Korean Tokenization for Beam Search Rescoring in Speech Recognition

Kyuhong Shim, Hyewon Bae, and Wonyong Sung
Department of Electrical and Computer Engineering
Seoul National University
Seoul, South Korea
Email: skhu20@snu.ac.kr, hyewon0309@snu.ac.kr, wysung@snu.ac.kr

Abstract—The performance of automatic speech recognition (ASR) models can be greatly improved by proper beam-search decoding with an external language model (LM). There has been an increasing interest in Korean speech recognition, but not many studies have been focused on the decoding procedure. In this paper, we propose a Korean tokenization method for neural network-based LM used for Korean ASR. Although the common approach is to use the same tokenization method for external LM as the ASR model, we show that it may not be the best choice for Korean. We propose a new tokenization method that inserts a special token, SkipTC, when there is no trailing consonant in a Korean syllable. Our experiments show that the proposed approach achieves a lower word error rate compared to the same LM model without SkipTC. In addition, we are the first to report the ASR performance for the recently introduced large-scale 7,600h Korean speech dataset.

Index Terms—speech recognition, Korean, tokenization, beam search, rescoring, language model

I. INTRODUCTION

For end-to-end automatic speech recognition (ASR) systems, language models (LMs) are widely used for reducing the error rate. LMs are especially useful when the speech audio itself is ambiguous as to which sentence is pronounced. External LM produces an additional likelihood score for the current beam during beam search decoding, where the score from the acoustic model (AM) and the LM are jointly considered. Extensive studies have been conducted to improve the quality of LM, mainly focused on exploiting more powerful architectures and training with a vast amount of data [1].

Tokenization is the key ingredient for the performance of LM, yet its effect on ASR decoding has not been much studied. Tokenization splits a sentence into a sequence of linguistic tokens, where the tokens compose a vocabulary used for speech recognition or language modeling. For the efficiency of the decoding, the same tokenization method is usually used for acoustic and language models, although exploiting the same vocabulary for ASR and LM is not always necessary. For example, two-pass rescoring [2] evaluates the complete sentences after the initial beam search for the second pass so that the token boundary alignment is not very important.

Recently, studies on Korean speech recognition have been actively conducted [3]. These advances in Korean ASR are mainly boosted by the release of large-scale Korean speech corpus, such as KsponSpeech [4] which includes about 970h of conversational dialogues. Several studies have achieved fairly good recognition accuracy in Korean ASR by adopting neural networks as AM [5]. However, the decoding process and corresponding LMs have not been very much studied for Korean ASR. For example, external LM was not employed [5] or only an N-gram LM was utilized [6].

In this paper, we propose a new Korean tokenization method that utilizes the characteristics of the Korean syllable composition for two-pass decoding of ASR with LM. The key idea of the proposed method is to add an auxiliary token, making the tokenized sequence more regular by forcing two types of tokens to appear alternately. Figure 1 shows the structure of Korean syllables (see Section II-B for details).

We can observe that each Korean syllable can be separated into two components: (1) leading consonant + vowel and (2) trailing consonant, where the former always exists but the latter is optional. During the tokenization, we insert the auxiliary token, named SkipTC token, when there is no trailing consonant in a Korean syllable. As a result, LM can always encounter alternating token types (1) and (2) and better learn the characteristics of Korean sentences. We employ the proposed tokenization method for RNN-based LM, and then use the LM for the rescoring.

Our experiments show that using the proposed SkipTC token for LM tokenization reduces the word error rate and the syllable error rate of Korean ASR by 2.68% and 0.73%, respectively. We identify that the improvement comes from the enhanced ability of LM to model the likelihood, presented as the reduced perplexity. We evaluate the proposed Korean to-
Tokenization | Token composition | Example
---|---|---
Syllable-based | LC + V + TC | / na/ neun/ / jip/ e/ / gan/ da/
Phoneme-based | LC, V, TC | / n/ a/ / eu/ n/ / j/ i/ p/ e/ / g/ / a/ n/ d/ a/
+ SkipTC | LC, V, {TC, SkipTC} | / n/ a/ / eu/ n/ / j/ i/ p/ e/ / g/ / a/ n/ d/ a/ -/ -/
Phoneme-based (ours) | LC + V, TC | / na/ neu/ n/ / ji/ p/ e/ / ga/ n/ da/
+ SkipTC | LC + V, {TC, SkipTC} | / na/ neun/ / jip/ e/ / gan/ da/

TABLE I: Comparison of different Korean tokenization methods. “ * ” symbol is the proposed SkipTC token, which is used to indicate the trailing consonant is missing in a syllable. For each example, the first row demonstrates the tokenized Korean sentence, and the second row shows the pronunciation of each token.

kernization method on the recently released large-scale 7,600h Korean speech dataset. Please note that we are the first to train an ASR model and report the performance on this dataset.

II. BACKGROUND AND RELATED WORK

A. Beam Search Decoding and RNN-LM

The goal of speech recognition is to find the most probable transcription for a given utterance. The ASR system transforms the input sequence of acoustic frames \( X = \{x_1, x_2, ..., x_T\} \) of length \( T \) to the output sequence of linguistic tokens \( Y = \{y_1, y_2, ..., y_L\} \) of length \( L \). Formally, the decoding process searches the most probable transcription \( Y^* \) that maximizes the score \( S(Y) \):

\[
Y^* = \text{argmax}_Y S(Y)
\]

\[
S(Y) = \log(p_{AM}(Y|X)) + \alpha \log(p_{LM}(Y)) + \beta |Y|,
\]

where \( |Y| \) indicates the number of tokens within the sequence and \( \alpha, \beta \) are weighting coefficients. \( \alpha \) is the LM weight that controls the contribution of LM, and \( \beta \) is the insertion bonus that encourages more words in the transcription, following [7].

Extensive studies have been conducted to improve the performance and efficiency of beam search decoding in a single-pass [8]. On the other hand, two-pass rescoring is also widely used [9]; usually, the first pass calculates the beam candidates with fast LM, such as N-gram LM, and then those candidates are re-evaluated by the relatively heavy RNN-LM.

B. Korean Syllable and Tokenization

We first introduce the basic structure of the Korean language. Korean alphabet consists of 14 consonants and 10 vowels that form a Korean syllable. A Korean syllable is divided into three parts: leading consonant (LC), vowel (V), and trailing consonant (TC) (see Figure 1). These three parts produce a unique sound, combined and pronounced together as a syllable. Each part may use a single consonant/vowel or combination of consonants/vowels, resulting in 19 leading consonants, 21 vowels, and 27 trailing consonants. In addition, silence can be used in place of LC and TC. When the silence comes at LC position, a consonant ‘\( \cdot \)’ is explicitly written. However, when the silence appears at TC, the position is left empty to indicate the silence. Therefore, it is natural to choose LC, V, and TC as minimum units for Korean speech.

Basically, four types of tokenization are commonly used for Korean [10]–[12]: (1) word-based, (2) morpheme-based, (3) syllable-based, and (4) consonant/vowel-based. For speech recognition, the former two tokenization methods often do not match actual sound boundaries. We also drop syllable-based tokenization because it requires a large token size; for example, the most common 2,306 syllables are often considered [4], which may be too large for CTC-based speech recognition. For the consonant/vowel-based approach, we can either consider each LC, V, TC as a single token or further combine these tokens. Based on the knowledge of native speakers, we decided to combine LC+V, because it can work as a single syllable independently. LC+V or LC+V+TC can become a single syllable but LC or TC itself cannot behave as a syllable. Therefore, we employ LC+V and TC separately as minimum units, for both ASR system and LM.

III. PROPOSED KOREAN TOKENIZATION

We propose SkipTC token for Korean tokenization to improve recognition performance. Specifically, we tokenize each Korean syllable into two parts, (1) LC+V and (2) TC. When TC is not included (skipped) in the syllable, we put an auxiliary token SkipTC (“\( * \)”) in the place. Table I compares various tokenization methods. Our tokenization always provides a sequence of altering token types of LC+V and TC, including SkipTC token. Although using SkipTC token increases the total sequence length, we claim that the difficulty of the probability density estimation can be decreased because of the regular input pattern (see Figure 2). Intuitively, RNN-LM can be trained to predict the next token only among the

\[19 \times 21 \times (27 + 1) = 11,172.\]
correct tokens that are in turn. We train RNN-LM using LSTM cells (thus, LSTM-LM) with the proposed tokenization, and then utilize the LM for the rescoring. We measure the quality of LM in two ways; by the perplexity of LM itself and by the speech recognition error rate.

The proposed tokenization method has three advantages. First, using SkipTC token helps RNN-LM to learn more regular patterns. Second, the method only requires a vocabulary of 427 tokens: 399 LC+V tokens, 27 TC tokens, and 1 SkipTC token. Third, our tokenization aligns well with the actual pronunciation which can be beneficial for both ASR and LM.

IV. EXPERIMENT

A. Dataset

We evaluate the proposed method on the Korean Free Conversation dataset [13], a recently introduced large-scale Korean speech corpus. Each utterance in the dataset is a recorded pronunciation of a colloquial sentence by native Korean speakers. Table II shows the statistics of the dataset. Because there is no specified test subset, we exploit the given valid dataset as a test dataset and split the training dataset (about 10%) as a separate valid dataset. For LM training, we employ the transcriptions from the same dataset as ASR; no external text source is used.

|        | #Utterances | #Syllables | Hours(h) |
|--------|-------------|------------|----------|
| Train  | 4,000,000   | 74,797,987 | 6,089.2  |
| Valid  | 421,770     | 7,878,410  | 640.9    |
| Test   | 557,564     | 10,116,858 | 871.4    |

We put a special effort into filtering and normalizing the data. First, we remove the non-linguistic event symbols (i.e., noise, laughter) and punctuation marks (i.e., period, exclamation mark, question mark, quotation mark) so that the sentence only contains syllables. Second, we filter out sentences consisting of less than 4 syllables. Third, we remove audio files that are too short, corrupted, or not recorded at 16kHz. After the cleaning, the average length and the number of syllables of each utterance are about 5.5 seconds and 18.6 syllables, respectively.

B. Model and Training Setup

We use Conformer-M [14] as the ASR encoder. The input frames are 80-dim log-Mel filterbanks extracted from 25ms window and 15ms overlap. We train the model with CTC loss using AdamW optimizer and peak learning rate of $1e^{-3}$. We follow additional training details from previous work that employed the same encoder [15].

LSTM-LM and Transformer-LM are a stack of LSTM and Transformer-XL layers, respectively. We train these neural network-based LMs with and without SkipTC, sharing the same training configuration. LMs consist of 4 layers where the hidden dimension of each layer is 512. Vocabulary size is set to 431, including 399 LC+V tokens, 27 TC tokens and 5 special tokens: <pad>, <eos>, <unk> and <space>. We train LSTM-LM using SGD with a momentum of 0.9, an initial learning rate of 0.1, weight decay of $1e^{-6}$, and exponential learning rate decay with a factor of 0.99. For Transformer-LM, we use Adam optimizer with a learning rate of 0.001.

C. LM Performance

To demonstrate the effectiveness of the proposed SkipTC, we train N-gram LMs, LSTM-LMs, and Transformer-LMs on the same training dataset. Table III compares the LM performance between different LM types. The result shows that utilizing SkipTC provides a considerable gain in performance for both LSTM-LM and Transformer-LM. Specifically, using SkipTC reduces the negative log-likelihood value by 21 and 26 on the valid and test datasets, respectively. N-gram models show the worst performance as expected. We observe that the difference in language modeling performance is marginal for LSTM-LM and Transformer-LM; therefore, we decide to employ LSTM-LM for the rescoring. The result empirically supports our speculation that patterning the input can be beneficial for neural network-based LMs in estimating the probability of a given sentence.

D. ASR Performance

Table IV shows the ASR performance using different LMs. We evaluate 4-gram, 6-gram LMs [16], and LSTM-LMs with and without SkipTC token. LSTM-LMs are employed for two-pass rescoring after the first pass decoding exploiting the 6-gram LM. We use a beam width of 128 for all cases. To search the best hyper-parameter $\alpha$ and $\beta$, we first perform a grid search of $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$ while keep $\beta = 0$. After

![Fig. 2: Illustration of the input sequence for RNN-LM tokenized with the proposed SkipTC token (“∗∗”). The sequence becomes regularly patterned when SkipTC token is employed.](image-url)
is selected, we fix $\alpha$ and search $\beta \in \{0, 0.1, 1.0, 2.0, 4.0\}$. We test hyper-parameter choices on the validation dataset and apply the best $\alpha, \beta$ for the test dataset.

Without the external LM decoding, the baseline ASR model achieves 9.27% WER and 2.19% SER on the test dataset. After the rescoring, LSTM-LM with SkipTC token reduces the WER to 6.59% and SER to 1.46%. We observe that utilizing SkipTC token shows better performance than the counterpart without SkipTC token; our method further lowers WER of validation and test dataset by 0.19% and 0.14%, respectively.

### E. Discussion and Future Work

There is a relatively big performance gap for WER than SER because many errors appear from the incorrect spacing, which is one of the most common errors in Korean ASR [4]. In addition, we observe that a large amount of palatalization, nasalization, and lateralization errors significantly disappear when SkipTC is employed. Note that these errors frequently appear for native Korean children (not used to writing), because dictating the utterance as-is would cause such mistakes, particularly related to understanding when to omit TC or not.

The core idea of this paper is to propose SkipTC token for the improved LM rescoring, but only (LC+V, TC) tokenization is utilized as a baseline. Since a standard method for Korean ASR has not yet been established, more research on tokenization should be conducted not only for LM but also for speech recognition systems.

## V. Conclusion

In this work, we proposed a Korean tokenization method for the RNN-LM used for ASR rescoring. We introduced a new auxiliary token, named SkipTC token, which is inserted when the trailing consonant is missing in a Korean syllable. We showed that using SkipTC token improves not only the LM performance but also the ASR performance. We evaluated LM and ASR performance on the recently released large-scale Korean Free Conversation dataset for the first time.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1013513). We gratefully acknowledge the GPU server support from the Artificial Intelligence Cluster Agency (AICA, Korea) in 2022.

## REFERENCES

[1] Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Paden Tomaselio, Jacob Kahn, Gilad Avidor, Ronan Collobert, and Gabriel Synnaeve, “Rethinking evaluation in asr: Are our models robust enough?,” Proc. Interspeech 2021, 2021.

[2] Anoop Deoras, Tomáš Mikolov, and Kenneth Church, “A fast rescoring strategy to capture long-distance dependencies,” in Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 1116–1127.

[3] Won Ik Cho, Sangwhan Moon, and Youngsook Song, “Open korean corpora: A practical report,” in Proceedings of Second Workshop for NLP Open Source Software (NLP-LOSS), 2020, pp. 85–93.

[4] Jeonguk Bang, Seung Yun, Seunghi Kim, Muyeol Choi, Minkyu Lee, Yeoejong Kim, Donghyun Kim, Jun Park, Youngjik Lee, and Sanghun Kim, “Kopenspeech: Korean spontaneous speech corpus for automatic speech recognition,” Applied Sciences, vol. 10, 2020.

[5] Hosung Park, Soonshin Seo, Daniel Jun Rim, Changmin Kim, Hyunsoo Son, Jeong-Sik Park, and Ji-Hwan Kim, “Korean grapheme unit-based speech recognition using attention-ctc ensemble network,” in 2019 International Symposium on Multimedia and Communication Technology (ISMAC). IEEE, 2019, pp. 1–5.

[6] Jeong-Uk Bang, Mu-Yeol Choi, Sang-Hun Kim, and Oh-Wook Kwon, “Extending an acoustic data-driven phone set for spontaneous speech recognition,” Proc. Interspeech 2019, pp. 4405–4409, 2019.

[7] Dario Amodei, Sundaram Ananthanarayan, Rishita Ambhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanazzo, Qiang Cheng, Guoliang Chen, et al., “Deep speech 2: End-to-end speech recognition in english and mandarin,” in International conference on machine learning, PMLR, 2016, pp. 173–182.

[8] Jennifer Drexler and James Glass, “Subword regularization and beam search decoding for end-to-end automatic speech recognition,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 6266–6270.

[9] Tara N Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mîrîkî Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, et al., “Two-pass end-to-end speech recognition,” Proc. Interspeech 2019, pp. 2773–2777, 2019.

[10] Ohwook Kwon, “Performance of lvcsr with morpheme-based and syllable-based recognition units,” in 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (Cat. No.00CH37100), 2000, vol. 3, pp. 1567–1570 vol.3.

[11] Kyubong Park, Jinho Lee, Seongbo Jang, and Dawoon Jung, “An empirical study of tokenization strategies for various korean nlp tasks,” 2020.

[12] Sunjoo Kim, Jeongmin Byun, Sion Baek, Yongseok Cho, and Alice Oh, “Subword-level word vector representations for korean,” in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 2429–2438.

[13] AI-Hub, “Ai-hub natural language dataset,” Oct 2021, Accessed on 22 October 2021.

[14] Anmol Gulati, James Qin, Ching-Cheng Chiu, Niki Parmar, Yu Zhang, Jiabai Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., “Conformer: Convolution-augmented transformer for speech recognition,” Proc. Interspeech 2020, pp. 5036–5040, 2020.

[15] Kyuhong Shim, Jungwook Choi, and Wonyong Sung, “Understanding the role of self attention for efficient speech recognition,” in International Conference on Learning Representations, 2022.

[16] Kenneth Heafield, “Kenlm: Faster and smaller language model queries,” in Proceedings of the sixth workshop on statistical machine translation, 2011, pp. 187–197.