

BUILDING TAILORED SPEECH RECOGNIZERS FOR JAPANESE SPEAKING ASSESSMENT

Yotaro Kubo, Richard Sproat, Chihiro Taguchi, Llion Jones

Sakana AI, Tokyo, Japan.

{yotarokubo, rws, chihirotaguchi, llion}@sakana.ai

ABSTRACT

This paper presents methods for building speech recognizers tailored for Japanese speaking assessment tasks. Specifically, we build a speech recognizer that outputs phonemic labels with accent markers. Although Japanese is resource-rich, there is only a small amount of data for training models to produce accurate phonemic transcriptions that include accent marks. We propose two methods to mitigate data sparsity. First, a multitask training scheme introduces auxiliary loss functions to estimate orthographic text labels and pitch patterns of the input signal, so that utterances with only orthographic annotations can be leveraged in training. The second fuses two estimators, one over phonetic alphabet strings, and the other over text token sequences. To combine these estimates we develop an algorithm based on the finite-state transducer framework. Our results indicate that the use of multitask learning and fusion is effective for building an accurate phonemic recognizer. We show that this approach is advantageous compared to the use of generic multilingual recognizers. The relative advantages of the proposed methods were also compared. Our proposed methods reduced the average of mora-label error rates from 12.3% to 7.1% over the CSJ core evaluation sets.

Index Terms— Automatic speech recognition, phonemic transcription, pitch-accented language, speaking assessment.

1. INTRODUCTION

Automatic speech recognition (ASR) technology has been evolving to enable the mapping of speech signals to canonical textual representations under a variety of conditions. In typical ASR systems, speaker errors, such as incorrect accent positions, mispronunciations, fillers, and hesitations, are intentionally discarded. Although many applications prefer such canonical output rather than a phonetically accurate transcription, in certain cases such as language education, this normalization effect can be viewed as a limitation because it hinders the accurate evaluation of users’ speaking proficiency.

There have been several attempts to build universal phonetic transcribers, with a particular focus on extending ASR to low-resource languages [1, 2, 3]. While multilingual models are potentially applicable to speaking skill assessment, assessing language-specific phenomena, such as pitch accent, is also essential for educational purposes. In this paper, we focus on phonemic transcription of Japanese along with pitch accent markers, for the purposes of building a speech recognizer tailored for Japanese speaking assessment. Considering the complexity and irregularity of Japanese pitch accent rules [4, 5], this presents a unique challenge that cannot be solved only by multilingual phonetic recognizers.

The training of phonetically accurate recognizers often suffers from the limited availability of accurate phonetic or phonemic transcriptions. Although Japanese is a relatively resource-rich language,

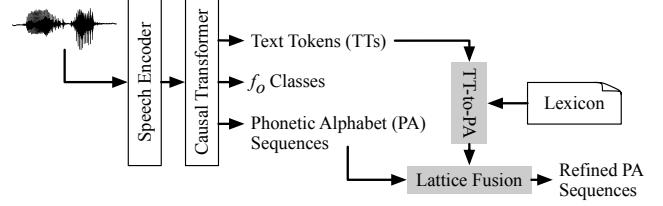


Fig. 1. Proposed architecture of our PA recognizer. Boxes with white backgrounds are neural components, and those with dark backgrounds are algorithmic processing.

the largest multi-speaker dataset with hand-annotated pitch accent is, to the best of our knowledge, the “core” subset of the Corpus of Spontaneous Japanese (CSJ) [6], containing only 45 hours of speech.

Prior work by [7] developed a phonemic recognizer conditioned on the outputs of external speech recognizers. Although their method did not include a built-in ASR component, there are several similarities to our proposal. Their system performs both implicit conditioning, employing text tokens as an auxiliary input, and explicit conditioning using text information as constraints on decoding. Unlike their implicit conditioning method, our approach does not require reliable text labels because it solves text and phonetic alphabet (PA) recognition simultaneously in a multitask learning framework. Further, as discussed in the following section, one can view our decoding method as an extension of their explicit conditioning method that can robustly handle weak text constraints. For a conventional ASR task, the iterative refinement method is proposed in [8] in order to estimate PA and text token sequences jointly. However, this method does not estimate accent locations, and does not utilize a dictionary to enhance accuracy in small data scenarios.

The technical contributions in this paper are twofold: First, a multitask training method that is effective for pitch-accent recognition; second, a decoding method that decodes an optimal PA sequence robustly by considering both text tokens and PA estimation results. These two novel methods are implemented in a streamable ASR model, and comparative experiments were conducted to confirm the effectiveness of the proposed methods. Fig. 1 illustrates our proposed method. The next three sections describe the components in the figure.

2. SYSTEM DESIGN

2.1. Connectionist Temporal Classification

Conventionally, language models are integrated into ASR systems in order to improve the output quality leveraging the frequencies of phrases and words. However, since one of our goals is to detect errors in speech that manifest as infrequent fluctuations in token se-

quences, such corrections by language models may conflict with this objective.

Connectionist Temporal Classification (CTC) can be a solution to this problem. Unlike other ASR architectures, such as RNN-T [9] and LAS [10], a CTC model does not include a module for capturing mutual dependency within the output sequences. Therefore, it is expected that CTC models are incapable of correcting erroneous utterances into canonical ones. While it has been reported that even CTC can implicitly learn an internal language model [11, 12], CTC is still considered the best choice if correction needs to be minimized.

2.2. Corpus of Spontaneous Japanese (CSJ)

For our purposes, a training set annotated with PAs and accent markers is required. The transcription in the dataset must be faithful to the original speech, including errors. CSJ includes manual annotations for such speech variation. By training on this dataset, one can obtain a recognizer that can detect such errors represented in PAs. A subset of CSJ (called “core”) provides mora-wise phonemic annotations in katakana and the perceived location of accent;¹ however, this subset is only a small fraction of the entire dataset, consisting of only 23,683 training utterances, or less than 6% of the entire CSJ training dataset. Although Japanese is considered to be a resource-rich language, given the sparsity of accent-annotated datasets, it is still not straightforward to build an accurate recognizer out of the available data. The next sections describe practical remedies for this limitation.

3. MULTITASK LEARNING

The sparsity of phonetic labels with accent information is the central challenge in developing speech recognizers tailored for Japanese speaking assessment. The first remedy we employ is multitask learning. Orthographic transcription is cheaper to obtain compared to phonemic annotation, especially if the phonemic annotation also requires accent annotation. Since text transcriptions and phonemic annotations are expected to be mutually dependent, solving two tasks simultaneously can help learning a common representation of speech.

On the other hand, estimated pitch information is also cheap to obtain via a high-quality fundamental frequency (f_o) extraction algorithm. Since Japanese has lexical pitch accent, pitch information is also expected to be highly correlated with the target labels. Thus, in this paper, we train a transformer with three estimation tasks: one for PAs, one for text tokens (TTs), and one for estimated f_o classes.

3.1. Encoder Description

To maximize downstream utility, the proposed model is designed to be a streamable model. In order to ensure streamability, we adopt a streamable speech encoder and a causal transformer. For the speech encoder, we adopt the Mimi [14] model pretrained with 7 million hours of multilingual (mostly English) datasets [15]. Since our model does not require discrete inputs, the quantization and down-sampling modules are removed from Mimi.

¹Japanese is a *mora-timed* language, a mora being a prosodic unit corresponding to a short vowel (V) or a single post-vocalic consonant (C) (Syllable-initial consonants do not count for moraic purposes). Syllables of the form (C)V are monomoraic, and those of the form (C)VV or (C)VC are bimoraic. Each mora ideally occupies about the same amount of time, though see [13] for a detailed review of the situation for Japanese.

Our transformer model is based on Llama-2 [16]. Although Llama-2 was originally proposed for auto-regressive decoder-only modeling, our method adopts this architecture for making an encoder-only CTC model. Model parameters are randomly initialized. The embedding dimensionality is set to 512, the number of layers to 24, and the number of attention heads to 8. To overcome overfitting, dropout is introduced in several layers. A dropout with probability of 0.2 is applied to the input embeddings from the Mimi encoder and the attention probabilities in each self-attention layer.

3.2. Phonetic Alphabet and Text Token Tasks

Our PA tokenizer assumes a katakana transcription of the utterance. If an accent is placed on a mora, this is denoted by appending an apostrophe to the mora label (e.g. キュ /kʲu/ with accent is denoted as キュ'). The long-vowel marker (ー) is replaced by a copy of the preceding vowel. For example, the キュー /kʲu:/ is replaced by キュウ /kʲu u/. Similarly, キュ'ー is replaced by キュ'ウ. We collect all tokens from the CSJ core training set, resulting in 243 unique output tokens.

Text tokens (TTs) are character-based. We collected all Unicode points from the CSJ training dataset (core and non-core) after NFKC normalization, resulting in 2,309 distinct tokens.

We use two CTC estimators for PAs and TTs, each containing a fully-connected layer for computing logits of the CTC label distribution—i.e. the TT or PA token plus a “blank” token indicating absence of the corresponding output label.

3.3. f_o Classifier Task

The objective of the f_o classifier task is to ensure that pitch information is propagated through the transformer, allowing the model to estimate f_o contours. For each utterance, the Harvest algorithm[17] is applied to estimate f_o each 10 ms. Since pitch accent is based not on the pitch itself, but rather its trajectory, the task is designed to estimate predefined classes that indicate if f_o is going up or down.

For defining classes over f_o patterns, three analysis windows are introduced for each input frame. Let t_n be the timestamp of the n -th input frame for the transformer. The time segments for left, central, and right analysis windows are defined as $L_n = [t_n - 1.5w, t_n - 0.5w)$, $C_n = [t_n - 0.5w, t_n + 0.5w)$, and $R_n = [t_n + 0.5w, t_n + 1.5w)$, respectively, where w is the window length set to 40 ms. For each analysis window, we compute voicedness $V(S) \in \{\text{voiced}, \text{unvoiced}\}$, where S is either L_n , C_n or R_n . If there is a valid f_o estimation in the time segment S , $V(S) = \text{voiced}$, and $V(S) = \text{unvoiced}$ otherwise. The average of log- f_o $P(S)$ is computed for the voiced time segment S , i.e. $V(S) = \text{voiced}$.

The number of possible combinations of $V(L_n)$ and $V(R_n)$ is 4. For the case where $V(L_n) = V(R_n) = \text{voiced}$, we can further split the class depending on whether log- f_o is going up or not, i.e. $P(L_n) < P(R_n)$. Therefore, we can classify the left L_n and right R_n windows into 5 classes, which can be further split by the voicedness of the central window $V(C_n)$. In the end, we classify a frame into 10 classes based on the estimated f_o trajectory.

A fully connected layer is introduced to estimate the logits over the f_o classes. The training loss is enhanced by adding a term representing the cross-entropy loss of this class estimator.

4. LATTICE FUSION

To increase the accuracy of the PA recognizer in the face of data sparsity, our method consolidates estimation results of TT and PA

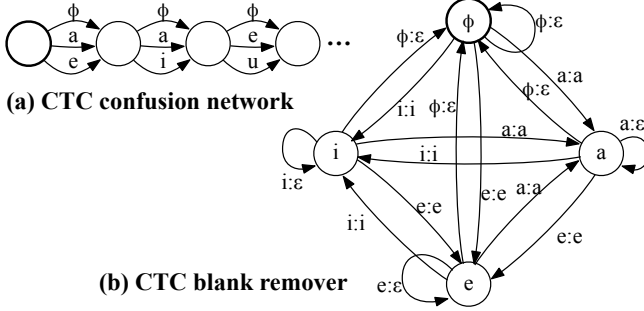


Fig. 2. Finite-state representation of (a) CTC outputs and (b) blank removal. For brevity, only phonemes /a, i, e/ and blank ϕ are shown.

recognition using finite-state transducers (FST). Whereas finite-state acceptors (FSA) represent probabilistic distributions over sequences, FSTs represent probabilistic mappings between sequences. We use FSAs—conventionally called “lattices”—to represent distributions over PA and TT sequences. The TT lattice is converted to a second PA lattice using an FST constructed from a pronunciation dictionary. Finally, the most plausible PA sequence from the union of the two PA lattices is extracted. We term this procedure “lattice fusion”.

4.1. Lattice Construction from CTC outputs

Let $Y \in \mathbb{R}^{T \times K}$ be an array of log probabilities of each class in CTC output layer, including the special blank symbol ϕ , where $y_{t,k}$ denotes the log-probability of label k emitted at time t . For each utterance, a confusion network over the CTC labels can be constructed as in Fig. 2 (a). In confusion networks, each non-final state represents the timestamp t and outgoing arcs from state t are defined for each possible k . The arc labeled as k from state t is designed to have a weight representing the negative log probability as $w = -y_{t,k}$. Let \mathcal{S}^P be the confusion network constructed by this procedure.

The CTC output labels must be post-processed by removing the same-label repetition and the blank ϕ symbols. This post-processing can be represented as a transducer as in Fig. 2 (b). By applying the composition operator to the confusion network \mathcal{S}^P and the CTC postprocessing FST \mathcal{B}^P , and projecting the FST to the output labels, the lattice \mathcal{L}^P that represents a probability distribution over all possible PA sequences can be computed. For computational efficiency, we further applied a lattice pruning method and further optimization for maximizing the efficiency of the subsequent procedures, as follows: $\mathcal{L}^P = \text{Opt}(\pi_O(\mathcal{S}^P \circ \mathcal{B}^P))$. Here, \circ denotes the composition operator on FSTs, π_O is a projection function that obtains an FSA from the FST by removing its input labels, and Opt is an optimization procedure that prunes, removes ϵ , determinizes, and minimizes [18] the input, in this order, over the log semiring. Following the procedure, a compact representation of PA distribution can be obtained from the output log-probabilities of the PA-CTC task.

We can similarly obtain TT sequence distributions from log-probabilities in the TT-CTC task. Let \mathcal{S}^T be the confusion network constructed in the same way from the TT log-probabilities. Following the above procedure, a lattice for TTs \mathcal{L}^T can be obtained.

4.2. Conversion from TT lattice to PA lattice

Our TT-to-PA FST is based on UniDic [19]. Of 876,803 UniDic entries, 873,647 were selected by discarding entries with null pronunciations or unknown PAs. Lexicon FST construction followed [18]. Using the TT-to-PA converter represented as an FST, a lattice repre-

senting distribution over PA sequences can also be obtained by converting the TT lattice to a PA lattice. This conversion can be done via FST composition. Let \mathcal{D} be an TT-to-PA FST, a PA lattice induced by the TT lattice can be computed as: $\mathcal{L}'^{T2P} = \text{Opt}(\pi_O(\mathcal{L}^T \circ \mathcal{D}))$. Furthermore, since the dictionary \mathcal{D} does not provide relative probabilities when a word has multiple pronunciation, it is necessary to accumulate PA weights to \mathcal{L}'^{T2P} . The PA probability can be taken from \mathcal{L}^P . The pronunciation-weighted lattice obtained in this way can be expressed as: $\mathcal{L}^{T2P} = \text{Norm}(\mathcal{L}^P \circ \mathcal{L}'^{T2P})$. Here, Norm is the normalization operator that normalizes the weights of outgoing arcs for each state to satisfy the sum-to-one constraint.

The explicit conditioning method proposed in [7] is equivalent to selecting the most plausible path from \mathcal{L}^{T2P} obtained as above. Unlike the explicit conditioning method, our method aims at robust decoding by fusing the estimation results from \mathcal{L}^{T2P} and \mathcal{L}^P . Fusing the two estimators can simply be performed by the union operator followed by optimization. The union lattice $\hat{\mathcal{L}}$ can be computed as: $\hat{\mathcal{L}} = \text{Opt}(\mathcal{L}^P \uplus \mathcal{L}^{T2P})$, where \uplus denotes the FST union operator. $\hat{\mathcal{L}}$ corresponds to the average of the two probabilistic distributions represented by \mathcal{L}^P and \mathcal{L}^{T2P} . The final recognition results can be obtained by computing the shortest path over the union lattice.

5. EXPERIMENTS

5.1. Experimental Setup

For analyzing the performances of the proposed systems, we compared three variants of the method: the full proposed method (MT+LF), a variant without lattice fusion (MT), and a variant with neither lattice fusion nor multitask learning (PA-only). Further, instead of lattice fusion, we implemented explicit conditioning [7] using our recognizers TT and PA output results (MT+Cond.). For LF systems, the MT+LF system computed the TT lattices using the TT outputs of the MT model. We additionally compared LF method with other sources of TT lattices. Whisper, and our “TT-only” variant were used as external TT lattice sources. For LF with Whisper, the lattice was constructed to have only a single path corresponding to the recognition result from Whisper. As side results, we also compare character error rates (CERs) of the MT model, Whisper, and our model without PA and f_o outputs (TT-only).

For training, the CSJ dataset is augmented with speed perturbation [20] followed by a time-domain variant of SpecAugment [21]. For speed perturbation, 20% of samples were modified to have 90% rate, and another 20% of samples were modified to have 110% rate. The time-domain SpecAugment is developed to apply masking as in the original SpecAugment but in the raw-waveform domain. This variant implements time-masking on the raw waveform, and the spectral masking on the (full utterance) discrete Fourier transform domain. The hyper-parameters for SpecAugment were set as follows: The number of time-domain masks was 10, the number of frequency-domain masks was 2, the maximum length of time-domain masks was $0.05T$ where T is the length of the input utterance, and the maximum length of frequency domain mask was $0.3F$, where F is the bandwidth of the input signal in mel. The task weights were preliminarily set to 0.3, 0.6, and 0.1 for the PA, TT, and f_o tasks respectively. We did not tune the task weights.

As baseline systems, we adapted Whisper [22] (whisper-1 model from OpenAI) and Multipa [3]. Since neither system is able to output our target PA directly, we made adapters that produce multiple PA outputs corresponding to the Whisper output text, or Multipa output IPAs. The minimum error rates among all possible out-

Table 1. Mora-label error rates [%] of PA recognition. ([†] = ASR output has multiple possible katakana representation, and minimum error rates are shown here. See main text for details.)

	eval1		eval2		eval3		JSUT	
accent err.	✓		✓		✓		✓	
Whisper[†]	20.6	24.3	15.1	19.7	13.5	16.5	2.8	6.8
Multipa[†]	17.2	–	19.8	–	18.3	–	16.2	–
PA-only	7.2	11.5	7.6	11.9	7.9	13.5	6.6	13.2
MT	4.4	7.3	4.9	7.8	5.0	9.1	5.8	9.9
MT+Cond.	4.3	7.3	5.2	8.2	5.0	9.3	6.1	10.3
MT+LF	4.1	7.0	4.8	7.7	4.9	9.1	6.1	10.3
MT+LF with external ASR models								
with Whisper	4.3	7.2	4.8	7.6	4.6	8.8	2.1	6.4
with TT-only	3.1	6.0	4.1	7.1	4.1	8.3	4.3	8.6

Table 2. Character error rates [%] of text transcription.

	eval1	eval2	eval3	JSUT
Whisper	18.9	17.2	15.1	8.3
TT-only	4.6	5.4	5.6	17.8
MT	6.3	6.8	7.8	22.6

puts from the adapter were used as a final metric for the systems with such adapters. For the Whisper adapter, since our adapter is based on UniDic, the adapter can also output PAs with accent markers based on the accent annotation given for each word in UniDic.

For evaluation, two datasets with accent annotations were employed. Since our model training is dependent on CSJ, core subsets of CSJ evaluation sets were used as our primary evaluation sets (CSJ eval1/ eval2/ eval3). In addition to CSJ, “basic5000” from the JSUT corpora was used [23]. This dataset (JSUT) contains read utterances from a single speaker. CSJ’s spontaneous and multi-speaker properties align more to our objective; however, JSUT is also important for probing the behavior of our methods in out-of-domain settings.

5.2. Discussion

Tables 1 and 2 show the mora-label error rates (MLERs) and CERs of the systems, respectively. MLERs were computed both with and without accent errors. We observed that the generic multilingual speech recognizer (Whisper) was not suitable for our phonemic transcription task. Even though our final metrics for Whisper were optimistic, computed by choosing the optimal pronunciations based on the reference labels, the results on the CSJ evaluation sets were not competitive with our tailored model. As discussed previously, this is due to the normalization effect from the language model. Table 2 shows that Whisper was also not competitive in CER. This was because CSJ utterances contain a lot of speaker errors, which Whisper tended to ignore. On the other hand, for JSUT, since this dataset contains read speech without speaker errors, Whisper worked almost perfectly. In that case, if we could choose the pronunciation of each

Table 3. Mora-label error rates [%] of each task combination. “noncore” shows whether the noncore subset was used in the training.

	noncore	eval1	eval2	eval3	JSUT
PA-only	–	11.5	11.9	13.5	13.2
PA + TT	✓	7.7	8.1	9.5	10.1
PA + f_o	–	11.8	11.9	14.0	13.7
PA + f_o	✓	11.8	12.4	13.9	12.5
PA + TT + f_o (MT)	✓	7.3	7.8	9.1	9.9

word optimally, the MLER can be very low (6.8%).

The Multipa results showed the difficulty of mapping phonetic transcriptions to language-specific phonemic transcriptions. Spontaneous speech exhibits wide variation in its phonetic realization, making it difficult to recover the phonemic labels as accurately as native speakers do. This shows that the generic multilingual phonetic transcriber is also not relevant for obtaining phonemic transcriptions.

Comparing PA-only and MT, it was shown that training with the noncore subset was important. Our method could successfully leverage the noncore subset which lacks accent annotations. The improvements made by MT were shown to be transferrable to the out-of-domain JSUT task where MLERs reduced from 13.2% to 9.9%.

From the results of the decoding methods (“MT+Cond” and “MT+LF”), it was shown that the lattice fusion was effective. The outputs of “MT+Cond” suggested that explicit conditioning is fragile if the recognizer cannot recognize textual representation of errors. We observed that neither decoding method was effective in the JSUT task, due to the inaccuracy of TT outputs used to compute TT lattices (see Table 2). This issue was mitigated by introducing an external speech recognizer as a source of TT lattices. As shown in Table 1, “MT+LF with Whisper” achieved the best result for the JSUT task. Similarly, since our “TT-only” model was better than Whisper for CSJ tasks, “MT+LF with TT-only” showed the best results for the CSJ tasks. Thus, we confirmed that “MT+LF” approach could further be enhanced if we had an accurate TT recognizer.

To further analyze how each task contributes in multitask training, comparative experiments over each combination of auxiliary task were conducted. As shown in Table 3, most of the advantage of multitask training is attributable to the TT task. The f_o task was not effective if it was only an auxiliary task (PA+ f_o rows). However, f_o yields a considerable gain when it was combined with TT, suggesting that f_o can be used to regularize the TT task. Since TT labels can only have limited information on accents, adding f_o in conjunction with TT was important to obtain a reliable recognizer that can also leverage prosodic information.

6. CONCLUSIONS

In this paper, we proposed and evaluated methods for building tailored speech recognizers for Japanese speaking assessment. The proposed recognizer is equipped with phonetic alphabet and text token decoders, and the phonetic alphabet results are decoded using a lattice fusion technique that integrates the recognition results from text token decoders with using a pronunciation dictionary. The recognizer is trained on CSJ, which contains phonetic annotations of mispronunciations, hesitations and accents. Our model training uses a multitask learning scheme consisting of phonetic transcription prediction, text-token prediction, and f_o pattern classification.

Our results indicate that the use of multitask learning and lattice fusion was essential to build an accurate phonemic recognizer. We also verified that a general purpose speech recognizer is not very suitable for building a speaking assessment recognizer since it tends to correct speaker errors. The extra subtasks, text-token prediction and f_o classification, were shown to both contribute to improving estimation accuracy of the main phonemic transcription task.

In this paper, we only focused on the performance of decoders after a single phase of training. However, given the high accuracy of our models, an iterative training scheme that completes the unannotated part of the training dataset by using self-generated labels is also promising. Developing a training method with generated labels is a promising future research direction.

7. REFERENCES

- [1] Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R Mortensen, Graham Neubig, Alan W Black, and Metze Florian, “Universal phone recognition with a multilingual allophone system,” in *Proc. ICASSP*, 2020, pp. 8249–8253.
- [2] Qiantong Xu, Alexei Baevski, and Michael Auli, “Simple and effective zero-shot cross-lingual phoneme recognition,” in *Proc. INTERSPEECH*, 2022, pp. 2113–2117.
- [3] Chihiro Taguchi, Yusuke Sakai, Parisa Haghani, and David Chiang, “Universal automatic phonetic transcription into the international phonetic alphabet,” in *Proc. INTERSPEECH*, 2023, pp. 2548–2552.
- [4] Yoshinori Sagisaka and Hirokazu Sato, “Prosodic rules for speech synthesis from japanese text,” *The Journal of the Acoustical Society of America*, vol. 75, no. S1, pp. S40–S40, 1984.
- [5] Haruo Kubozono, *Organisation of Japanese prosody*, Ph.D. thesis, University of Edinburgh, 1987.
- [6] Kikuo Maekawa, “Corpus of Spontaneous Japanese: Its design and evaluation,” in *Proc. ISCA & IEEE workshop on Spontaneous Speech Processing and Recognition*, 2003, pp. 7–12.
- [7] Hien Ohnaka, Yuma Shirahata, Byeongseon Park, and Ryuichi Yamamoto, “Grapheme-coherent phonemic and prosodic annotation of speech by implicit and explicit grapheme conditioning,” in *Proc. INTERSPEECH*, 2025.
- [8] Yotaro Kubo and Michiel Bacchiani, “Joint phoneme-grapheme model for end-to-end speech recognition,” in *Proc. ICASSP*, 2020, pp. 6119–6123.
- [9] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.
- [10] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. ICASSP*, 2016, pp. 4960–4964.
- [11] Zeyu Zhao and Peter Bell, “Regarding the existence of the internal language model in CTC-based E2E ASR,” in *Proc. ICASSP*, 2025, pp. 1–5.
- [12] Zijian Yang, Minh-Nghia Phan, Ralf Schlüter, and Hermann Ney, “Label-context-dependent internal language model estimation for CTC,” *arXiv preprint arXiv:2506.06096*, 2025.
- [13] Natasha Warner and Takuyuki Arai, “Japanese mora-timing: A review,” *Phonetica*, vol. 58, no. 1–2, pp. 1–25, 2001.
- [14] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour, “Moshi: a speech-text foundation model for real-time dialogue,” Tech. Rep., Kyutai, September 2024.
- [15] “Hugging Face: kyutai/mimi,” <https://huggingface.co/kyutai/mimi> [Accessed: 2025-09-11].
- [16] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al., “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [17] Masanori Morise, “Harvest: A high-performance fundamental frequency estimator from speech signals,” in *Proc. INTERSPEECH*, 2017, pp. 2321–2325.
- [18] Mehryar Mohri, “Finite-state transducers in language and speech processing,” *Computational Linguistics*, vol. 23, no. 2, pp. 269–311, 1997.
- [19] Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura, “A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation,” in *Proc. International Conference on Language Resources and Evaluation (LREC)*, 2008.
- [20] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “Audio augmentation for speech recognition,” in *Proc. INTERSPEECH*, 2015, pp. 3586–3589.
- [21] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proc. INTERSPEECH*, 2019, pp. 2613–2617.
- [22] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proc. International Conference on Machine Learning (ICML)*, 2023, pp. 28492–28518.
- [23] Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari, “JSUT corpus: free large-scale japanese speech corpus for end-to-end speech synthesis,” *arXiv preprint arXiv:1711.00354*, 2017.