Hybrid CTC-Attention Network-Based End-to-End Speech Recognition System for Korean Language

Hosung Park[1], Changmin Kim[2], Hyunsoo Son[1], Soonshin Seo[3] and Ji-Hwan Kim[1,*]

[1]*Sogang University, Seoul, South Korea*
[2]*LG Electronics, Seoul, Korea*
[3]*Naver Corporation, Gyeonggi Province, South Korea*
E-mail: hosungpark@sogang.ac.kr; changmin0.kim@lge.com; sonhyunsoo@sogang.ac.kr; sunshin.seo@navercorp.com; kimjihwan@sogang.ac.kr
[*]*Corresponding Author*

Received 13 August 2021; Accepted 18 August 2021;
Publication 03 January 2022

Abstract

In this study, an automatic end-to-end speech recognition system based on hybrid CTC-attention network for Korean language is proposed. Deep neural network/hidden Markov model (DNN/HMM)-based speech recognition system has driven dramatic improvement in this area. However, it is difficult for non-experts to develop speech recognition for new applications. End-to-end approaches have simplified speech recognition system into a single-network architecture. These approaches can develop speech recognition system that does not require expert knowledge. In this paper, we propose hybrid CTC-attention network as end-to-end speech recognition model for Korean language. This model effectively utilizes a CTC objective function during attention model training. This approach improves the performance in terms of speech recognition accuracy as well as training speed. In most

Journal of Web Engineering, Vol. 21_2, 265–284.
doi: 10.13052/jwe1540-9589.2126

languages, end-to-end speech recognition uses characters as output labels. However, for Korean, character-based end-to-end speech recognition is not an efficient approach because Korean language has 11,172 possible numbers of characters. The number is relatively large compared to other languages. For example, English has 26 characters, and Japanese has 50 characters. To address this problem, we utilize Korean 49 graphemes as output labels. Experimental result shows 10.02% character error rate (CER) when 740 hours of Korean training data are used.

Keywords: End-to-End speech recognition, hybrid CTC-attention network, Korean speech recognition.

1 Introduction

An automatic speech recognition (ASR) system is factorized into several subtasks including acoustic, lexicon, and language model. Especially, deep neural network/hidden Markov model (DNN/HMM)-based approach has driven substantial improvements in acoustic model [1]. In this system, the acoustic model, language model, and lexicon model are developed separately and integrated into a weighted finite state transducer (WFST) for efficient decoding. In this integration, it is difficult for non-experts to develop ASR system for new applications and languages.

The goal of end-to-end approach has simplified the above system into a single-network architecture within a deep learning framework in order to address the above difficulty. This approach successfully leads to more optimised ASR system compared with DNN/HMM-based approach [2].

There are two types of end-to-end ASR architecture: connectionist temporal classification (CTC) and encoder–decoder network with attention mechanism. CTC model is a sequence training model that monotonically maps an input sequence to an output sequence of shorter length [3]. CTC uses forward–backward algorithm to solve sequential problems efficiently. The key difference of CTC compared to DNN–HMM architecture is that the output labels can be phonemes or characters directly instead of HMM states.

Encoder–decoder network with attention mechanism is also known as attention model [4]. The attention model has been applied in machine translation [4], image captioning [5], and speech recognition [6, 7]. This model learns stacked recurrent layers from audio features sequence to output label directly. The attention model transforms input sequence of variable length into fixed dimensional vector using the encoder, and recovers output labels

from the fixed dimensional vector using the decoder. However, the attention model allows non-sequential alignments, although the alignments in speech recognition are monotonic.

To solve this problem, a hybrid CTC-attention model is proposed in [8]. The hybrid CTC-attention model effectively utilizes a CTC objective function during attention model training. CTC in this hybrid model encourages alignments to be monotonic. This hybrid model is trained by objective functions of CTC and attention simultaneously. This approach improves the performance in terms of recognition accuracy as well as training speed [8].

The parameters of end-to-end model extremely depend on language characteristics because the number of characters differs from language to language. For example, end-to-end speech recognition of English has 26 output labels representing each English character [9] and that of Japanese has 50 output labels representing each Japanese Kana symbol [10]. English and Japanese end-to-end speech recognition systems achieved significantly high recognition accuracy, thereby reducing the difference with the hybrid DNN–HMM architecture. However, end-to-end speech recognition of Chinese has approximately 5,000 output labels of character symbols [11]. This leads to a two-fold error rate in terms of character error rate compared with DNN–HMM-based methods [11].

There are 11,172 character symbols in Korean end-to-end speech recognition. This is an inefficient method in end-to-end speech recognition since the model has many parameters to predict compared to other languages. There were previous studies on the straightforward way, which uses all 11,172 characters, but the results were poor compared to the existing studies [9, 12].

This performance degradation problem is caused by the linguistic characteristic of Korean language. Korean syllables are made up of a combination of three types of graphemes, namely, the Cho-sung (initial consonant), Jung-sung (vowel), and Jong-sung (final consonant). There are 19 Cho-sungs, 21 Jung-sungs, and 28 Jong-sungs, while the Jong-sungs use all the 19 graphemes of the Cho-sungs. Because of this, the Korean language contains 11,172 syllables that consist of two or three graphemes [13, 14]. For example, in English, the word "car" is represented by the three graphemes 'c,' 'a,' and 'r.' Each syllable has one grapheme. Korean is a syllabic language type, where the word "소금" is represented by five graphemes, namely, 'ㅅ,' 'ㅗ,' 'ㄱ,' 'ㅡ,' and 'ㅁ,' despite the fact that it consists of only two characters, namely '소,' and '금.'

In this paper, we propose an efficient Korean speech recognition system that has only 49 individual graphemes, compared with the technology that

needs 11,172 kinds of syllables. To make this, we use a hybrid CTC-attention model as Korean end-to-end speech recognition system that uses graphemes as recognition units. We applied Korean graphemes as output unit to this hybrid model and showed a performance improvement over the Korean syllable-based end-to-end model.

This paper is organized as follows: Section 2 presents related studies about end-to-end structure and recognition unit. Section 3 describes the hybrid CTC-attention-based, end-to-end speech recognition architecture. Section 4 describes the method used to compose and decompose Korean language between syllables and graphemes. Section 5 explains the experiments. Section 6 concludes the study.

2 Related Studies

The end-to-end speech recognition system categorizes three different methods. The first method uses CTC [3, 11, 15]. CTC is proposed as an approach to extract phoneme sequences from speech features without forced alignment. This comprises a necessary method in the DNN–HMM model. CTC is a kind of loss function as a neural network's output layer. CTC does not require forced alignment because CTC extracts phoneme sequences from the speech frames. The HMM-based model's auto segmentation method includes identifying the best route according to the time sequence of phonemes. However, CTC results in a lower performance because each frame of the speech features cannot present the entire quality of the phone [16].

The second method is the attention model. This model is a sequence-to-sequence model that represents inputs/outputs as data series [4, 17, 18]. The attention model is a kind of encoder and decoder model, and is one of the most famous kinds of sequence-to-sequence models. Given that this model type requires the learning of information, it uses the recursive neural network (RNN) model [17]. In attention network, the energy weights based on the output vector generated by the encoder during the decoding and joins results of two networks. Its system depends on the consideration of the output value, which is significant among the outputs of the encoder. Compared with other end-to-end models, the model performance is measured similar to the DNN–HMM in a specific domain. However, the model is not utilized in streaming. the sequences must be used into a one network in case of this model [19].

The RNN transducer is the third method [19, 20]. This method uses both sequence training and speech-to-word transform at the same time with

two RNNs. The RNN transducer comprises of a transcription network that transforms speech features to word sequences and a prediction network learned by the language sequences. According to the number of input frames, the number of input nodes in the transcription network using RNN generates transcription vector sequences. The prediction RNN has the maximum number of input labels as input nodes. This model generates the same number of vector sequences. The output vectors of two RNNs trained at the same time. The two RNNs' output vectors trained at the same time. In this model, fewer computations are required during the decoding compared to CTC, which considers all the possible outcomes. This achieves a precise alignment based on training data. However, a large amount of training data and time is required for computing and the complicated network and limited input size make it unsuitable for sentence unit-based speech recognition [19]. Among the three methods, the CTC method is the most efficient convergence when the CTC model is learnt with phoneme-unit outputs [16]. In comparison to the attention model, the CTC model is simpler as it only computes frame unit outputs, and a relatively smaller number of training data is required for convergence. Additionally, it shows superior compatibility, especially the ability to use weighted finite-state transducer (WFST) decoding like the DNN–HMM-based model. The method that yields the best performance is the attention network–based method. This model, which uses the entire sequence, has the highest accuracy among the end-to-end systems. Recently, a bidirectional long short-term memory (LSTM) was used for bidirectional training on a sequence. This resulted in an improved accuracy. Additionally, a hybrid model that combined the CTC and the attention network was also been proposed. It combined the advantages of each model and exhibited better performance [16].

3 Hybrid CTC-Attention Architecture

This section describes the hybrid CTC-attention network for the Korean end-to-end model in speech recognition. Figure 1 illustrates the structure of the network. The input feature sequence is denoted by x, and its transcription by y. The frame length of each element of x has 20 ms (milliseconds). However, the output sequence length is typically shorter than that of x. In encoder–decoder model, the encoder is shared to CTC and attention. The encoder output vector h is used to create the attention weight vector a and the context vector c. The encoder uses a bidirectional LSTM network structure [8].

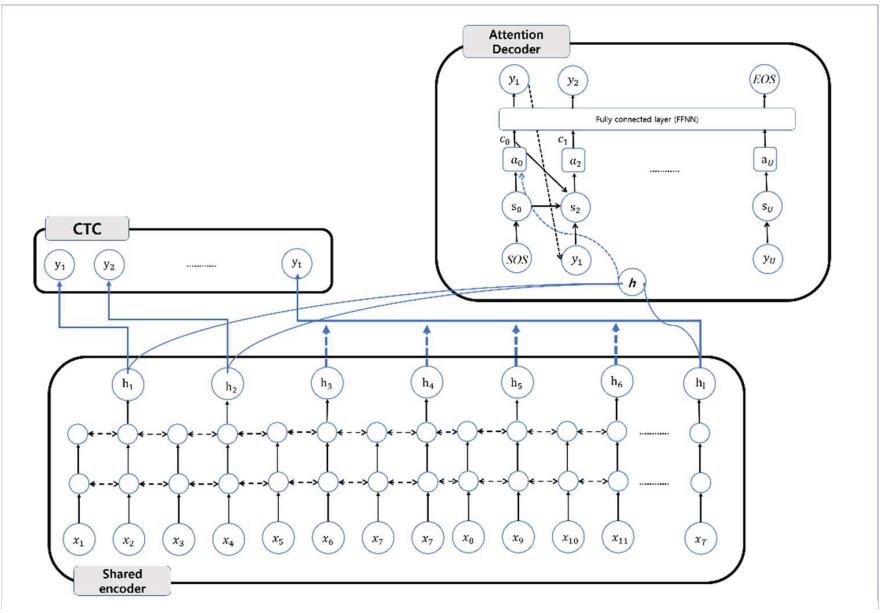

Figure 1 Hybrid connectionist temporal classification (CTC) and attention-based speech recognition network: the input x is transformed the encoder vector h as the output by shared encoder. The shared encoder is trained both the CTC model and the attention model. The CTC and the attention generate the probability distribution for the output y.

3.1 Shared Encoder

In speech recognition problems, the number of inputs is much larger than the number of outputs. A speech signal is represented by multiple vector sequences with the use of feature extractors, such as perception linear prediction (PLP) and Mel-frequency cepstral coefficients (MFCCs). Each vector sequence normally has a length, which spans 20–25 ms approximately. For example, a phoneme [*t*] has 16–25 features on average. The encoder transforms the speech feature vector *x* to the encoder vector *h* to compress the input vector. The encoder vector *h* is used as the input based on the decoder's CTC criterion known as "Attention decoders." In this study, we used bidirectional LSTM–RNN as the encoder network [18]. A LSTM–RNN network is used to solve sequential problems in deep learning tasks. Because the network refers to many more inputs to generate encoder vectors efficiently, this recurrent network not only refers to the next node weight but also to the previous. Figure 2 shows a bidirectional LSTM. In this case, x_1, x_2, and x_t are inputs from the speech signal, h_1, h_2, and h_t are from forward

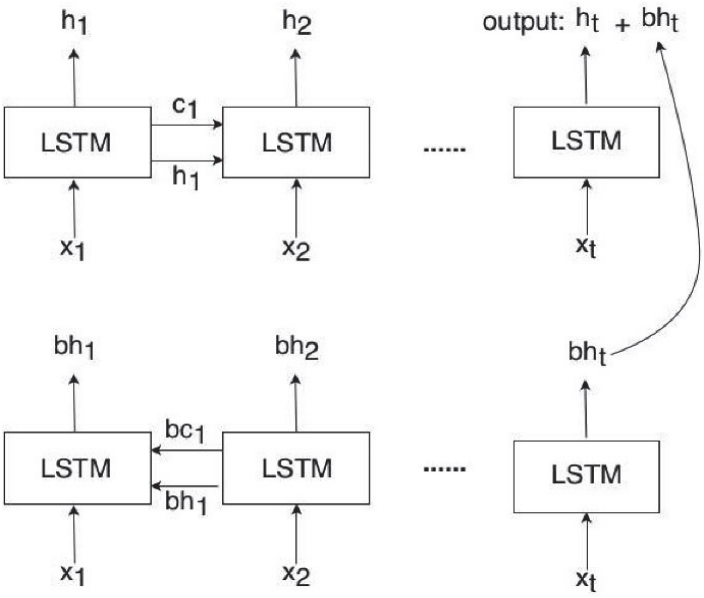

Figure 2 Bidirectional long short-term memory (LSTM) architecture proposed in this study.

LSTM layers, and bh_1, bh_2, and bh_t are from backward LSTM layers, while $c_1, c_2, c_t, bc_1, bc_2, bc_t$ are from memory cell values in each LSTM layer.

3.2 Connectionist Temporal Classification (CTC)

In CTC, an output is produced for each input *x*. In Korean, a number of frames generate just one pronunciation sequence. To handle this, blank symbols are used to determine the boundary of the output in CTC. CTC is applied to target the encoder output vector *h*, which is the encoder network's output [6].

$$\mathrm{y}^* = argmax_y P(y|h) \tag{1}$$

(1) is the CTC basic formula. y^* is the closest to the *y* for the encoder output *h*. P(*y*|*h*) is illustrated as indicated in (2).

$$\mathrm{P(y|h)} \approx \prod_{t=1}^{T} P(y_t|x) = \prod_{t=1}^{T} q_t(y_t) \tag{2}$$

CTC is a loss function on the network's output layer and find for an output sequence using (2). $\mathrm{q_t}(y_t)$ is the softmax activation value of y_t of the encoder

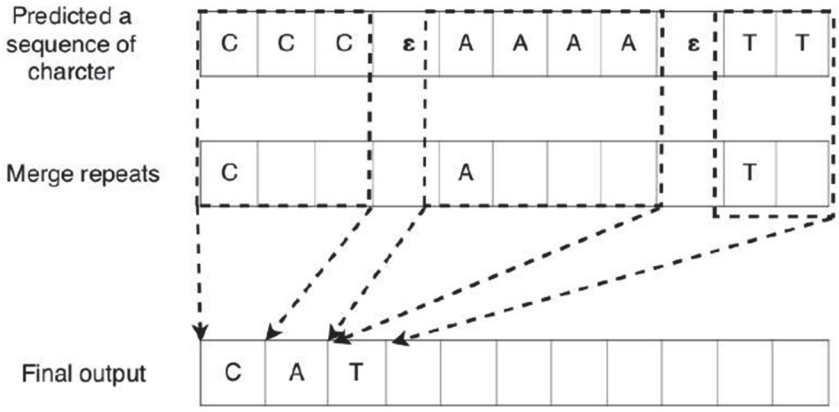

Figure 3 CTC networks for speech recognition used in this study. In this case, e represents a blank symbol.

layer *q* at time *t*. The CTC loss function is shown in formula (3).

$$\mathrm{L_{CTC}} = -\mathrm{lnP(y|x)} = -ln \sum_{t=0}^{T} (\alpha_t \beta_t)/q_t \tag{3}$$

The loss function of CTC is calculated with the forward and backward algorithm. The forward value α represents the sum of all routes from 0 to *t* for CTC training. The backward value β represents the sum of the ways from time *T* to *t* when the forward value exists. Figure 3 represents the CTC algorithm. The first stage is the prediction step. The deep neural network (ex $>$ feed-forward neural network (FFNN), RNN) outputs a sequence of syllable or a phoneme. The second step is the alignment step. A blank symbol is dropped and a repeated character is merged to ensure alignment. Finally, the merged character or phoneme is assigned as the output.

3.3 Attention Decoder

The RNN–LSTM-based attention decoder takes the shared encoder output *h* and transforms it into the attention weight from the previous time in decoder to train the model to maximize the output parameter.

Figure 4 shows attention decoder architecture. In this case, $h_1 \ldots\ h_5$ are the encoder vectors given by the shared encoder. Additionally, c_1, c_2, and c_3 represent context vectors as the decoder's input. The context vectors guide the decoder to identify the encoder vectors that deserve to pay attention to solve

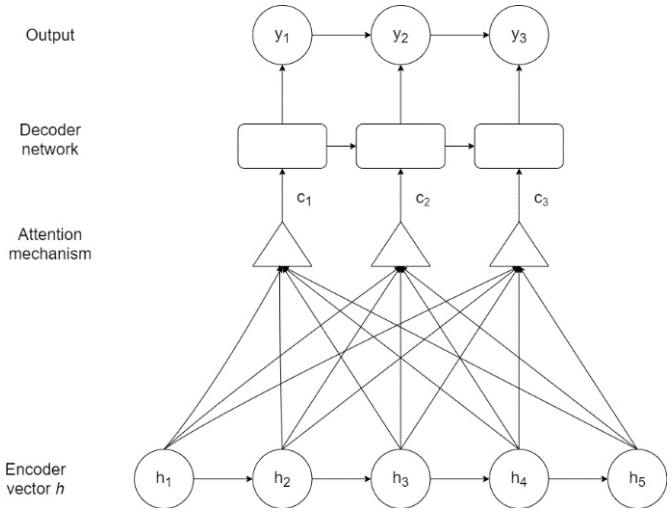

Figure 4 Decoder network with an attention mechanism as used in this study.

the problem. Additionally, *y* is a sequence of characters used as the output.

$$P(y|h) = \prod_{u=1}^{T} P(y_u|h,\ y_{1:u-1}) \tag{4}$$

(4) represents the attention-based decoder. The output is the multiplication of the probabilities for the encoder output *h* and decoder output $y_{1:\mathrm{u}-1}$ of the previous point for time step *u*.

$$C_u = \sum_{l} (a_{u,l} h_l) \tag{5}$$

For *u*th step in (4), decoder model generates a context vector C_{u}, as in (5). $a_{\mathrm{u,l}}$ represents the attention weight vector at each time. This study uses the location attention method to compute attention weight. The location algorithm computes the attention weight using the RNN decoder weight, the encoder output, and the previous time point [18].

$$a_{u,l} = softmax(e_{u,l}) \tag{6}$$

$$e_{u,l} = \omega^T \tanh(W s_{u-1} + V h_l + M f_{u,l} + b) \tag{7}$$

$$T f_u = F * a_{u-1} \tag{8}$$

(6), (7), and (8) represent the attention weight's computation. The attention energy is represented by e. As shown in (7), it is computed with f_{u}, the outcome of the one-dimensional convolutional parameter F on the value obtained from the previous time's attention weight. The trainable weight matrices are *W, V, M, b*, and ω. *W, V,* and *M,* and *b* denotes the bias.

$$s_u = LSTM(s_{u-1}, y_{u-1}, c_u) \tag{9}$$

$$y_u = FFNN(s_u, c_u) \tag{10}$$

(9) receives as inputs $y_{\mathrm{u}-1}$ the decoder's output at the u–1 time, the context vectors c_{u} and $s_{\mathrm{u}-1}$, and the LSTM state at the u–1 time to generate c_{u}. The c_{u} is fed as inputs to generate the output y_{u}. In attention decoder, the beginning of the sentence is defined as a start of sequence (SOS) and the end of a sentence is defined as an end-of-sequence (EOS) symbol [16].

$$\mathrm{Loss} = \lambda \mathrm{Loss}_{\mathrm{ctc}} + (1-\lambda)\mathrm{Loss}_{\mathrm{attention}}. \tag{11}$$

Equation (11) defines the loss function in the hybrid model, and λ is a non-trainable parameter value in the range between 0 and 1.

Figure 5 shows the hybrid CTC-attention architecture, whereby x_{t} represents a sequence of input speech features, and h_{t} is the encoder vector from the shared encoder. The final output y_{t} is the weighted sum of the output probabilities from CTC and attention.

3.4 Hybrid Model

The CTC-attention hybrid network is proposed to handle the weaknesses of the two models and improve their accuracy [8]. The CTC and attention loss functions reduce errors from network training. In speech recognition, a language model is required to improve the performance. CTC cannot read the left and right context of input frame given that the CTC results for each speech frame. The attention network, which learns the left and right contexts as input, solves this problem in CTC. Moreover, the attention model does not require an alignment process because the attention model is trained on an entire speech signal at the same time. Therefore, the attention model is trained without information of alignment for input and output sequences. This lowers the accuracy of the speech recognition results. The problem would be solved by using the CTC, which generates an output for speech input [16].

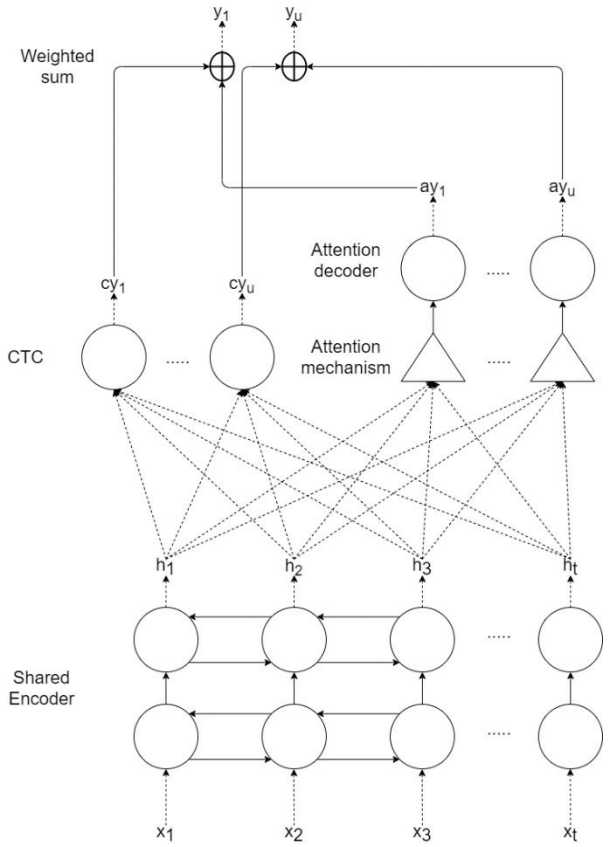

Figure 5 Hybrid CTC-attention architecture.

4 Korean Graphemes

There are graphemes, phonemes, morphemes, syllables, and words in Korean recognition units. The end-to-end speech recognition models generate the recognition units' probability as output. The recognition units can change the number of parameters to update. Syllable-based units are the most widely used ones in DNN–HMM-based acoustic model. During decoding process, the model converts phonemes as inputs into word, and generates results with language model. However, word and morpheme units are not utilized in end-to-end speech recognition model. There are approximately more than 200,000 vocabularies and morphemes; however, this is an excessive number.

Table 1 An example of a split Korean sentence "공부를 합니다." Symbol '/': recognition boundary, symbol '_': space

Unit	Split Sequence of Token
Word	공부를 / 합니다.
Morpheme	공부 / 를 / _/ 하 / ㅂ / 니 / 다
Syllable	공 / 부 / 를 / _/ 합 / 니 / 다
Grapheme	ㄱ / ㅗ / ㅇ / ㅂ / ㅜ / ㄹ / ㅡ / ㄹ / _/ ㅎ / ㅏ / ㅂ / ㄴ / ㅣ / ㄷ / ㅏ

Moreover, the out-of-vocabulary (OOV) problem is caused by the number of recognition units [13].

Syllable-based recognition has relatively fewer dictionaries compared with word or morpheme. For example, English and Japanese have 26 characters and 50 kanas respectively. However, in case of Korean, there are 11,172 numbers of possible syllables. Moreover, there are about 8,000 syllables used in Chinese. This number of syllables is inappropriate for softmax function in end-to-end speech recognition. In case of these languages, phoneme- and grapheme-based recognitions are appropriate for end-to-end speech recognition. The Korean grapheme has 49 numbers of graphemes. However, grapheme-based units require a decoding to sentence. To split syllables to graphemes, we used Unicode Korean. The Unicode is one of the most commonly used writing systems worldwide for representing text. In this study, the 8-bit Unicode transformation format (UTF-8) is capable of encoding all 1,112,064 characters and is used to represent text [27]. Unicode Korean begins with the AC0016 hexadecimal. One perfect Korean syllable is calculated like (12).

$$K = ((C * 21) + V) * 28 + J + AC00_{16} \tag{12}$$

where K shows a Korean syllable, C denotes the Cho-sung (initial consonant), V denotes the Jung-sung (vowel), and J represents the Jong-sung (final consonant). Following (12), each position of grapheme can be calculated as follows.

$$C = ((x - AC00_{16})/28))/21 + 1100_{16} \tag{13}$$

$$V = ((x - AC00_{16})/28))\%21 + 1161_{16} \tag{14}$$

$$J = (x - AC00_{16})\%28)\%28 + 11A8_{16} - 1 \tag{15}$$

where *x* is a target Korean syllable. The hexadecimals at the end represent the first Unicode of each grapheme.

5 Experiments

5.1 Datasets

In this section, the speech recognition corpora were described to verify the proposed model. A total of 740 hours of audio data were compiled by mixing white noise from a quiet office environment with 320 hours of Korean dialogue data and adding these to dialogue recorded on mobile phones.

Table 2 represents the corpora used in this study. Additionally, we used data augmentation to improve the performance of deep neural networks in the domain of speech recognition. The data augmentation we used in this study involved the addition of background noise to the original speech. The background noise was from Soundsnap [21]. Noise signals from a cafe, bus, bus stop, train, restaurant, department store, and car noise, were used.

5.2 Experimental Setup

The End-to-End speech processing toolkit (ESPnet) is used to build speech recognition model [22]. External toolkits, namely Chainer [23] and Pytorch [24], were used for training and decoding, whereas Kaldi was used for feature extraction and data structures. Rescoring factors, such as language model and pronunciation dictionaries were not used to measure just the performance of end-to-end speech recognition model. The shared encoder consists of bidirectional RNN with eight hidden layers, which had 320 nodes each. 320 nodes mean that input frame length was 3 seconds. The attention decoder consists of one layer of unidirectional RNN–LSTM and 49 output nodes, i.e., the number of possible onsets, nuclei, and codas. 83 speech features were used, with delta, pitch, and pitch information added to the 80 filterbank probability of the voice (PoV) information [25]. The error rate of the model was measured with the grapheme error rate (GER) based on the 1,000 sentences of the test dataset. These 1,000 sentences were extracted from SiTEC's reading DB 02 corpus.

Table 2 Corpora used in this study

Training Corpus	Number of Utterances	Hours
ETRI Korean reading DB	100,000	277.78
SiTEC Korean reading DB 01	20,806	57.79
Korean mobile assistant DB	92,874	100.00
Test corpus	Number of utterances	Hours
SiTEC Korean reading DB 02 [partial]	1,000	2.78

Table 3 Experimental results obtained from each tested model (GER: grapheme error rate, and SER: syllable error rate)

System	Token	GER (%)	CER (%)
Hybrid CTC-attention	Grapheme	4.1	10.02
Hybrid CTC-attention	Character	–	70.5
CTC	Grapheme	4.4	18.23
Attention	Grapheme	5.2	10.11

5.3 Experimental Results

Table 3 shows that the proposed hybrid CTC-attention model outperformed both the CTC and attention models. They also show that the grapheme-based approach yields a meaningful result. Note that the best results were achieved with a hybrid model. A Korean syllable-level hybrid CTC-attention model was used, and the results yielded a character error rate (CER) of 70.5%. This suggests that the decision of recognition units had a great impact on the performance of speech recognition. The CTC model yielded a grapheme error rate (GER) of 4.4% and a CER of 18.23%. The CTC, which generated phonemes for each chunk, had a high accuracy in the grapheme unit, but yielded a somewhat lower performance in the entire context. However, the attention model yielded a GER of 5.2% and a CER of 10.11%. Compared with the CTC, the attention network yielded a better performance regarding CER compared with the CTC model.

5.4 Discussion

In this study, most of the speech recognition errors occurred at which the pronunciations differed from the words (e.g., "밥먹었니" is spoken as "밤머건니" when vocalized). Phoneme-based recognition units can decrease the number of these errors. However, this would require an additional procedure for reconverting phonemes into words. Furthermore, when the same syllable has different pronunciations in the onset and coda, recognition errors would occur. For example, 'ㅅ' is pronounced as [s] as an onset but is pronounced as [t] as a coda. If recognition units were determined, differentiating the onsets and codas would decrease errors.

6 Conclusion

We have introduced a method for grapheme-based end-to-end speech recognition system using hybrid CTC-attention network. The method fits all

end-to-end speech recognition system of Korean. This method uses 49 kinds of graphemes in Korean as the output unit, and shows higher performance compared to the end-to-end Korean speech recognizer using 11,172 syllables. Experimental results are measured to absolute 10.02% syllable error rate.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-01772, Development of QA systems for Video Story Understanding to pass the Video Turing Test).

References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, N., ... and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing magazine, Vol. 29, No. 6, pp. 82–97, 2012.

[2] D. Palaz, M. Magimai-Doss, and R. Collobert, "End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition," Speech Communication., Vol. 108, pp. 15–32, 2019.

[3] A. Graves, and S. Fern, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," Proceedings of the 23rd International Conference on Machine learning, pp. 369–376, 2006.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Proceedings of ICLR, 2015.

[5] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," Proceedings of ICML, 2015.

[6] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," arXiv preprint arXiv:1412.1602, 2014.

[7] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, pp. 4945–4949, 2016.

[8] Z. Xiao, Z. Ou, W. Chu, and H. Lin, "Hybrid CTC-Attention based end-to-end speech recognition using subword units," arXiv preprint arXiv:1807.04978, 2018.
[9] B. Li, Y. Zhang, T. Sainath, Y. Wu, W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Brighton, UK, Vol. 1, pp. 5621–5625, 2019.
[10] T. Moriya, J. Wang, T. Tanaka, R. Masumura, Y. Shinohara, Y. Yamaguchi, and Y. Aono, "Joint maximization decoder with neural converters for fully neural network-based Japanese speech recognition," Proceedings of INTERSPEECH, Graz, Austria, pp. 4410–4414, 2019.
[11] D. Amodei et al., "Deep Speech 2: End-to-end speech recognition in English and Mandarin," Proceedings of the 33rd International Conference on International Conference on Machine Learning, Vol. 48, pp. 173–182, 2016.
[12] J. Ha, K. Nam, J. Kang, S. Lee, S. Yang, H. Jung, E. Kim, H. Kim, ... S. Kim, "ClovaCall: Korean Goal-Oriented Dialog Speech Corpus for Automatic Speech Recognition of Contact Centers," retrieved at arXiv:2004.09367 [cs.LG], 2020.
[13] D. Lee, J. Park, K. Kim, J. Park, J. Kim, G. Jang, and U. Park, "Maximum likelihood-based automatic lexicon generation for AI assistant-based interaction with mobile devices," KSII Transactions on Internet & Information Systems, Vol. 11, No. 9, pp. 4264–4279, 2017.
[14] Y. Lee, "Onset analysis of Korean on-glides," in Theoretical Issues in Korean Linguistics, Center for the Study of Language, pp. 133–156, USA, 1994.
[15] E. Variani, T. Bagby, K. Lahouel, E. McDermott, and M. Bacchiani, "Sampled connectionist temporal classification," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Canada, pp. 4959–4963, 2018.
[16] 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," retrieved at arXiv:1706.02737v1[cs.CL], 2017.
[17] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, "Attention-based models for speech recognition," Proceedings of advances in neural information processing systems, Montreal, Canada, pp. 577–585, 2015.
[18] W. Chan, et al. "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," Proceedings of IEEE

International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, pp. 4960–4964, 2016.

[19] Y. He, T. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," arXiv preprint arXiv:1811.06621, 2018.

[20] E. Battenberg, et al. "Exploring neural transducers for end-to-end speech recognition," Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop, Okinawa, Japan, pp. 206–213, 2017.

[21] Soundsnap, "Download sound effect | soundsnap sound library," Soundsnap, https://www.soundsnap.com/, accessed Dec. 26. 2014.

[22] S. Watanabe T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, T. Ochiai, "ESPnet: End-to-end speech processing toolkit," Proceedings of INTERSPEECH, Hyderabad, India, pp. 2207–2211, 2018.

[23] M. I. Abouelhoda and E. Ohlebusch, "CHAINER: Software for comparing genomes," Proceedings of the 12th International Conference on Intelligent Systems for Molecular Biology 3rd European Conference on Computational Biology, Glasgow, UK, pp. 1–3, 2004.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," Proceedings of NIPS, Long Beach, CA, USA, pp. 1–4, 2017.

[25] H. Ting, Y. Yingchun, and W. Zhaohui, "Combining MFCC and Pitch to Enhance the Performance of the Gender Recognition," Proceedings of ICSP, Minneapolis, MN, USA, pp. 3–6, 2007.

[26] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding, Arizona, USA, pp. 167–174, 2016.

[27] T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Multi-head decoder for End-to-end speech recognition," Proceedings of INTERSPEECH, Hyderabad, India, pp. 801–805, 2018.

[28] D. Povey et al., "The Kaldi Speech Recognition Toolkit," Proceedings of ASRU, Hawaii, USA, pp. 1–4, 2011.

[29] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," Computer Speech & Language, Vol. 16, no. 1, pp. 25–47, 2002.

[30] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," IEEE Transaction on acoustic, speech, and signal processing, Vol. 37, No. 3, pp. 393–404, 1989.
[31] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using Lattice-free MMI," Proceeding of Inter-speech 2018, pp. 12–16, 2018.
[32] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," Proc. The 31st International Conference on Machine Learning, Beijing, China, pp. 1764–1772, 2014.
[33] V. Valtchev, J. Odell, P. C. Woodland, and S. J. Young, "Lattice-based discriminative training for large vocabulary speech recognition," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, USA, Vol. 2, pp. 605–608, 1996.
[34] D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya, "Multilingual language processing from bytes," Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies., San Diego, California, pp. 1296–1306, 2016.

Biographies

Hosung Park received the B.E. degree in computer science and engineering from Handong Global University in 2016. He also received the M.E. degree in computer science and engineering from Sogang University in 2018. He is currently pursuing the Ph.D. degree in computer science and engineering with Sogang University. His research interests include speech recognition and spoken multimedia content.

Hosung Park received the B.E. and M.E. degrees in computer science and engineering from Sogang University in 2019 and 2021, respectively. He is a Research Engineer with LG Electronics Institute of Technology, where he was engaged in development of sound recognition for robot devices. His research interest includes speech recognition and object detection.

Hyunsoo Son received the B.E. degree in computer science and engineering from Sogang University in 2019. He is currently pursuing the M.E. degree in computer science and engineering with Sogang University. His research interests include speech recognition and spoken multimedia content search.

Soonshin Seo received the B.A. degree in linguistics and the B.E. degree in computer science and engineering from Hankuk University of Foreign Studies in 2018. He is currently pursuing the Ph.D. degree in computer science and engineering with Sogang University. Since 2021, he has also been a Research Engineer with Naver Corporation. His research interests include speech recognition and spoken multimedia content search.

Ji-Hwan Kim received the B.E. and M.E. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST) in 1996 and 1998, respectively, and the Ph.D. degree in engineering from the University of Cambridge in 2001. From 2001 to 2007, he was a Chief Research Engineer and a Senior Research Engineer with LG Electronics Institute of Technology, where he was engaged in development of speech recognizers for mobile devices. In 2004, he was a Visiting Scientist with MIT Media Lab. Since 2007, he has been a Faculty Member with the Department of Computer Science and Engineering, Sogang University. He is currently a full Professor. His research interests include spoken multimedia content search, speech recognition for embedded systems, and dialogue understanding.