performance of the fine-tuned ASR model was measured using the Character Error Rate (CER), and our experimental results revealed a CER of 27.6%. These results were inferior compared to ASR performance when using a normal speech corpus, which was likely because the CEJC is a casual conversation corpus with spontaneous speech, so it includes fillers, backchannels, and overlaps, which make ASR more challenging. But when compared with the recognition results for other ASR models when processing CEJC speech [15], the error rate of our model was inferior by only 4.1 points, therefore we determined that recognition performance of our proposed model for CEJC speech was generally satisfactory, so it was selected for use.

V. Experiment setup

A. Feature extraction

MFCC, Convolutional Neural Network (CNN), and wav2vec 2.0 were each used as feature extractors. MFCC was a 40-dimensional feature obtained under the following conditions: sampling rate $=$ 16 kHz, frame size $=$ 32 ms, frame shift size $=$ 10 ms, and Mel-filter bank $=$ 80-dimensionals. The CNN used a model in which speech is convolved four times in the following order; 1D convolution, 1D max pooling, and layer norm. A kernel size of 2 and stride of 1 were used. It should be noted here that, unlike the other feature extractors, the CNN was trained end-to-end simultaneously with the classifier, which will be described at the end of the following subsection. As mentioned in Section II, the wav2vec 2.0 was fine-tuned for ASR. The four types of features used from the wav2vec 2.0 are: features obtained from the final layer of wav2vec 2.0 (hidden states), the probability state of each word token (logits), the state in which a single token is determined with a mixture of blanks (symbol sequence), and features obtained from the final layer of wav2vec 2.0 without FT for ASR (hidden states w/o FT).

B. Experiment Detail

In this experiment, the task is to estimate the remaining time until the end of the current utterance from a fixed-length voice segment. We remade a new dataset from an existing one for this task. Figure 3 illustrates a concept of a sample in the dataset. The CEJC dataset was used as the existing

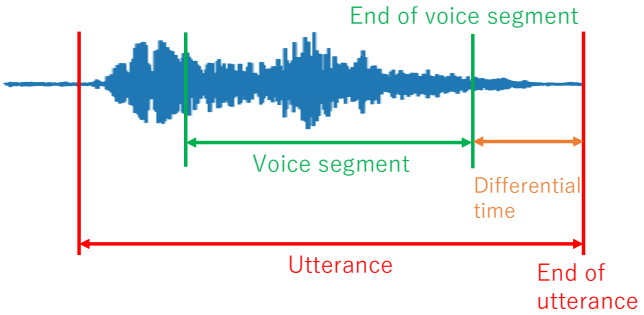

Fig. 3: Definition of each time point and voice segment in an utterance.

dataset, similar to the FT of ASR. The CEJC dataset includes tagging the start and end times of each utterance, making it easy to obtain a single utterance. We obtained a single utterance using these start and end times, excluding utterances with only backchannels or fillers. Concretely, we regarded utterances longer than 2 seconds as available ones. The length corresponds to the length of "Utterance" in Fig. 3. Then, we extracted a fixed-length voice segment obtained before the end of the utterance from one utterance, which is shown as "Voice segment" in Fig. 3. In this experiment, the length of the voice segment was set to 3 seconds. Subsequently, the differential time between the end of the voice segment and the end of the utterance was calculated, and this is shown as "Differential time" in Fig. 3. We repeated this process by randomly changing the differential time within 1 second and obtained various combinations of voice segments and the differential time. Through such repeated manipulations, we built a dataset that includes voice segments and the corresponding remaining time until the end of an utterance. We built a model to estimate the differential time from a voice segment by training it with this dataset.

We used five-class classification in our experiment (which is easier to analyze than regression) to confirm trends in the estimation of time until the end of utterance. The classification model uses three types of feature extractors as the encoder, five-class classifiers with two layers of long short-term memory (LSTM) [16], [17], and three layers of fully-connected layers as the decoder. The differential time classes to be estimated were as follows: 0.0–0.2 seconds, 0.2–0.4 seconds, 0.4–0.6 seconds, 0.6–0.8 seconds, and 0.8–1.0 seconds. Finally, Adam [18] was used as the optimization algorithm, and Cross Entropy Loss was used as the loss function.

C. Evaluation Method

To evaluate the results of this experiment, we utilized overall estimation accuracy, as well as the macro-averages of precision, recall, and F1 score for each class. In addition, the classification results for the input feature with the highest accuracy were summarized in a confusion matrix, and classification accuracy for each differential time class was analyzed.

TABLE I: Experimental results of utterance end-time prediction.

Feature		Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
MFCC		29.1	28.0	29.1	28.0
CNN		31.2	29.8	31.2	29.8
wav2vec 2.0 w/ FT	hidden states	**37.0**	**35.8**	**37.0**	**36.1**
	logits	36.9	35.6	36.9	36.0
	symbol sequence	27.4	25.8	27.4	23.4
wav2vec 2.0 w/o FT	hidden states	35.1	33.8	35.1	34.0

TABLE II: Confusion matrix of the classification results using "hidden states". Values in the table show the number of samples classified in each class. Correctly estimated samples are shown in bold type.

		Estimated class				
		0.0–0.2 sec	0.2–0.4 sec	0.4–0.6 sec	0.6–0.8 sec	0.8–1.0 sec
True class	0.0–0.2 sec	**10,705**	3,467	1,236	907	1,329
	0.2–0.4 sec	4,466	**5,467**	3,129	2,355	2,227
	0.4–0.6 sec	2,429	3,628	**3,834**	3,814	3,939
	0.6–0.8 sec	1,816	2,207	3,033	**4,480**	6,109
	0.8–1.0 sec	1,623	1,562	2,329	3,941	**8,190**

VI. EXPERIMENT RESULTS

A. Comparative Experiment

The results of the experiments conducted with different feature extractors are shown in Table I. MFCC, CNN, and "hidden states w/o FT" are results when using only acoustic features, while "hidden states", "logits", and "symbol sequence" are results when considering linguistic features, using wav2vec 2.0 with FT for ASR. It was found that using "hidden states" as features yielded the It was found that using the linguistic feature "hidden states" yielded the best time-to-end-of-utterance estimation results. Results for both "hidden states" and "logits", which are features from wav2vec 2.0 with FT for ASR, outperformed the acoustic-only features of MFCC, CNN, and "hidden states w/o FT". This suggests that consider of linguistic features allows more accurate estimation of utterance end-times compared to using only acoustic features. Furthermore, among the features from wav2vec 2.0 with FT for ASR, "hidden states" and "logits" achieved the highest prediction accuracy, but the "symbol sequence" achieved the lowest accuracy in this experiment. This indicates that using hidden representations obtained before the final ASR output is more effective than using the uniquely determined ASR results.

B. Confusion matrix of classification results

A confusion matrix of the classification results when using "hidden states", which achieved the highest classification accuracy in this experiment, is shown in Table II. The diagonal classification totals shown in bold type represent correct time-to-end-of-utterance classifications, and the farther the other totals are from this diagonal, the greater the error. We can see in Table II that the classification totals closer to the diagonal are larger, indicating that approximate estimation is possible. When allowing for one-class error, the accuracy rate becomes 73%. This suggests that even if the predicted class was incorrect, it was highly likely that the prediction fell into a class close to the correct value. Furthermore, when comparing the prediction results for the 0.0–0.2 seconds class with those for the 0.8–1.0 seconds class, note that the classification totals in the classes that are one class off of the correct 0.0–0.2 seconds results are less than half of the correct totals, whereas in the 0.8–1.0 seconds class, they are three-quarters of the correct totals. This indicates that as the distance from the end of the utterance increases, predicting its end time becomes more difficult.

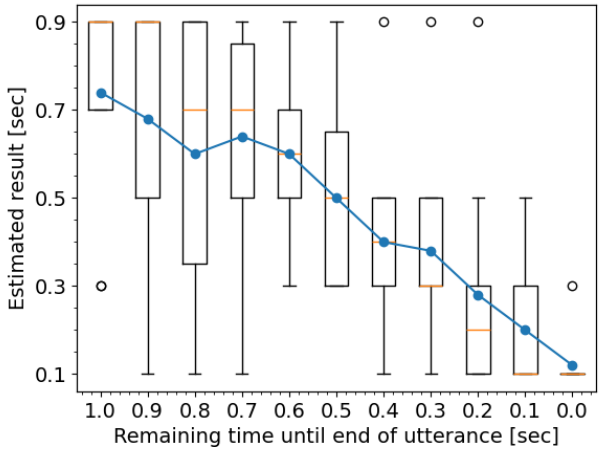

Fig. 4: Estimation results every remaining time to the end of an utterance, where results of ten utterances are averaged. Boxes indicate variance.

When we listened to the voice segments that were classified into the classes farthest from the correct 0.0–0.2 seconds class, the voice segments had elongated endings or semantic content that made it reasonable to predict the utterance would continue. Additionally, the voice segments farthest from the correct 0.8–1.0 seconds class would not be unnatural if they ended here. As an utterance gets closer to ending and the next utterance begins to be heard, the likelihood of significant errors decreases. This demonstrates the difficulty of predicting the end of an utterance in spontaneous utterance.

C. Results of utterance end time prediction using this model

The average prediction results when running this model at 0.1 seconds intervals within 1 second of the end of the utterance, using 10 utterances from the test data, are shown in Fig. 4. In this case, the prediction results take the median value of each class, e.g., 0.1 seconds for the 0.0–0.2 seconds class, 0.3 seconds for the 0.2–0.4 seconds class, etc. The line graph shows that the estimated time-to-end decreases as the utterance approaches its end, and that as remaining time decreases, the variance also decreases, suggesting that estimation tends to stabilize. Therefore, it can be concluded that utterance end-time estimation using this model follows a valid prediction trend.

VII. Conclusion

In this study, we proposed a method for estimating the ending time of an input utterance in order to achieve more human-like interactions in spoken dialog systems. To estimate the ending times of utterances, we constructed a model that outputs the time until the end of the utterance into a five-class classification, based on a fixed duration voice segment. For the input to the model, we tested three types of feature extractors: MFCC, CNN, and wav2vec 2.0. Our experimental results showed that a wav2vec 2.0 fine tuned for ASR which considered the linguistic features achieved more accurate end-time estimation than methods using only acoustic features. In the future, we plan to conduct further experiments, varying the length of the input voice segments and combining multiple features. Additionally, as the predictions in this study were limited to a range of 0 to 1 seconds, we plan to extend the prediction range both forward and backward. Ultimately, we aim to integrate the method into an actual system and evaluate how naturally it can respond to real conversation.

References

[1] Seyedeh Zahra Razavi, Benjamin Kane, and Lenhart K Schubert, "Investigating Linguistic and Semantic Features for Turn-Taking Prediction in Open-Domain Human-Computer Conversation," in *Interspeech*, pp. 4140–4144, 2019.

[2] Jiudong Yang, Peiying Wang, Yi Zhu, Mingchao Feng, Meng Chen, and Xiaodong He, "Gated Multimodal Fusion with Contrastive Learning for Turn-Taking Prediction in Human-Robot Dialogue," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7747–7751, 2022.

[3] Kohei Hara, Koji Inoue, Katsuya Takanashi, and Tatsuya Kawahara, "Turn-taking Prediction Based on Detection of Transition Relevance Place," in *Interspeech*, pp. 4170–4174, 2019.

[4] Shuo-yiin Chang, Bo Li, Tara N. Sainath, Chao Zhang, Trevor Strohman, Qiao Liang, and Yanzhang He, "Turn-Taking Prediction for Natural Conversational Speech," in *Interspeech*, 2022.

[5] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451—3460, 2021.

[7] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, "Language models are unsupervised multitask learners," OpenAI Blog, vol. 1, no. 8, 2019.

[8] Edmilson Morais, Matheus Damasceno, Hagai Aronowitz, Aharon Satt, and Ron Hoory, "Modeling Turn-Taking in Human-To-Human Spoken Dialogue Datasets Using Self-Supervised Features," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023.

[9] Erik Ekstedt, and Gabriel Skantze, "TurnGPT: a Transformer-based Language Model for Predicting Turn-taking in Spoken Dialog," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2981—2990, 2020.

[10] Jinhan Wang, Long Chen, Aparna Khare, Anirudh Raju, Pranav Dheram, Di He, Minhua Wu, Andreas Stolcke, and Venkatesh Ravichandran, "Turn-Taking and backchannel prediction with acoustic and large language model fusion," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12121–12125, 2024.

[11] Mehdi Fatan, Emanuele Mincato, Dimitra Pintzou, and Mariella Dimiccoli, "3M-Transformer: A Multi-Stage Multi-Stream Multimodal Transformer for Embodied Turn-Taking Prediction," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8050–8054, 2024.

[12] Tanya Stivers, N. J. Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter de Ruiter, Kyung-Eun Yoon, and Stephen C Levinson, " Universals and cultural variation in turn-taking in conversation, " in *Proceedings of the National Academy of Sciences of the United States of America 106*, 10587—92, 2009.

[13] Hanae Koiso, Haruka Amatani, Yasuharu Den, Yuriko Iseki, Yuichi Ishimoto, Wakako Kashino, Yoshiko Kawabata, Ken'ya Nishikawa, Yayoi Tanaka, Yuka Watanabe, and Yasuyuki Usuda, "Design and Evaluation of *the Corpus of Everyday Japanese Conversation, Proceedings of LREC2022*," pp. 5587–5594, 2022.6.

[14] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *The 23rd International Conference on Machine Learning (ICML ' 06)*, 369—376, 2006.

[15] Nagito Shione, Yukoh Wakabayashi, and Norihide Kitaoka, "Construction of Automatic Speech Recognition Model that Recognizes Linguistic Information and Verbal/Non-verbal Phenomena," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 2306–2311, 2023.

[16] Sepp Hochreiter, and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] Haşim Sak, Andrew Senior, Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[18] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.