POLYPHONE DISAMBIGUATION AND ACCENT PREDICTION USING PRE-TRAINED LANGUAGE MODELS IN JAPANESE TTS FRONT-END

Rem Hida[⋆]*, Masaki Hamada*[⋆]*, Chie Kamada*[†]*, Emiru Tsunoo*[⋆]*, Toshiyuki Sekiya*[⋆]*, Toshiyuki Kumakura*[†]

[⋆] Sony Group Corporation, Tokyo, Japan [†] Sony Corporation of America, San Jose, CA, USA

ABSTRACT

Although end-to-end text-to-speech (TTS) models can generate natural speech, challenges still remain when it comes to estimating sentence-level phonetic and prosodic information from raw text in Japanese TTS systems. In this paper, we propose a method for polyphone disambiguation (PD) and accent prediction (AP). The proposed method incorporates explicit features extracted from morphological analysis and implicit features extracted from pre-trained language models (PLMs). We use BERT and Flair embeddings as implicit features and examine how to combine them with explicit features. Our objective evaluation results showed that the proposed method improved the accuracy by 5.7 points in PD and 6.0 points in AP. Moreover, the perceptual listening test results confirmed that a TTS system employing our proposed model as a front-end achieved a mean opinion score close to that of synthesized speech with ground-truth pronunciation and accent in terms of naturalness.

Index Terms— Japanese text-to-speech, TTS front-end, polyphone disambiguation, accent prediction, pre-trained language models

1. INTRODUCTION

The quality of text-to-speech (TTS) systems has improved in recent years to approach human levels of naturalness, owing to the development of deep learning-based approaches [1, 2]. Traditional TTS systems consist of three parts: a TTS front-end, an acoustic model, and a vocoder. The TTS front-end extracts linguistic features, including phonetic and prosodic features from raw text, the acoustic model converts linguistic features into acoustic features such as a mel-spectrogram, and the vocoder produces waveforms from acoustic features. One of the difficulties in TTS systems is the high language dependency of the front-end. Recent studies have shown that end-to-end TTS systems, especially Japanese ones, require not only phonetic but also prosodic information in order to achieve high quality speech [3, 4]. The similar acoustic information is also effective for other languages [5].

To enable the TTS front-end to precisely estimate both phonetic and prosodic information, two problems must be solved: polyphone disambiguation (PD) and accent prediction (AP). PD estimates the correct pronunciation of polyphonic words. Polyphonic words have multiple candidate pronunciations, and the correct pronunciation depends on the context. AP consists of accent phrase boundary prediction (APBP) and accent nucleus position prediction (ANPP). APBP chunks words into accent phrases, and ANPP determines where pitches change from high to low in each accent phrase.

In this paper, we propose the application of pre-trained language models (PLMs) for performing PD and AP. PLMs have been successfully used for Japanese phrase break prediction [6], for TTS front-ends in other languages [7, 8, 9], and as additional input features for acoustic models [10, 11, 12, 13]. We combine the explicit features derived from morphological analysis and the implicit features derived from PLMs. Explicit features include part-of-speech (POS), pronunciation, accent type, and other kinds of linguistic information. These features contain phonetic and prosodic information that cannot be obtained from raw text. On the other hand, PLMs such as BERT and Flair can provide context-aware information that cannot be obtained from morphological analysis. Thus, explicit features and implicit features are complementary to each other.

Our contributions are summarized as follows:

1. To the best of our knowledge, this work is the first to incorporate PLMs for PD and AP in a Japanese TTS front-end.
2. We investigated strategies for combining explicit linguistic features with PLMs.
3. Our proposed method improves both PD and AP performance in objective evaluations.
4. In the subjective listening test, the proposed method achieved almost the same quality as synthesized speech with ground-truth pronunciation and accent.

2. PROBLEM SETTING

The Japanese TTS front-end aims to convert the input text into linguistic features, including phonetic and prosodic features. Fig. 1 shows a pipeline of the TTS front-end for an input text "京都タワー上空の方に雲がある (There are clouds above Kyoto Tower)." This consists of four components: text normalization, morphological analysis, PD, and AP. The input text is first normalized and then tokenized by a morphological analyzer, which also assigns linguistic features to each word. Subsequently, words with linguistic features are input into the PD and AP modules. Finally, they are converted into a sequence of pronunciations with the pitch (low or high) of each mora. In the following sections, we describe the problem setting of PD and AP. We also briefly explain the conventional approaches to these problems.

2.1. Polyphone disambiguation (PD)

Kanji characters in Japanese have multiple candidate pronunciations, and each one corresponds to a different meaning. For example, in Fig. 1, "方" can be pronounced in two ways with two respective meanings: "ho-o" (direction) and "ka-ta" (way or person). Conventional morphological analysis still fails to estimate the pronunciation of such polyphonic words, because it uses only local context. Therefore, an additional PD module that can consider meaning in the context is necessary for estimating the pronunciation of polyphonic words.

Some studies have explored solving PD by regarding pronunciation estimation (including non-polyphonic words) as a sequence-to-sequence problem, and by applying machine translation approaches [14, 15]. For Mandarin, on the other hand, some studies adopt a classification approach that estimates the correct pinyin of

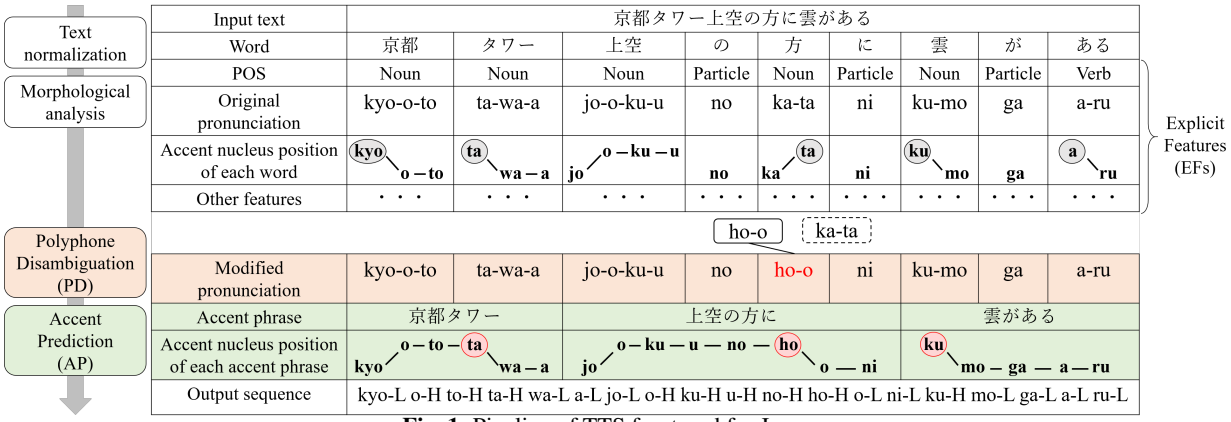

Fig. 1. Pipeline of TTS front-end for Japanese

Table 1. Description of explicit features (EFs). The right three columns show the EFs used in each task.

Feature ID	Feature type	Description	EF_{PD}	EF_{APBP}	EF_{ANPP}
EF1	POS	Part-of-speech tag	✓	✓	✓
EF2	Basic linguistic features	Form and type of conjugation, word type, etc.		✓	✓
EF3	Phonetic features	The number of morae, the first two morae in the pronunciation, etc.		✓	✓
EF4	Prosodic features (word)	Accent type, accent combination type, etc.		✓	✓
EF5	Prosodic features (accent phrase)	Number of words in accent phrase, etc.			✓
EF6	Accent nucleus position rule features	How an accent nucleus change according to rules			✓
EF7	n-gram features	Unigram and bigram information derived from Wikipedia		(Optional)	

the polyphonic character [16, 17, 18]. Because polyphonic words appear only in certain parts of the sentence, we regard PD as a classification problem, similar to the approach for Mandarin.

2.2. Accent prediction (AP)

Japanese is a pitch-accent language that has accent phrases and accent nucleus positions. The accent phrase is a unit in which a pitch ascent occurs, followed by a descent. The accent nucleus position is the mora just before the pitch descends in the accent phrase. These features are not explicitly written in Japanese raw text; however, they are important for prosodic naturalness in Japanese TTS systems [3, 4].

AP consists of two parts: APBP and ANPP. APBP estimates whether each word boundary is an accent phrase boundary in order to chunk words into accent phrases. ANPP estimates the accent nucleus position change of each word in each accent phrase based on the predicted label of APBP in inference. We regard APBP and ANPP as sequence labeling problems, because the correct accent labels depend on the context of the label sequence.

APBP is particularly challenging in the cases of adjacent nouns. For example, there is no boundary between the first adjacent nouns "京都 (Kyoto)" and "タワー (Tower)" in Fig. 1, but there is one between the second adjacent nouns "タワー" and "上空 (above)." Accent nucleus positions also change depending on the context. For example, Fig. 1 shows the accent nucleus position change for "京都". Its original accent nucleus position is the first mora "kyo," as indicated by the circle in Fig. 1. However, its accent nucleus position changes when it is compounded by the following word "タワー." To address these challenges, several approaches have been proposed for AP, such as conventional rule-based methods [19], statistical methods [20], and deep learning methods [15].

3. PROPOSED METHOD

The proposed model combines the explicit features derived from morphological analysis and the implicit features derived from PLMs. As stated previously, explicit features contain not only basic linguistic information but also phonetic and prosodic information, whereas implicit features can represent contextualized information. Thus, explicit and implicit features are complementary.

3.1. Explicit features

We categorize the explicit features (EFs) into seven parts in Table 1. EF1–4 are derived from morphological analysis. EF5 is derived from APBP, EF6 is derived from accent nucleus position rules, and EF7 is derived from text corpora such as Wikipedia. The three right-side columns of Table 1 show the EFs used in each task. Since PD is based on the meaning of words, only EF1 is used. APBP and ANPP are related to phonetic and prosodic phenomena; thus, in addition to EF1, EF2–4 are used for both, EF5 and 6 are used for ANPP, and EF7 is optionally used for APBP, based on [20]. In summary, EF_{PD}, EF_{APBP}, and EF_{ANPP} denote EF1, EF1–4, and EF1–6, respectively.

3.2. Implicit features

In this paper, we propose the use of two types of PLMs as implicit features: BERT and Flair. BERT [21] is a Transformer [22] encoder that is pre-trained by using masked language modeling and next sentence prediction objectives on a raw text corpus. Flair [23] is an LSTM [24] encoder that is pre-trained by using a character-level language modeling objective on a raw text corpus. Thanks to their bi-directional architectures and massive training corpora, both of them have the capability to capture long context and semantic information. One of the differences between BERT and Flair is the token unit. BERT encodes text by subword and then the first subword feature was used as the word feature. Flair encodes text by character, and character-level features were converted to word-level features as in [23]. Additionally, previous studies have shown that BERT contains some syntactic information [25] and can distinguish the sense of words more clearly than Flair [26]. Based on this, we hypothesize that BERT is more suitable for PD and APBP because they are related to syntactic and semantic information, and Flair is more suitable for ANPP, which is related to finer units of information such as phonemes and accents.

Table 2. Estimation examples of BiLSTM models for PD

Example	EF_{PD} (PD3)	EF_{PD} + BERT (PD6)	Reference
今は辛い過去も忘れて幸せに暮らしている。 (Now I forget the **painful** past and live happily.)	"ka-ra-i" (spicy)	**"tsu-ra-i" (painful)**	"tsu-ra-i" (painful)
試合の前日は辛いものを作ることが少なくありません。 (It is not uncommon to make **spicy** food the day before a game.)	"tsu-ra-i" (painful)	**"ka-ra-i" (spicy)**	"ka-ra-i" (spicy)

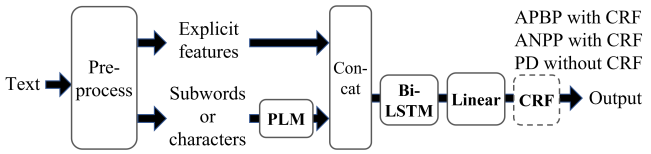

Fig. 2. Proposed model architecture

3.3. Proposed model with explicit and implicit features

Fig. 2 shows the architecture of the proposed model. The text is pre-processed by using the morphological analyzer MeCab[1] with the dictionary UniDic[2], which performs tokenization and POS tagging and extracts other information explicitly. A PLM extracts linguistic information implicitly from token sequences. Subsequently, the embeddings of the explicit features and the PLM are concatenated and input into the BiLSTM layer with a linear layer. For PD, the pronunciation of the polyphonic word is the output. For APBP and ANPP, since they are sequence labeling problems, the hidden states of the BiLSTM layer with a linear layer are then passed through an additional conditional random field (CRF) [27] layer to output the accent phrase boundary and accent nucleus position. The outputs of APBP are binary labels which indicate whether there is an accent phrase boundary before each word, and the outputs of ANPP are multiple labels which describe how the original accent nucleus positions change, based on [20]. The loss functions that we used to train the model are cross-entropy for PD and CRF loss [28] for AP.

4. EXPERIMENTS

4.1. Polyphone disambiguation (PD) objective evaluation

4.1.1. Experimental setup

We compared the performance of the proposed method with baselines as an objective evaluation. We collected 39,353 sentences as an in-house dataset sampled from Wikipedia, TV captions, novels, CSJ [29], and JSUT [30] and manually annotated the pronunciation in each case. The dataset included 39,897 polyphonic words. We split the dataset into training, validation, and test sets of 24,117, 5,156, and 10,080 sentences, respectively. The JNAS corpus [31] was also used as a public test set. In the experiments, we focused on 92 frequently used polyphonic words.

Our baselines were based on morphological analysis. We used MeCab and a KyTea-trained model for comparison. The KyTea-trained model is a morphological analyzer that uses pointwise prediction, and it was trained using the aforementioned training data, based on [32]. The proposed method is based on BiLSTM, explained in Sec. 3.3. We adopted a one-layer BiLSTM (with 512 units), which was implemented with a Flair framework [33]. The model was trained using an SGD optimizer with a mini-batch size of 32. The initial learning rate was 0.1, which was halved each time the validation accuracy did not improve for four consecutive epochs. We stopped training when the learning rate fell below 10^{-4}. As implicit features, the BERT-base model and the Flair model which are pre-trained on Japanese Wikipedia were used[3,4] . When BERT was used as an implicit feature, the last four layers were concatenated[5]. During the training, the parameters of the PLM were fixed[6]. The BiLSTM models were trained with seven different random seeds, and the average accuracy of each test set was reported. We performed statistical hypothesis tests using the t-test.

[1]https://taku910.github.io/mecab/

[2]https://unidic.ninjal.ac.jp/

Table 3. Performance of different systems on PD in the in-house (IH) dataset and JNAS. PD1 and 2 use a morphological analyzer only. EF_{PD} denotes the POS. * denotes a statistically significant difference with the best performance at $p < .05$.

ID	Model	Explicit	Implicit	Accuracy (IH/JNAS)
PD1	MeCab	—	—	81.03*/96.27*
PD2	KyTea-trained [32]	—	—	88.67*/96.63*
PD3	BiLSTM	EF_{PD}	—	88.15*/95.24*
PD4	BiLSTM	—	BERT	94.24/96.64
PD5	BiLSTM	—	Flair	92.86*/96.00*
PD6	BiLSTM	EF_{PD}	BERT	**94.34**/96.72
PD7	BiLSTM	EF_{PD}	Flair	93.46*/**96.77**

4.1.2. Experimental results

Table 3 shows the accuracy of the baselines and BiLSTM models for PD. In both the in-house and JNAS test sets, the BiLSTM model with both explicit and implicit features achieved the highest performance. In the in-house data, the BiLSTM model with EF_{PD} and BERT (PD6) improved accuracy by 5.7 points over PD2, which was the best model without implicit features. We found that BERT was generally superior to Flair as an implicit feature. This is in line with the same findings in the word sense disambiguation task [26], which is similar to PD in terms of requiring semantic information.

Table 2 shows the examples of PD in different BiLSTM systems. The proposed method using EF_{PD} and BERT correctly predicted the pronunciation, while the BiLSTM with only EF_{PD} failed to predict these examples. These examples indicate that BERT contributes to taking context into account and improving PD performance.

4.2. Accent prediction (AP) objective evaluation

4.2.1. Experimental setup

As an objective evaluation, we compared the performance in APBP, ANPP, and overall AP. We collected 9,497 sentences as an in-house dataset sampled from TV captions. We split the dataset into training, validation, and test sets of 7,768, 864, and 865 sentences, respectively. We also used a sub-corpus (basic5000) of JSUT [30] and its accent label[7] as a public test set for overall AP performance. The basic5000 test set consists of 5,000 sentences along with their pronunciations. Sentences with pronunciations that did not match any

[3]https://huggingface.co/cl-tohoku/bert-base-japanese-v2

[4]"ja-forward" and "ja-backward" on https://github.com/flairNLP/flair

[5]Preliminary experiments showed that using all the layers performed as well or worse than using only the last four layers.

[6]Preliminary experiments showed that there was no significant improvement achieved by fine-tuning the PLM.

[7]https://github.com/sarulab-speech/jsut-label

Table 4. F1-score of different systems on APBP in the in-house data. EF_{APBP} denotes the explicit features for APBP (see Sec. 3.1).

Explicit	Implicit	All	Adjacent nouns
Rule-based		91.20*	47.00*
TASET [20]		95.43*	74.46*
EF_{APBP} (+ n-gram)	—	95.44* (95.53*)	77.94* (79.90*)
—	BERT	95.95*	82.77*
—	Flair	95.91*	78.34*
EF_{APBP} (+ n-gram)	BERT	**96.30** (96.23)	**85.61** (85.16)
EF_{APBP} (+ n-gram)	Flair	96.17 (96.18)	82.16* (85.26)

Table 5. Accuracy of different systems on ANPP in the in-house data. EF_{ANPP} denotes the explicit features for ANPP (see Sec. 3.1).

Explicit	Implicit	All	Long accent phrases
Rule-based		83.07*	63.95*
TASET [20]		95.24*	90.83*
EF_{ANPP}	—	95.34*	91.56*
—	BERT	84.38*	78.63*
—	Flair	87.12*	81.51*
EF_{ANPP}	BERT	94.57*	90.19*
EF_{ANPP}	Flair	**95.99**	**92.98**

of the 5-best analysis results from MeCab were excluded. As a result, 4,210 sentences were left for evaluation.

We compared the combination of explicit features (EFs, Sec. 3.1) and implicit features (PLMs, Sec. 3.2) in the proposed method, the rule-based method [19], and TASET[8]. The TASET is an accent prediction tool that uses CRF [20], and it was trained on the same training set. For the BiLSTM experiments, we adopted an architecture and learning setting similar to PD. The only difference was the addition of a CRF layer. We trained the model with five different random seeds and reported the average of the F1-score or accuracy on a test set. For the overall AP evaluation, we compared pairs of different APBP and ANPP models in terms of accuracy. We also performed statistical hypothesis tests using the t-test.

4.2.2. Accent phrase boundary prediction (APBP) results

Table 4 shows the F1-score for all words and adjacent nouns. The results show that the systems using EF_{APBP} and BERT achieved the highest F1-scores. We observed the following two findings. First, incorporating PLMs as implicit features was effective for APBP, where BERT was better than Flair, especially for adjacent nouns. Second, n-gram features were effective for adjacent nouns even when Flair was adopted, but not when BERT was adopted. This implies that the BERT embedding provides n-gram information.

4.2.3. Accent nucleus position prediction (ANPP) results

Table 5 shows the accuracy for all accent phrases and a subset of long accent phrases having more than two words. The results show that the system using EF_{ANPP} and Flair embeddings achieved the highest accuracy. We observed that EF_{ANPP} was a powerful feature for ANPP, when compared with only PLMs. The results also show that there is room for improvement in the model using only EF_{ANPP}, compared to the model using both EF_{ANPP} and Flair. The results indicate that Flair is more effective than BERT for ANPP, which requires finer unit information such as phonemes and accents.

4.2.4. Overall accent prediction results

Table 6 shows the rate of sentences in which the pitch of all moras is correct (Snt-Exact) and the accuracy of the pitch of each mora (Mora-Accuracy). The pair of the APBP model using EF_{APBP} + BERT and the ANPP model using EF_{ANPP} + Flair, which were the best models for each task, outperformed the other pairs in both in-house and JSUT test sets. The in-house data showed that the best system improved over the model using only EFs by 6.0 points in terms of Snt-Exact and 0.5 points in terms of Mora-Accuracy.

[8]https://sites.google.com/site/suzukimasayuki/accent

Table 6. Performance of different systems on overall accent prediction in the in-house (IH) dataset and JSUT.

ID	APBP model	ANPP model	Snt-Exact (IH/JSUT)	Mora-Accuracy (IH/JSUT)
AP0	Rule-based	Rule-based	17.34*/16.35*	90.86*/96.56*
AP1	EF_{APBP} + n-gram	EF_{ANPP}	52.67*/25.91*	96.15*/97.07*
AP2	EF_{APBP} + BERT	EF_{ANPP} + Flair	**58.68/26.98**	**96.66/97.33**

Table 7. MOS evaluation of TTS with 95% confidence intervals computed from the t-distribution for different systems.

Systems	MOS
Oracle symbols	3.69 ± 0.07
Proposed (AP2)	$\mathbf{3.67 \pm 0.07}$
EFs (AP1)	3.46 ± 0.07*
Rule-based (AP0)	3.18 ± 0.08*

4.3. Text-to-speech quality subjective evaluation

To confirm the effectiveness of the proposed model in the Japanese TTS front-end, we carried out a mean opinion score (MOS) test as a subjective evaluation. We adopted Tacotron2 [1] with global style tokens (GST) [34] as an acoustic model and Parallel WaveGAN [35] as a vocoder. We used the sub-corpus (basic5000) of JSUT and its accent label to train both the acoustic model and the vocoder.

In the MOS test, 30 native Japanese speakers were asked to evaluate synthesized speech samples on a 5-point Likert scale. We compared the following four AP systems: (1) **Oracle symbols**: annotated labels. (2) **Rule-based**: a rule-based AP system (AP0 in Table 6). (3) **EFs**: BiLSTM models using only EFs for both APBP and ANPP (AP1 in Table 6). (4) **Proposed**: our best combination of EF_{APBP} + BERT APBP model and EF_{ANPP} + Flair ANPP model (AP2 in Table 6). For the MOS test, 25 utterances were randomly selected from the in-house test set of AP and were then synthesized with the aforementioned input variations from four systems and the reference GST computed by a JSUT sample. We confirmed that the pronunciations of all 25 utterances were correctly estimated by our proposed PD model; thus, the effectiveness of AP was purely evaluated.

Table 7 shows the MOS test results of different systems. It shows that the proposed method outperformed the system without PLMs (AP1) and achieved almost the same speech quality as oracle symbols. We found that the subjective evaluation score was greatly influenced by the accuracy of the adjacent noun accents, which was especially improved by the proposed method. This indicates that using both explicit features and implicit PLM features is effective for the Japanese TTS in terms of naturalness.

5. CONCLUSION

In this paper, we demonstrated the effectiveness and characteristics of PLMs such as BERT and Flair for the Japanese TTS front-end. The objective evaluation results showed that the combination of explicit features from morphological analysis and implicit features from PLMs improved the performance of PD and AP compared with the individual implicit/explicit features. Moreover, the subjective evaluation results showed that our proposed method in the TTS front-end is effective for generating natural speech.

6. ACKNOWLEDGMENTS

The authors would like to thank Takuma Okamoto at the National Institute of Information and Communications Technology for discussing the evaluation design.

7. REFERENCES

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *ICASSP*, 2018, pp. 4779–4783.

[2] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *ICLR*, 2021.

[3] T. Fujimoto, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Impacts of input linguistic feature representation on Japanese end-to-end speech synthesis," in *ISCA Workshop on Speech Synthesis*, 2019, pp. 166–171.

[4] K. Kurihara, N. Seiyama, and T. Kumano, "Prosodic features control by symbols as input of sequence-to-sequence acoustic modeling for neural TTS," *IEICE Tran. Inf. & Syst.*, vol. E104.D, no. 2, pp. 302–311, 2021.

[5] D. S. Ram Mohan, V. Hu, T. H. Teh, A. Torresquintero, C. G.R. Wallis, M. Staib, L. Foglianti, J. Gao, and S. King, "Ctrl-P: Temporal control of prosodic variation for speech synthesis," in *INTERSPEECH*, 2021, vol. 5, pp. 3361–3365.

[6] K. Futamata, B. Park, R. Yamamoto, and K. Tachibana, "Phrase break prediction with bidirectional encoder representations in Japanese text-to-speech synthesis," in *INTERSPEECH*, 2021, pp. 3126–3130.

[7] Z. Bai and B. Hu, "A universal BERT-based front-end model for Mandarin text-to-speech synthesis," in *ICASSP*, 2021, pp. 6074–6078.

[8] A. Talman, A. Suni, H. Celikkanat, S. Kakouros, J. Tiedemann, and M. Vainio, "Predicting prosodic prominence from text with pre-trained contextualized word representations," in *NoDaLiDa*, 2019, pp. 281–290.

[9] B. Yang, J. Zhong, and S. Liu, "Pre-trained text representations for improving front-end text processing in Mandarin text-to-speech synthesis," in *INTERSPEECH*, 2019, pp. 4480–4484.

[10] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, "Pre-trained text embeddings for enhanced text-to-speech synthesis," in *INTERSPEECH*, 2019, pp. 4430–4434.

[11] T. Kenter, M. Sharma, and R. Clark, "Improving the prosody of RNN-based English text-to-speech synthesis by incorporating a BERT model," in *INTERSPEECH*, 2020, pp. 4412–4416.

[12] Z. Hodari, A. Moinet, S. Karlapati, J. Lorenzo-Trueba, T. Merritt, A. Joly, A. Abbas, P. Karanasou, and T. Drugman, "CAMP: A two-stage approach to modelling prosody in context," in *ICASSP*, 2021, pp. 6578–6582.

[13] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "PnG BERT: Augmented BERT on Phonemes and Graphemes for Neural TTS," in *INTERSPEECH*, 2021, pp. 151–155.

[14] J. Hatori and H. Suzuki, "Japanese pronunciation prediction as phrasal statistical machine translation," in *IJCNLP*, 2011, pp. 120–128.

[15] N. Kakegawa, S. Hara, M. Abe, and Y. Ijima, "Phonetic and prosodic information estimation from texts for genuine Japanese end-to-end text-to-speech," in *INTERSPEECH*, 2021, pp. 126–130.

[16] K. Park and S. Lee, "g2pM: A neural grapheme-to-phoneme conversion package for Mandarin Chinese based on a new open benchmark dataset," in *INTERSPEECH*, 2020, pp. 1723–1727.

[17] H. Sun, X. Tan, J. W. Gan, S. Zhao, D. Han, H. Liu, T. Qin, and T. Y. Liu, "Knowledge distillation from BERT in pre-training and fine-tuning for polyphone disambiguation," in *ASRU*, 2019, pp. 168–175.

[18] Z. Cai, Y. Yang, C. Zhang, X. Qin, and M. Li, "Polyphone disambiguation for Mandarin Chinese using conditional neural network with multi-level embedding features," in *INTERSPEECH*, 2019, pp. 2110–2114.

[19] Y. Sagisaka and H. Sato, "Accentuation rules for Japanese word concatenation," *IEICE Trans. Inf. & Syst. (Japanese Edition), D*, vol. 66, no. 7, pp. 849–856, 1983.

[20] M. Suzuki, R. Kuroiwa, K. Innami, S. Kobayashi, S. Shimizu, N. Minematsu, and K. Hirose, "Accent sandhi estimation of Tokyo dialect of Japanese using conditional random fields," *IEICE Trans. Inf. & Syst.*, vol. 100, no. 4, pp. 655–661, 2017.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[23] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *COLING*, 2018, pp. 1638–1649.

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

[25] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *NAACL-HLT*, 2019, pp. 4129–4138.

[26] G. Wiedemann, S. Remus, A. Chawla, and C. Biemann, "Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings," in *KONVENS*, 2019, pp. 161–170.

[27] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001, pp. 282–289.

[28] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," 2015.

[29] K. Maekawa, "Corpus of spontaneous Japanese: its design and evaluation," in *ISCA/IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003, pp. 7–12.

[30] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis," *arXiv preprint arXiv:1711.00354*, 2017.

[31] ASJ Japanese Newspaper Article Sentences Read Speech Corpus (JNAS), http://research.nii.ac.jp/src/JNAS.html.

[32] G. Neubig and S. Mori, "Word-based partial annotation for efficient corpus construction," in *LREC*, 2010, pp. 2723–2727.

[33] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, "FLAIR: An easy-to-use framework for state-of-the-art NLP," in *NAACL Demonstrations*, 2019, pp. 54–59.

[34] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *ICML*, 2018, pp. 5180–5189.

[35] R. Yamamoto, E. Song, and J. M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*, 2020, pp. 6199–6203.