10th ISCA Speech Synthesis Workshop
20-22 September 2019, Vienna, Austria

Subword tokenization based on DNN-based acoustic model for end-to-end prosody generation

Masashi Aso, Shinnosuke Takamichi, Norihiro Takamune, and Hiroshi Saruwatari

Graduate School of Information Science and Technology, The University of Tokyo, Japan.

aso@g.ecc.u-tokyo.ac.jp,
{shinnosuke_takamichi,norihiro_takamune,hiroshi_saruwatari}@ipc.i.u-tokyo.ac.jp

Abstract

This paper presents a method for determining subword units for end-to-end prosody generation. End-to-end prosody generation using deep neural networks (DNNs) is expected to directly generate a prosody sequence from text without any professional knowledge in the target language. In natural language processing, language model-based language-independent subword tokenization was previously proposed for determining subwords suitable for end-to-end language processing. However, the subwords determined by the language models are not appropriate for end-to-end speech processing. In this paper, we propose a language-independent algorithm for determining subwords that maximize acoustic model likelihoods. The proposed algorithm iterates expectation-maximization (EM)-based training of DNN acoustic models and likelihood-based construction of the subword vocabulary. In the experimental evaluation, we discuss the stability of the EM-based training and analyze subword vocabularies determined by the conventional language model-based and proposed acoustic model-based methods.

Index Terms: end-to-end, prosody generation, subword, deep neural network, expectation maximization

1. Introduction

Spoken language processing without professional knowledge in the target language enables a variety of applications. For example, unsupervised representation of speech [1] is used for clarifying how young children learn to talk before they learn to read and write. Discovering phonetic inventory [2] enables speech-to-text recognition for unwritten languages. In speech synthesis, *end-to-end text-to-speech synthesis* [3, 4, 5] that directly generates speech features or waveforms from text has a great potential to synthesize voices in not only rich-resourced languages but also low-resourced languages that lack professional linguistic knowledge. DNNs with sequence-to-sequence (seq2seq) conversion [6, 7] have been successfully applied to character-to-speech-feature conversion. Currently, end-to-end text-to-speech systems without professional knowledge have succeeded in a very limited number of rich-resourced languages, such as English [3], Spanish [4], and Chinese [8], and the systems for other languages still require language-dependent front-end processing (e.g., Japanese[9]).

Towards purely end-to-end text-to-speech synthesis, an issue targeted in this paper is *language units* of DNN inputs. The basic unit is a character [3, 4, 8] but other choices (e.g., phoneme [9], word [10, 11]) are available[1] as shown in Fig. 1. The use of phonemes or characters (left in Fig. 1) can capture segmental features (i.e., spectral features) but suffers from poor prediction of suprasegmental features (i.e., F_0). On the other hand, the use of words (right in Fig. 1) can capture suprasegmental features but suffers from poor prediction of segmental features. It also suffers from a sparsity problem, i.e., prosody prediction becomes inaccurate in an open word-vocabulary. Therefore, we expect that language units suitable for spectrum generation and those suitable for F_0 generation are different. In this paper, we focus on the latter, *end-to-end prosody generation*. Previously [14], we utilized subword tokenization on the basis of language models [15]. The approach is to deal with the open vocabulary issue discussed above and the subword tokenization breaks up rare words into subword units [16] (e.g., "language" into "lang" and "uage"). The language model-based tokenization [15] constructs a vocabulary of frequently appeared subwords and tokenizes text into the most likely (i.e., higher language model likelihood) subword sequence. The use of these subwords is expected to avoid the sparsity problem and capture suprasegmental features better than the use of characters. However, subwords tokenized by language models are not appropriate for prosody generation. In other words, subwords should be tokenized to accurately predict prosody features from subword.

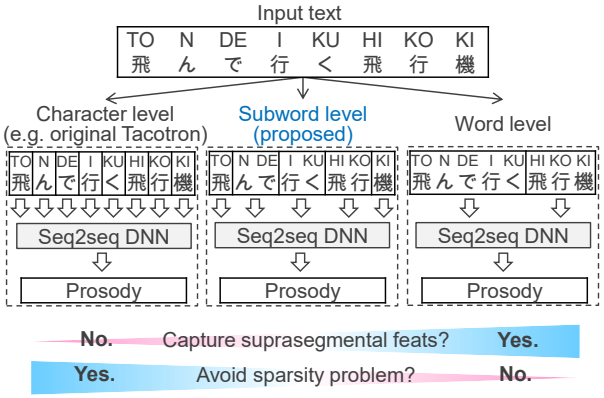

Figure 1: *Problem definition: how to tokenize input text for prosody generation. Our proposed method is language-independent tokenization trained using acoustic model likelihoods.*

In this paper, we propose a language-independent unsupervised algorithm for determining subword units suitable for end-to-end prosody generation. By using aligned pairs of characters and F_0 segments, the proposed algorithm determines the subword vocabulary on the basis of acoustic model likelihoods, i.e., determining subwords that accurately predict the F_0 from subwords. The proposed algorithm iterates EM-based training of the DNN acoustic models and likelihood-based vocabulary

[1] The language-dependent hierarchical use of these components was also proposed [12, 13].

10.21437/SSW.2019-42

construction. Regarding subword sequences as hidden variables, we formulate the generative model of the prosody sequence from text as a hidden Markov model (HMM) whose output probability is written as DNNs. Previously [14], we used Viterbi-based training of the acoustic models, and EM-based training formulated in this paper is its extended version that does not require the Viterbi approximation. The likelihood-based vocabulary construction *acoustically* collects the most probable subwords. The experimental results demonstrate that 1) the EM-based training is empirically stable, 2) the EM-based training improves acoustic model likelihoods more than the Viterbi-based training [14], 3) the acoustic model-based subword deletion improves acoustic model likelihood more than the liguistic model-based one and 4) subwords tokenized by our proposed method might be more related to accent phrases than those tokenized by a conventional language model-based method [15].

2. Language model-based subword tokenization

Language model-based subword tokenization [15] has two steps: training (**Section 2.1**) and tokenization (**Section 2.2**). The training step estimates a subword vocabulary (i.e., a set of subwords) $\mathcal{V}$ and unigram probabilities of its subwords. The tokenization step tokenizes an input text on the basis of the estimated subword vocabulary and the unigram probabilities. The model can be thought of a HMM-like model whose hidden variables are subword sequences.

2.1. Training step

To estimate the subword vocabulary $\mathcal{V}$, we first heuristically make a reasonably big seed vocabulary from the training text corpus. Then, we iterate estimation step (**Section 2.1.1**) and subword deletion step (**Section 2.1.2**) until the vocabulary size (i.e., the number of subwords in the vocabulary) reaches a predetermined desired vocabulary size.

2.1.1. Estimation step

This step estimates unigram probability while fixing $\mathcal{V}$. Here, let $\boldsymbol{X}$ be a sentence, $\boldsymbol{z}$ be a segmentation candidate, s_* be a segment node, $f(\cdot;\boldsymbol{X})$ be mapping from a segment node to a corresponding subword. Examples are follows.

$\boldsymbol{X}$: "language"
$\boldsymbol{z}$: [s_{123}, s_{45}, s_{67}]
$\mathcal{V}$: { "lan", "gu", "age", $\cdots$ }
$f(s_{123};\boldsymbol{X})$ = "lan"
$f(s_{45};\boldsymbol{X})$ = "gu"
$f(s_{67};\boldsymbol{X})$ = "age"

Probability of $\boldsymbol{X}$ given $\boldsymbol{z}$ is calculated as:

$$P(\boldsymbol{X}|\boldsymbol{z}) = \prod_{s\in\boldsymbol{z}} P^{\text{uni}}(f(s;\boldsymbol{X})). \tag{1}$$

The unigram probability $P^{\text{uni}}(x)$ of a subword x ($\in \mathcal{V}$) is estimated to maximize the following likelihoods.

$$P(\boldsymbol{X}) = \sum_{\text{all } \boldsymbol{z}} P(\boldsymbol{X}|\boldsymbol{z})\, P(\boldsymbol{z}). \tag{2}$$

This model is shown in the top of Fig. 2 and the Baum-Welch algorithm can apply to this model.

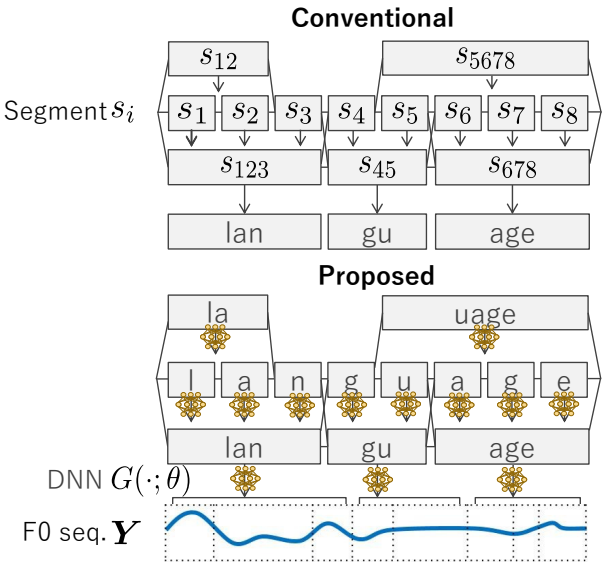

Figure 2: *Conventional and proposed models.*

2.1.2. Subword deletion step

This step deletes unnecessary subwords from $\mathcal{V}$ by the unigram probabilities estimated in the training step. First, we compute the sentence likelihood $\mathcal{L}$. Next, we compute the sentence likelihood $\mathcal{L}_x$ when a subword x is removed from $\mathcal{V}$. Note that we approximate $\mathcal{L}_x$ by the sentence likelihood estimated in the E-step of the Baum-Welch algorithm with unigram probability fixed. The loss value deleting x is calculated as $\mathcal{L} - \mathcal{L}_x$. Then, we sort subwords in $\mathcal{V}$ by the loss values and remove η (e.g., $\eta = 0.25$) of subwords with lower loss values from $\mathcal{V}$. Note that subword vocabulary must contain all single characters in the training data to prevent all segmentation candidates from including an unknown subword.

2.2. Tokenization step

At runtime, first, a lattice structure toward $\boldsymbol{X}$ is constructed on the basis of the subword vocabulary estimated in the estimation step. Then, the most probable segmentation $\boldsymbol{z}^*$ in all segmentation candidates over the lattice structure is defined by

$$\boldsymbol{z}^* = \operatorname{argmax} P(\boldsymbol{z}|\boldsymbol{X}), \tag{3}$$

and this can be obtained by the Viterbi algorithm using $\mathcal{V}$ and $P(x)$ estimated in the training step. In our previous work[14], subwords given by this Viterbi path were used as the inputs of the DNN.

3. Proposed acoustic model-based subword tokenization

Inspired by the language model-based subword tokenization [15] described in **Section 2**, this section proposes acoustic model-based subword tokenization for estimating a subword vocabulary appropriate for prosody prediction. The acoustic model that predicts prosody from a subword is given as a DNN $G(\cdot;\theta)$, and model parameters of the DNN, θ, are estimated using the EM algorithm. The same as in **Section 2**, the proposed algorithm has two steps: training and tokenization, which are described below.

3.1. Formulation

The training data is parallel data of sentences and continuous F_0 sequences [17]. Here, we re-define a segmentation candidate $\boldsymbol{z}$ to have alignments between a subword and continuous F_0 segments. Let $\boldsymbol{Y}$ be a continuous F_0 sequence. For $g\left(s;\boldsymbol{Y}\right)$, a variable-length continuous F_0 segment corresponding to a segment node $s\left(\in\boldsymbol{z}\right)$ is resampled by interpolation to a fixed-length segment and then we obtain the lower-order components of discrete cosine transform of the resampled segment [10]. For instance, when $f(s_{123};\boldsymbol{X}) =$ "lan", $g\left(s_{123};\boldsymbol{Y}\right)$ is calculated using a continuous F_0 segment corresponding to "lan." A DNN acoustic model $G\left(\cdot;\theta\right)$ predicts $g\left(s_*;\boldsymbol{Y}\right)$ from $f(s_*;\boldsymbol{X})$.

A probability of $\boldsymbol{Y}$ (approximated by $g(\cdot;\boldsymbol{Y})$) given $\{\boldsymbol{z},\boldsymbol{X},\theta\}$ is calculated as:

$$P\left(\boldsymbol{Y}|\boldsymbol{z},\boldsymbol{X},\theta\right)\simeq\prod_{s\in\boldsymbol{z}}P\left(g(s;\boldsymbol{Y})|s,\boldsymbol{X},\theta\right), \tag{4}$$

$$P\left(g(s;\boldsymbol{Y})|s,\boldsymbol{X},\theta\right)=\mathcal{N}\left(g(s;\boldsymbol{Y});G\left(f(s;\boldsymbol{X});\theta\right),\sigma^2\boldsymbol{I}\right), \tag{5}$$

where $\mathcal{N}\left(\cdot;G\left(\cdot;\theta\right),\sigma^2\boldsymbol{I}\right)$ is the Gaussian distribution with a mean $G\left(\cdot;\theta\right)$ and a covariance matrix $\sigma^2\boldsymbol{I}$, where σ is a constant standard deviation and $\boldsymbol{I}$ is an identity matrix. In training, we optimize acoustic model likelihoods as follows.

$$P\left(\boldsymbol{Y}|\boldsymbol{X},\theta\right)=\sum_{\text{all }\boldsymbol{z}}P\left(\boldsymbol{Y}|\boldsymbol{z},\boldsymbol{X},\theta\right)P\left(\boldsymbol{z}|\boldsymbol{X},\theta\right). \tag{6}$$

The proposed model can be viewed as a HMM whose output probability is shown as Eq. (5) and the bottom of Fig. 2. In this paper, we set transition probabilities to equal probabilities.

3.2. Training step

As described in **Section 2.1**, we iterate the estimation and subword deletion steps until the vocabulary size reaches a predetermined desired vocabulary size.

3.2.1. Estimation step

The maximization of $P(\boldsymbol{Y}|\boldsymbol{X};\theta)$ can be expressed as the optimization of the following $\mathcal{Q}$ function with the EM algorithm:

$$\mathcal{Q}(\theta,\theta^{(\text{old})})=\sum_{i=1}^{N}P(s_i|\boldsymbol{Y},\boldsymbol{X};\theta^{(\text{old})})\log P(g(s_i;\boldsymbol{Y})|s_i,\boldsymbol{X};\theta), \tag{7}$$

where s_i $(i=1,\cdots,N)$ means all segment nodes.

At the E-step of the EM algorithm, we estimate the posterior probability $\gamma_i = P(s_i|Y,X;\theta^{(\text{old})})$, where i $(i=1\cdots N)$ is the subword index. By the Baum-Welch algorithm, forward probability α_i, backward probability β_i and the posterior probability are given as

$$\alpha_i\simeq P(s_i,\boldsymbol{Y}_{\leq i}|\boldsymbol{X};\theta^{(\text{old})}), \tag{8}$$

$$.\beta_i\simeq P(\boldsymbol{Y}_{>i}|s_i,\boldsymbol{X};\theta^{(\text{old})}), \tag{9}$$

$$\gamma_i=\alpha_i\beta_i/P(\boldsymbol{Y}|\boldsymbol{X};\theta^{(\text{old})}), \tag{10}$$

where $\boldsymbol{Y}_{\leq i}$ is a continuous F_0 sequence made up of frames before the end frame of s_i in $\boldsymbol{Y}$ and $\boldsymbol{Y}_{>i}$ is a continuous F_0 sequence made up of frames after the end frame of s_i in $\boldsymbol{Y}$. Note that α_i and β_i are approximated value because the observation data in the HMM model is the a fixed-length segment converted from a continuous F_0 sequence by $g(\cdot;\boldsymbol{Y})$.

At the M-step of the EM algorithm, we update DNN parameters θ by maximizing the following $\mathcal{Q}$ function with γ_i:

$$\mathcal{Q}(\theta,\theta^{(\text{old})})=\sum_{i=1}^{N}\frac{-\gamma_i}{2\sigma^2}|g(s_i,\boldsymbol{Y})-G(f(s_i,\boldsymbol{X});\theta)|^2 + \text{const.}. \tag{11}$$

Since this can be viewed as minimization of the weighted sum of squared errors [18], we can train the DNN by a gradient-descent optimization algorithm. Note that because of a gradient method in the M-step, the above algorithm does not guarantee the monotonic non-decreasing of the likelihood, whereas the standard EM algorithm does.

3.2.2. Subword deletion step

The same as in **Section 2.1.2**, we calculate loss of acoustic model likelihood when a subword is deleted by the E-step with DNN parameters θ fixed and delete subwords with lower loss values.

3.3. Tokenization step

At runtime, prosody information is not observed. Thus, we calculate subword-level unigram probability $P^{\text{uni}}(x)$ in training and use the language model-based tokenization described in **Section 2.2**. Here, the unigram probability is expressed using γ_i $(i=1,\cdots,N)$ estimated in the E-step and output probability (5) by $G(\cdot;\theta)$ estimated in the M-step as

$$P^{\text{uni}}\left(x\right)=\frac{\sum_{j\in\mathcal{J}(x)}\gamma_j P(f(s_j,\boldsymbol{X})|s_j,\boldsymbol{X},\theta)}{\sum_{i=1}^{N}\gamma_i}, \tag{12}$$

where $\mathcal{J}(x)$ is a set of j such that $x=f(s_j;\boldsymbol{X})$. Of course, the segmentation can be re-estimated using the EM algorithm with the trained acoustic model [19, 20].

4. Experimental evaluation

4.1. Experimental condition

We conducted an experimental evaluation using 21,006 utterances collected from two Japanese corpora (basic5000 subcorpus from JSUT [21] and JNAS [22]). Before the training, we obtain alignments between character and continuous F_0 sequences. To align the character and continuous F_0 sequences, we used fast_align [23] for character-phoneme alignment and Julius [24] for phoneme-F_0 alignment. We split the utterances into 90% for training and 10% for evaluation. We extracted continuous F_0 sequences by WORLD [25] (D4C edition [26]). The sampling frequency was 16 kHz. The frame shift was set to 5 ms. To deal with the multi-speakers' utterances, we normalize a continuous F_0 sequence for each utterance to have zero mean and unit variance. The acoustic model was a Feed-Forward neural network that has a word embedding layer as an input layer, 3×512 gated linear units as hidden layers, a linear layer as an output layer. An optimizer of the DNN training was Adagrad [27] with a 0.01 learning rate. We assigned 1.0 to the constant standard deviation σ of the Gaussian distribution. Sentencepiece [15] was used for a conventional language model-based subword tokenization. Subword vocabulary is initialized by a set of 138,585 subwords built from training data with enhanced suffix arrays, a data structure that can efficiently calculate all subwords [28]. In the proposed training, we iterate the estimation and subword deletion steps until the vocabulary

size reaches a predetermined desired vocabulary size. In the estimation step, we did 30 EM iterations. In the M-step of the EM algorithm, we used minibatch training, the number of the minibatch iterations in one M-step was 30 and the minibatch size was 1,000 sentences. In the subword deletion step, At the subword deletion step, the rate of deletion η is set at 0.25 unless the vocabulary size reaches a desired vocabulary size.

4.2. Experimental evaluation of EM algorithm

In this section, we evaluated the proposed training step with a desired vocabulary size set at 4000. First, we investigated on the learning curve convergence of the proposed method and then compare the proposed and conventional methods by acoustic model log-likelihood. As described in **Section 3.2.1**, the proposed EM algorithm given by a subword vocabulary does not guarantee theoretical convergence. We investigated whether or not the algorithm performs empirical convergence. We calculated the negative log likelihood for the training data at each step of the EM iteration. The top of Fig. 3 shows the five learning curves during 30 EM iterations in the proposed method and the bottom of Fig. 3 enlarges the learning curves from the 15th to the 30th EM iteration. since the log-likelihoods slightly decrease at some steps as shown in the bottom of Fig. 3, the learning curves are not monotonic non-decreases. However, since the log-likelihoods globally increase as shown in the top of Fig. 3, we can confirm the empirical convergence of the proposed EM-based training.

As described in **Section 3.2.1**, the proposed EM-based training given a subword vocabulary is an extended version of the Viterbi-based training with subword sequence approximation [14] and as described in **Section 3.2.2**, the proposed subword vocabulary keeps acoustic model likelihoods high when subwords are deleted from the vocabulary whereas the conventional one keeps linguistic model likelihoods high. We demonstrated that the proposed EM-based training achieves higher acoustic model likelihoods $P\left(\boldsymbol{Y}|\boldsymbol{X},\theta\right)$ than the Viterbi-based training and that acoustic model-based subword deletion performs higher acoustic model likelihood than language-based subword deletion. We calculated the acoustic model likelihoods for training and evaluation data among three methods: Viterbi (Viterbi-based training and language model-based subword deletion), Est. (EM-based training and language model-based subword deletion) and Est. & Del. (EM-based training and acoustic model-based subword deletion). In the Viterbi-based training, a DNN is trained toward the Viterbi-path 900 times. Table 1 lists the results. Both training and evaluation data in Table 1 show that the proposed EM-based training achieves higher likelihoods than the Viterbi-based training and that the subword deletion step based on a acoustic model keeps acoustic model likelihoods higher than that based on a language model. Therefore, we can safely say the proposed method is effective towards the sentence likelihood.

4.3. Analysis of subword vocabulary

In this section, we looked into subword vocabularies generated by the proposed and conventional methods with a desired vocabulary size set at 8000. Here, we used twice as many subwords as in **Section 4.2** to make clear the difference of the subword vocabularies of two models, because both of the vocabularies share all single characters in the training data and the number of them is about 2,500. We compared conventional language model-based and proposed acoustic model-based subword vocabularies with the length of the subwords in the vocabularies and the parts-of-speech (POSs) of the subwords. As described in **Section 3.2.2**, subwords in a proposed vocabulary contribute largely to acoustic model likelihood. First, we compared the length of the subwords included in conventional and proposed vocabularies. Fig. 4 shows the histogram of the length of the subwords. From this figure, we can see that the subword vocabulary by the proposed method contains more two-character subwords than the conventional method. This suggests that shortening a subword length contribute more to high accuracy of DNN-based prosody prediction than to high frequency of the subword combination. Moreover, we analyzed what kind of POSs forms a subword unit. First, we tokenized the evaluation data into subword sequences with conventional and proposed subword vocabularies. Next, we tagged the sentences with POS using MeCab [29], a Japanese POS tagger. Finally, we analyzed whether the subwords consist of a particular POS tag combination. We observed more subwords in a proposed vocabulary are constituted by two POS tags (noun + particle, particle + noun, verb + particle, and particle + verb) than by the conventional method, as listed in Table 2. This result is natural for Japanese language because the number of such POS tag combination is enormous, i.e., the language model likelihood becomes smaller and such subwords will be split or merged. On the other hand, such combinations consist of Japanese accent phrases, i.e., the acoustic model likelihood becomes larger and such subwords will be used.

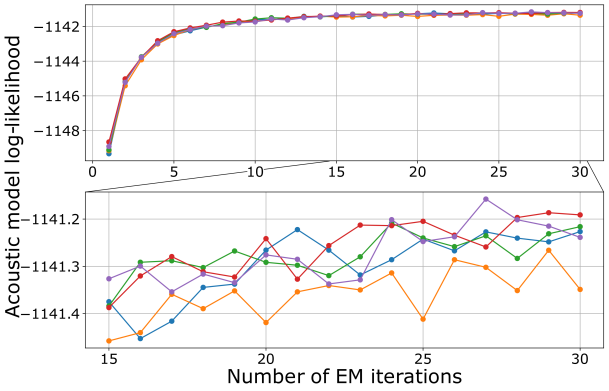

Figure 3: *Five learning curves of proposed method.*

	Viterbi [14]	Est.	Est. & Del.
Training data	-1,145	-1,141	-1,059
Evaluation data	-1,152	-1,131	-1,077

Table 1: *Comparison of acoustic model log-likelihood between different models.*

5. Conclusion

This paper presented an acoustic model-based subword tokenization for obtaining subword units appropriate for end-to-end prosody generation. The proposed algorithm iterated 1) expectation-maximization (EM)-based training of a deep neural network (DNN) acoustic model that predicts a fixed-length F_0 segment from a subword and 2) acoustic model likelihood-based subword vocabulary construction. The experimental evaluation demonstrated the stability of the EM-based training and improvements of the acoustic model likelihoods. Also, we found some differences between subword vocabularies of the conventional language model-based and proposed acous-

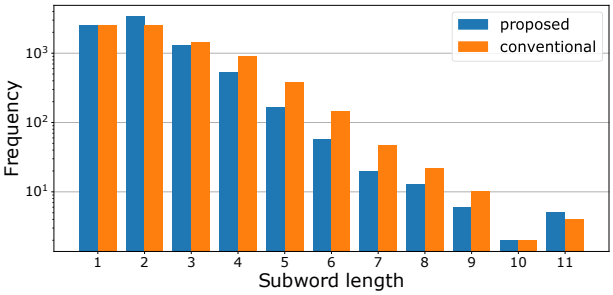

Figure 4: *Comparison of subword length histogram between different subword vocabularies.*

POS constitution	language model [15]	acoustic model
noun + particle	918	1,142
particle + noun	1,097	1,454
verb + particle	1,590	1,941
particle + verb	1,736	2,108

Table 2: *Comparison of the frequency of particular POS constitutions in a subword unit between different models. A particle means a postpositional particle of Japanese.*

tic model-based methods. For future work, we will investigate the effectiveness of the proposed method in end-to-end prosody generation.

Acknowledgements: Part of this research and development work was supported by JSPS KAKENHI 18K18100 and 17H06101.

6. References

[1] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 7634–7638.

[2] P. K. Muthukumar and A. W. Black, "Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 2594–2598.

[3] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," vol. abs/1609.03499, 2017. [Online]. Available: https://arxiv.org/abs/1703.10135

[4] J. Sotelo, S. Mehri, K. Kumar, K. K. J. F. Santos, A. C. ville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, Toulon, France, Apr. 2017.

[5] W. Ping, K. Peng, A. Gibiansky, S. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, Vancouver, Canada, Apr. 2018.

[6] K. Cho, D. Bahdanau, F. Vougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[7] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, Lisbon, Portugal, Sep. 2015.

[8] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *Proc. ICASSP*, Calgary, Canada, Apr. 2018, pp. 4789–4793.

[9] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of Enhanced Tacotron Text-to-speech Synthesis Systems with Self-attention for Pitch Accent Language," in *Proc. ICASSP*, MrigAton, United Kingdom, May 2019, pp. 6905–6909.

[10] Y. Ijima, N. Hojo, and R. Masumura, "Prosody Aware Word-level Encoder Based on BLSTM-RNNs for DNN-based Speech Synthesis," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 764–768.

[11] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Word embedding for recurrent neural network based TTS synthesis," in *Proc. ICASSP*, Brisbane, Australia, Apr. 2015, pp. 4879–4883.

[12] H. Ming, L. He, H. Guo, and F. K. Soong, "Feature reinforcement with word embedding and parsing information in neural TTS," *arXiv*, vol. abs/1901.00707, 2019.

[13] H. Guo, F. K. Soong, L. He, and L. Xie, "Exploiting syntactic features in a parsed tree to improve end-to-end TTS," *arXiv*, vol. abs/1904.04764, 2019.

[14] T. Akiyama, S. Takamichi, and H. Saruwatari, "Prosody-aware subword embedding considering Japanese intonation systems and its application to DNN-based multi-dialect speech synthesis," in *Proc. APSIPA*, Hawaii, U.S.A., Nov. 2018, pp. 660–664.

[15] T. Kudo, "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates," in *Proc. ACL*, Melbourne, Australia, Jul. 2018, pp. 66–75.

[16] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proc. ACL*, Berlin, Germany, Aug. 2016, pp. 1715–1725.

[17] K. Yu and S. Young, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1071–1079, Jul. 2011.

[18] K. Tokuda, K. Hashimoto, K. Oura, and Y. Nankaku, "Temporal modeling in neural network based statistical parametric speech synthesis," in *ISCA Speech Synthesis Workshop*, Sunnyvale, U.S.A., Sep. 2016, pp. 113–118.

[19] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, Jun. 2000, pp. 1315–1318.

[20] S. Takamichi, T. Toda, Y. Shiga, S. Sakti, G. Neubig, and S. Nakamura, "Parameter generation methods with rich context models for high-quality and flexible text-to-speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 239–250, 2014.

[21] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis," vol. abs/1711.00354, 2017.

[22] "JNAS," http://research.nii.ac.jp/src/JNAS.html, accessed: 2018-12-05.

[23] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proc. NAACL*, Atlanta, U.S.A., May 2013, pp. 644–648.

[24] "Julius," https://github.com/julius-speech/julius, accessed: 2018-12-05.

[25] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE transactions on information and systems*, vol. E99-D, no. 7, pp. 1877–1884, Jul. 2016.

[26] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[27] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *EURASIP Journal on Applied Signal Processing*, vol. 12, pp. 2121–2159, Jul. 2011.

[28] M. I. Abouelhoda, S. Kurtz, and E. Ohlebusch, "Replacing suffix trees with enhanced suffix arrays," *Journal of discrete algorithms*, vol. 2, no. 1, pp. 53–86, 2004.

[29] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," in *Proc. EMNLP*, Barcelona, Spain, Jul. 2004, pp. 230–237.