Exploring Lexicon-Free Modeling Units for End-to-End Korean and Korean-English Code-Switching Speech Recognition

Jisung Wang, Jihwan Kim, Sangki Kim, Yeha Lee*

VUNO Inc., Seoul, Korea

jisung.wang@navercorp.com, jhkim@vuno.co, sangki.kim@vuno.co, yeha.lee@vuno.co

Abstract

Automatic speech recognition (ASR) tasks are usually solved using lexicon-based hybrid systems or character-based acoustic models to automatically translate speech data into written text. While hybrid systems require a manually designed lexicon, end-to-end models can process character-based speech data. This resolves the need to define a lexicon for non-English languages for which a standard lexicon may be absent. Korean is relatively phonetic and has a unique writing system, and it is thus worth investigating useful modeling units for end-to-end Korean ASR. Our work is the first to compare the performance of deep neural networks (DNNs), designed as a combination of connectionist temporal classification and attention-based encoder-decoder, on various lexicon-free Korean models. Experiments on the Zeroth-Korean dataset and medical records, which consist of Korean-only and Korean-English code-switching corpora respectively, show how DNNs based on syllables and sub-words significantly outperform Jamo-based models on Korean ASR tasks. Our successful application of using lexicon-free modeling units on non-English ASR tasks provides compelling evidence that lexicon-free approaches can alleviate the heavy code-switching involved in non-English medical transcriptions.

Index Terms: end-to-end speech recognition, modeling units, attention, connectionist temporal classification

1. Introduction

Sequence-to-sequence (seq2seq) learning using attention-based models has drawn increasing attention for tasks involving sequential data such as automatic speech recognition (ASR) [1, 2, 3]. End-to-end approaches to ASR, in contrast to hybrid systems, directly predict character-based units, mitigating the need to manually design a lexicon. Various acoustic units have been used as end-to-end seq2seq models on English ASR tasks, including graphemes [1] which are lexicon-free units, word-pieces [4], and sentence-pieces [5], as well as lexicon-related units spanning context-dependent states and context-independent phonemes [6]. It is evident from previous models that modeling units critically impact seq2seq models' performances.

The Korean language does not have a standard phoneme set nor lexicon in contrast to the English language with its CMU dictionary. Hence, a character-based end-to-end model would be attractive not only for Korean ASR tasks, but also any language without a standard phoneme set or lexicon. The Korean writing system consists of letters named Jamo, which can either be a consonant or vowel. These Jamo letters form a syllable block, a basic Korean character. A lexicon-free modeling unit can thus be made up of either a Jamo or syllable-based Korean character.

While modeling units have been devised for seq2seq learning [2, 6, 7, 8] for Mandarin Chinese [9, 10], there has yet to be a study on modeling units fit for Korean ASR. In this work, we introduce several modeling units applicable to Korean ASR and compare the performance of a deep neural network (DNN) on a standard Korean ASR benchmark and Korean-English code-switching task when using these different units. In particular, we experiment with Jamo, syllable, Jamo based sub-word, syllable based sub-word, and byte [7] models, where sub-words are generated using SentencePiece [11]. Our method alleviates the need to design a common phoneme set, lexicon, or language model in developing a code-switching ASR system and can significantly reduced the cost associated with designing such models. Experiments using a combination of connectionist temporal classification (CTC) and attention based decoder as the base DNN architecture suggest that a syllable-based sub-word model is ideal for Korean ASR, and that a combination of syllable-based sub-word unit and English sub-word unit is best for Korean-English code-switching tasks.

2. System Overview

2.1. Model Architecture

Our DNN architecture is mainly inspired by the Listen, Attend, and Spell (LAS) model [3] and was modified to better suit our task. As shown in Figure 1, our model is a sequence of recurrent neural network (RNN) based encoder, attention, and an RNN-based decoder. The encoder is modified by replacing sub-sampling layers in LAS with max-pooling operations which down-sample input signals across both time and frequency axes. Its RNN part is a 5-layer bi-directional long short term memory (BLSTM) [12] module with 512 cells followed by linear projection. The attention module is a 512 dimensional location-aware mechanism [13] with 10 convolutional channels and filter size of 100. It takes as input both the encoded signal and the previous prediction to incorporate the sequence history. The decoder is a 2-layer LSTM with 512 cells.

We also adopted a joint decoding method which takes the CTC predictions into account during inference [14]. In order to combine frame-synchronous CTC probabilities with label-synchronous attention probabilities, we followed the one-pass decoding method described in [14].

2.2. Optimization

DNNs were trained to maximize a convex combination of CTC criterion [15] and attention. Given a L-length character sequence $c \in \mathcal{C}^L$ with characters $c_i \in \mathcal{C}$, CTC is the log-likelihood of the sequence prediction on a speech input $x \in X^T$. Since a character sequence may be shorter than the speech input, a 'blank' symbol b is inserted before, in between, or at the end of character sequences to obtain the output space

[*] Currently working at NAVER Corp.

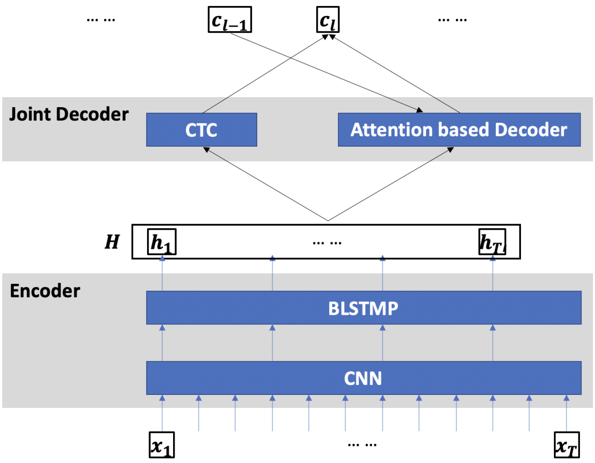

Figure 1: *Joint CTC/Attention ASR.*

$\mathcal{Y} = (\mathcal{C} \cup \{b\})^T$. The resulting CTC loss is the negative log-likelihood of a character sequence

$$\mathcal{L}_{CTC} = -\log\left(\sum_{y}\prod_{t} p\left(y_t|x\right)\right), \tag{1}$$

assuming independence among characters $y_1, \ldots, y_T$, where $p(y_t|x)$ is the DNN's prediction for character y_t.

An attention loss accounts for the character sequence's structure by conditioning each character prediction on its history:

$$\mathcal{L}_{att} = -\log\left(\prod_{l} p\left(c_l|c_1, \ldots, c_{l-1}, h\right)\right), \tag{2}$$

where $p\left(c_l|c_1, \ldots, c_{l-1}, h\right) = \mathcal{A}\left(h, c_{l-1}\right)$ is the attention-based decoder's output on the encoder's output h and previous character c_{l-1}. The resulting objective function is their convex combination

$$\mathcal{L}_{tot} = \lambda\mathcal{L}_{CTC} + (1-\lambda)\,\mathcal{L}_{att}, \lambda \in [0, 1]. \tag{3}$$

2.3. Modeling Units

2.3.1. Korean Alphabet

Hangul is the Korean writing system which consists of 51 Jamo letters comprising 30 consonants and 21 vowels. A syllable block is built as a combination of several Jamo letters and spans two dimensions. This is illustrated in Fig. 2 where in the first example, three Jamo letters, ㄱ, ㅏ, and ㅂ are combined clockwise to form a syllabic block 갑. A syllable block cannot be a single Jamo letter and is necessarily a group of 2 or three Jamo letters. The first Jamo letter of a group is called choseong followed by jungseong. Optionally, a third Jamo letter called jongseong may be present in a syllable block.

Not all Jamo letters can be used as every element of a syllable block: only 19 Jamo letters can be used as a choseong, 21 vowels for jungseong, and 28 consonants (including none) for jongseong. While this results in $11,172 = 19 \times 21 \times 28$ possible combinations, a small subset is used in the Korean language. Thus, we only consider $2,350$ most frequently used syllables.

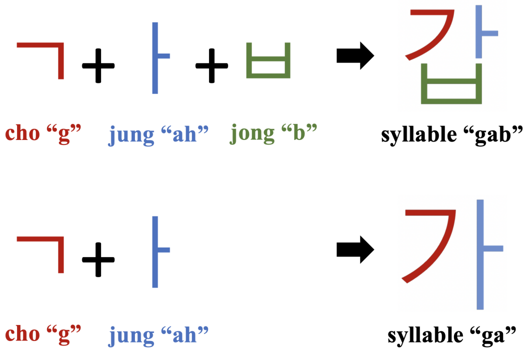

Figure 2: *Illustration of grouping Jamo letters: (left) three and two Jamo letters are grouped forming a (right) syllable block.*

2.3.2. Sub-word Units

Sub-word units are generated by applying SentencePiece [11] on the aforementioned units: syllable-based Korean character (right in Fig. 2) and Jamo characters (left) obtained by decomposing the syllable blocks into cho, jung, and jongseong. Jamo-based sub-word units include Jamo characters, (partial) syllable blocks, and an entire word. Syllable-based sub-word units range from a syllable block to the entire word. The number of sub-word units in the SentencePiece model is a user-controllable hyperparameter.

Table 1: *Examples of various units in a sentence, "학교에 간다 (I'm going to school)". One of its code-switching versions is "school 에 간다". Token <sp> refers to 'space'. 'sw' stands for the sub-word unit. A '/' is used to distinguish Korean/English letters.*

Units	Examples
syllable	학, 교, 에, <sp>, 간, 다
jamo	ㅎ, ㅏ, ㄱ, ㄱ, ㅛ, ㅇ, ㅔ, <sp>, ㄱ, ㅏ, ㄴ, ㄷ, ㅏ
syll-sw	학교, 에_, 간, 다
jamo-sw	학ㄱ,ㅛ, ㅇ, ㅔ_,가,ㄴ,다
(Eng.) char	I,',m,<sp>,g,o,i,n,g,<sp>,t,o,<sp>,s,c,h,o,o,l
(Eng.) sw	I'm, _go, ing, _to_, s, ch, ool
Jamo/char	s,c,h,o,o,l, ㅇ, ㅔ,<sp>, ㄱ, ㅏ, ㄴ, ㄷ, ㅏ
syll/char	s,c,h,o,o,l, 에, <sp>, 간, 다
syll-sw/sw	s, ch, ool_, 에_, 간, 다
Jamo-sw/sw	s, ch, ool_, ㅇ, ㅔ_,가,ㄴ,다

Table 1 shows a sample sentence tokenized in different units: syllable, Jamo, syllable-based sub-word, Jamo-based sub-word unit. When using the SentencePiece model, <sp> is replaced with an under-bar token which may or may not be joined by another character. For the Korean-English code-switching dataset, hybrid units (bottom of Table 1) with English characters and sub-word units were used.

3. Experiments

3.1. Datasets

The lexicon-free acoustic models were evaluated on two different ASR corpora: Zeroth-Korean [16], developed by a Korean ASR open source project known as Zeroth [17] based on Kaldi [18], and Medical Record (MedRec) which consists of a large amount of real medical records obtained from Korean hospitals. Zeroth-Korean contains a morpheme-based segmenter called morfessor [19] and transcribed audio data. This morphologically segmented text was used in our experiments. MedRec was used to evaluate the models on Korean-English code-switching ASR. Text data in MedRec was morphologically segmented using an in-house rule-based tool.

Table 2: *Statistics for Zeroth-Korean (16 kHz, 16 bit) and Medical Record (8 kHz, 8 bit) corpora. Language (Lang.) is the percentage Korean/English alphabets occupy. Column 'Single (s)' is the average duration of wave files in seconds.*

Zeroth	Lang. (%)	Total (h)	Single (s)	# Spkrs
Train	100/0	51.6	8 (3 ∼ 20)	105
Test	100/0	1.19	9 (5 ∼ 20)	10
MedRec	**Lang. (%)**	**Total (h)**	**Single (s)**	**# Spkrs**
Train	40.4 / 51.1	2530	17 (2 ∼ 58)	160
Test	41.6 / 49.2	1.16	25 (2 ∼ 59)	10

Eighty-dimensional log-mel filterbank coefficients were extracted from 3-dimensional pitch values using the method described in [20] with 10ms intervals and a 25ms Hamming window. These features were normalized using pre-computed means and standard deviations computed from each training set.

3.2. Implementation

The number of target units used for each modeling unit is shown in Table 3. A syllable unit (first row) contains 2,350 syllable characters with additional <sp>, <unk>, and <symb> tokens. Since an unknown token is absent in the Jamo-based system, only <sp> and <symb> were added to 68 Jamo characters for the Jamo unit model. No extra labels were added for the byte unit [7] model. Sub-word units were generated using SentencePiece with 3k and 6k target sets on a syllable-based text, and 2k and 3k targets on Jamo-based text. Two common tokens, <blk> for CTC and <sos/eos> indicating the beginning/end of a sentence, were added for all modeling units including sub-word sets for the attention-based decoder. For the Korean-English code-switching ASR task, an apostrophe token was included for a total of 27 classes. These additional classes resulted in a model having 39M to 49M parameters, depending on the number of target labels.

Table 3: *Number of output classes for each modeling unit. <symb>includes spoken symbols such as #, %, &, and numbers 0 ∼ 9, 10, 100 etc., for a total of 19 classes.*

Units	# outputs	Labels
syllable	2371	syll2350 + <sp>+ <unk>+ <symb>
Jamo	88	jamo68 + <sp>+ <symb>
byte	256	00 ∼ ff

The convex combination parameter λ in Eq. (3) was set as 0.2. Implementations were done using the ESPnet toolkit [21] and Chainer CTC. Unigram label smoothing [1] was employed to help training, and the models were optimized using Adadelta [22] with gradient clipping. Beam search was used for inference, with a beam width set as 30.

4. Results and discussion

Tables 4 and 5 show the unit (UER), word (WER), and sentence error rate (SER) of the joint CTC/Attention model for Zeroth-Korean and MedRec, respectively, using 5 different modeling units: syllable, jamo, syllable-based sub-word, jamo-based sub-word, and byte.

Table 4: *UER, WER, SER (%) of joint CTC/Attention model on the Zeroth-Korean test set using different modeling units.*

Units	UER (%)	WER (%)	SER (%)
syllable	**1.8**	2.6	**3.3**
jamo	6.3	19.9	83.6
syll-subword (3k)	3.0	3.2	5.0
syll-subword (6k)	2.3	**2.5**	**3.3**
jamo-subword (2k)	*75.3*	4.1	4.8
jamo-subword (3k)	4.2	4.3	4.4
byte	2.5	4.4	13.1

The syllable-based sub-word unit model achieved the best average error rate, and the model using Jamo unit had the highest WER. This is reasonable, as syllable-based units contain the largest group of Jamo letters, and can thus utilize long-term language dependencies. SentencePiece combines Jamo letters, enhancing the model's performance as shown in the second, fifth, and sixth rows. These results collectively imply that units containing larger groups are more beneficial than units with smaller groups such as Jamo letters.

Table 5: *UER, WER, SER (%) on test set of Medical Record when modeling with different modeling units.*

Units	UER (%)	WER (%)	SER (%)
(ko) syllable + (en) char	**4.6**	**8.1**	66.1
(ko) jamo + (en) char	6.7	16.8	92.3
syll-subword (3k)	9.7	8.2	65.5
syll-subword (6k)	7.7	**6.9**	**64.2**
byte	6.4	10.4	77.0

While using Jamo-based sub-word units yields an extremely high UER value of 75.3%, the model achieves a low WER value of 4.1% on the Zeroth dataset. This phenomenon is exemplified in Fig. 3, where a syllable-based model requires learning a fewer combinations than that needed for a Jamo-based system by using larger groups. Consequently, the syllable-based model achieves significantly lower UER values as shown in the fifth and sixth rows in Table 4.

Although Jamo is similar to English letters in that it is either a consonant or vowel, a syllable-based Korean character with an English alphabet outperformed the Jamo and English alphabet model on the code-switching task (Table 5). This suggests that the performance of ASR models is not solely dependent on the size of letter groups, and that other factors may affect the model's performance more, especially on code-switching tasks. On the other hand, a combination of syllable-based sub-word unit and English sub-word unit proved the most effective on this task, which was the case for Korean ASR. Another interesting observation is that byte unit [7], which can be used universally across any language, achieved lower WER than Jamo on both tasks.

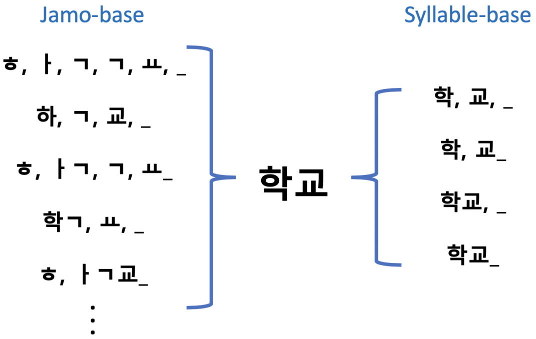

Figure 3: *A sample word '학교', meaning 'school' in English. Word '학교' can be encoded into various Jamo-based sub-word units as shown on the left. Only 4 cases are possible for syllable-based sub-word units on the right.*

5. Conclusion and future work

Lexicon-free modeling units for Korean and Korean-English code-switching ASR tasks were investigated in this study. We systematically compared the performance of a CTC/attention-based seq2seq model using syllable, Jamo, and syllable/Jamo-based sub-word units, and showed how sub-word unit based on Korean syllables performed the best, which is consistent with existing code-switching algorithms. Our results suggest that acoustic modeling of non-English ASR tasks could significantly benefit from using various lexicon-free approaches. It would be interesting to see whether our observations stand true when using a Transformer [23] instead of the CTC/attention-based network.

6. Acknowledgements

We thank Seo Taek Kong for comments that greatly improved the manuscript and reviewing English grammar.

7. References

[1] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," *ICASSP*, pp. 4774–4778, 2018.

[2] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," *Interspeech*, pp. 939–943, 2017.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *ICASSP*, pp. 4960–4964, 2012.

[4] M. Schuster and K. Nakajima, "Japanese and korean voice search," *ICASSP*, pp. 5149–5152, 2012.

[5] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, and X. W. S. W. T. Y. W. Z. N. E. Y. Soplin, R. Yamamoto, "A comparative study on transformer vs rnn in speech recognition," *arXiv:1909.06317v2*, 2019.

[6] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of attention in sequence-to-sequence models," *Interspeech*, pp. 3702–3706, 2017.

[7] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," *ICASSP*, pp. 5621–5625, 2019.

[8] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," *Interspeech*, pp. 3800–3804, 2019.

[9] S. Zhou, L. Dong, S. Xu, and B. Xu, "A comprison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese," *arXiv:1805.06239v2*, 2018.

[10] W. Zou, D. Jiang, S. Zhao, and X. Li, "A comparable study of modeling units for end-to-end mandarin speech recognition," *arXiv:1805.03832v2*, 2018.

[11] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 66–71, 2018.

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

[13] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 577–585, 2015.

[14] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," *arXiv:1706.02737v1*, 2017.

[15] A. Graves, S. Fern´andez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," *ICML*, pp. 369–376, 2006.

[16] "http://www.openslr.org/40/," *Zeroth Korean.*

[17] L. Jo and W. Lee, "https://github.com/goodatlas/zeroth," *Zeroth Project.*

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[19] S. Virpioja, P. Smit, S. Grönroos, and M. Kurimo, "Morfessor 2.0: Python implementation and extensions for morfessor baseline," *School of Electrical Engineering*, 2013.

[20] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," *ICASSP*, pp. 2494—-2498, 2014.

[21] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," *Interspeech*, pp. 2207–2211, 2018.

[22] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv:1212.5701v1*, 2012.

[23] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model," *ICASSP*, pp. 5884–5888, 2018.