applied sciences

MDPI

Article

KsponSpeech: Korean Spontaneous Speech Corpus for Automatic Speech Recognition

Jeong-Uk Bang, Seung Yun *, Seung-Hi Kim, Mu-Yeol Choi, Min-Kyu Lee, Yeo-Jeong Kim, Dong-Hyun Kim, Jun Park, Young-Jik Lee and Sang-Hun Kim

Artificial Intelligence Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea; jubang@etri.re.kr (J.-U.B.); seunghi@etri.re.kr (S.-H.K.); mychoi@etri.re.kr (M.-Y.C.); mk@etri.re.kr (M.-K.L.); yeojeong@etri.re.kr (Y.-J.K.); dawnkann@etri.re.kr (D.-H.K.); junpark@etri.re.kr (J.P.); ylee@etri.re.kr (Y.-J.L.); ksh@etri.re.kr (S.-H.K.)

* Correspondence: syun@etri.re.kr; Tel.: +82-42-860-5835

Received: 27 August 2020; Accepted: 29 September 2020; Published: 3 October 2020

Abstract: This paper introduces a large-scale spontaneous speech corpus of Korean, named KsponSpeech. This corpus contains 969 h of general open-domain dialog utterances, spoken by about 2000 native Korean speakers in a clean environment. All data were constructed by recording the dialogue of two people freely conversing on a variety of topics and manually transcribing the utterances. The transcription provides a dual transcription consisting of orthography and pronunciation, and disfluency tags for spontaneity of speech, such as filler words, repeated words, and word fragments. This paper also presents the baseline performance of an end-to-end speech recognition model trained with KsponSpeech. In addition, we investigated the performance of standard end-to-end architectures and the number of sub-word units suitable for Korean. We investigated issues that should be considered in spontaneous speech recognition in Korean. KsponSpeech is publicly available on an open data hub site of the Korea government.

Keywords: spontaneous speech corpus; automatic speech recognition; end-to-end

1. Introduction

In artificial intelligence (AI) services, speech recognition systems are used in various applications, such as AI assistants, dialog robots, simultaneous interpretation, and AI tutors. The performance of automatic speech recognition (ASR) systems has been markedly improved by applying deep learning algorithms and collecting large speech databases [1]. Most conventional ASR systems [2,3] consist of various modules, such as the acoustic model, language model, and pronunciation dictionary, which are trained separately. In recent years, end-to-end ASR systems [4–7], which can be directly trained to maximize the probability of a word sequence given an acoustic feature sequence, have been the research focus. Many researchers [7,8] reported that end-to-end ASR systems can significantly simplify the speech recognition pipelines and outperform the conventional ASR systems on several representative speech datasets.

For a natural conversation between humans and machines, spontaneous speech recognition is an essential technology. This involves complex spontaneous speech phenomena, such as unwanted pauses, word fragments, elongated segments, filler words, self-corrections, and repeated words. Thus, spontaneous speech recognition is still challenging and worth investigating [9,10]. In this situation, the end-to-end approach provides one solution for dealing with various spontaneous speech phenomena without intermediate modules carefully designed by human knowledge [11,12]. To build a high-quality end-to-end ASR system [7,13], a large-scale spontaneous speech corpus must be collected to handle a variety of spontaneous speech phenomena.

However, only few Korean spontaneous speech corpora are available. The major languages, such as English, Chinese, and Japanese, have representative conversational and spontaneous speech corpus such as FISHER [14], TED-LIUM2/3 [15,16], HKUST [17], and CSJ [18]. In addition, these datasets are widely used as benchmark datasets for evaluation of ASR models. Meanwhile, Korean is a low-resource language [19], and only a few small-scale or goal-oriented speech corpora have been released, such as the Zeroth project [20] and ClovaCall [21].

In this paper, we introduce a large-scale Korean spontaneous speech corpus on AIHub [22]. AIHub is an open data hub site of the Korea government. As one of the various available open datasets, we are releasing KsponSpeech, an open-domain spontaneous speech corpus in cooperation with the National Information Society Agency (NIA) [22]. This corpus includes 969 h containing 622,545 utterances recorded from 2000 Korean native speakers. Its evaluation data consist of two evaluation datasets of 2.6 and 3.8 h spoken by 60 speakers. In this paper, we describe the recording and transcription method of the Korean spontaneous speech corpus and define the composition of datasets for a speech recognition experiment.

We demonstrate the baseline performance of an end-to-end model trained with KsponSpeech. The experiment was performed with two end-to-end models used as the standard in the ESPnet toolkit [23]: (1) attention-based recurrent neural network (RNN) [24,25] and (2) Transformer [26,27]. Since each model is jointly trained with a connectionist temporal classification (CTC) [24,28] objective function, they are also known as Hybrid CTC/Attention [24] and Hybrid CTC/Transformer models, respectively [26]. However, we abbreviate these to RNN and Transformer, respectively, following previously-used notations [27]. For the speech recognition experiment, we present the baseline performance of two standard end-to-end models. We examined the effectiveness of a data augmentation method known as SpecAugment [29] and then investigated the number of sub-word units [30] suitable for the Korean language. Finally, we compared the performance of the large Transformer architecture.

The rest of the paper is organized as follows: Section 2 describes the KsponSpeech corpus. Section 3 presents the baseline performance of the end-to-end models trained with this corpus. Finally, conclusions are presented in Section 4.

2. KsponSpeech Corpus

2.1. Recording Environment

2.1.1. Speaker Information

The training data include a total of 2000 speakers, and the sex ratio is 1077 women (54%) and 923 men (46%). The evaluation data include a total of 60 speakers who did not participate in the recording of the training data, and the sex ratio is 30 women (50%) and 30 men (50%). The ratio of the participants by age and region was not considered. The number of participants in their 10s, 20s, 30s, 40s, and 50s+ was 315 (16%), 1436 (72%), 127 (6%), 62 (3%), and 60 (3%) respectively. Most of the participants lived in the capital area and spoke a standard language. In addition, speaker information was removed by changing and shuffling the filename to protect privacy.

2.1.2. Collection Environment

The entire dataset was recorded in a quiet office environment. In detail, furniture or sound-absorbing materials were placed inside the office to record in an environment with little reverberation. The recording was performed while two people were wearing headsets, and the distance between the two participants was adjusted to avoid overlapping speech. Speech signals were input from the headset using the unidirectional dynamic microphone Shure WH20 (Shure Inc., Niles, IL, USA) and then stored on a laptop through the Tascam US-122 audio interface (TEAC Corp., Montebello, CA, USA).

2.1.3. Conversation Topic

The conversation topic was selected freely by two participants in consultation with each other. The recording manager was careful not to bias the topic of the conversation. The conversations consist of general topics such as daily conversation, shopping, broadcast, etc. Table 1 provides details about the conversation topics in KsponSpeech.

Table 1. Conversation topics included in KsponSpeech.

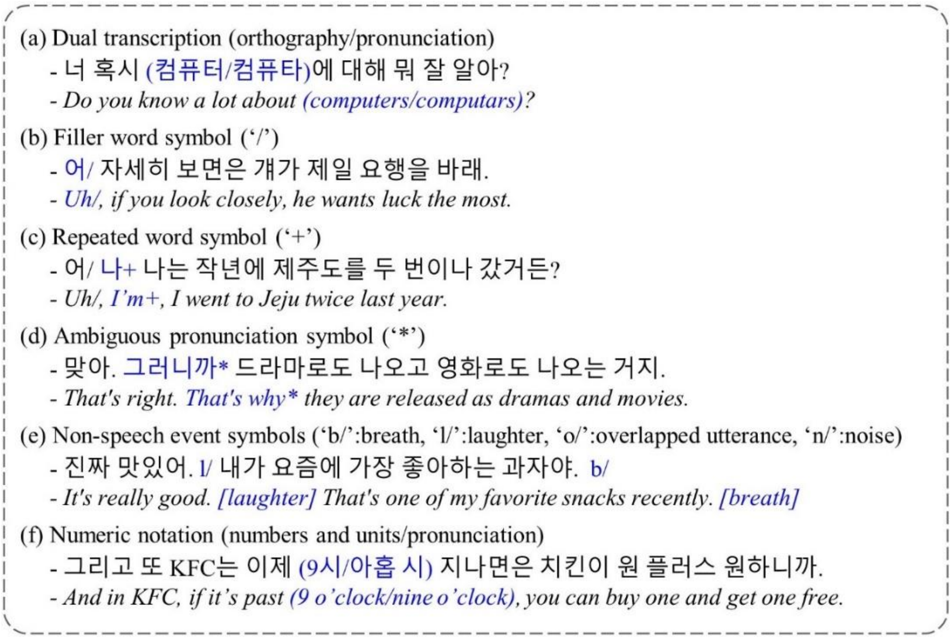
Topic	Sub-Topic	Topic	Sub-Topic
Daily conversation	Anniversary	Weather	Hot/cold weather
	Corporate life		Rainy/heavy snow
	Reason friends		Season
	Residence		Snow/rain/fog
	School life		Temperature
	Self-introduction		Yellow/fine dust
Shopping	Clothing	Hobbies	Blog
	Electronics		Book
	Household goods		Car
	Musical Instrument		Exercise
Broadcasts	Celebrity		Exhibition
	Current		Food
	Drama		Game
	Entertainment		Music
	Movie		Picture
Politics and Economy	Politics		Show
	Real estate		Sports
	Stocks		Travel

2.1.4. Post-Processing

After recording the voices of the two speakers on a stereo channel, we separated them into mono channels for editing and transcription. The obtained data were edited for the following conditions: (1) when saving the speech signals as a file, the speech signals should not be cut off in the middle; (2) long speech signals were divided based on a long silence because it is difficult to divide a spontaneous speech into sentence-level segments; and (3) speech signals were stored in a 16 kHz, 16 bits linear, and little endian PCM format.

2.2. Transcription Rules

This section describes the KsponSpeech transcription process. All transcriptions were performed according to our pre-defined transcription rules and saved in EUC-KR format. Figure 1 depicts a transcription example. Here, special symbols, such as /, (,), *, and +, in the transcription are used only for the purpose of representing dual transcription, disfluent words, and non-speech events. When the special symbols were actually spoken, they were transcribed in the form they were pronounced. The edited and transcribed data were thoroughly checked to see if the speech and transcription were identical and were written according to the transcription rules.



(a) Dual transcription (orthography/pronunciation)

- 너 혹시 (컴퓨터/컴퓨타)에 대해 뭐 잘 알아?
- *Do you know a lot about (computers/computars)?*

(b) Filler word symbol ('/')

- 어/ 자세히 보면은 걔가 제일 요행을 바래.
- *Uh/, if you look closely, he wants luck the most.*

(c) Repeated word symbol ('+')

- 어/ 나+ 나는 작년에 제주도를 두 번이나 갔거든?
- *Uh/, I'm+, I went to Jeju twice last year.*

(d) Ambiguous pronunciation symbol ('*')

- 맞아. 그러니까* 드라마로도 나오고 영화로도 나오는 거지.
- *That's right. That's why* they are released as dramas and movies.*

(e) Non-speech event symbols ('b/':breath, 'l/':laughter, 'o/':overlapped utterance, 'n/':noise)

- 진짜 맛있어. l/ 내가 요즘에 가장 좋아하는 과자야. b/
- *It's really good. [laughter] That's one of my favorite snacks recently. [breath]*

(f) Numeric notation (numbers and units/pronunciation)

- 그리고 또 KFC는 이제 (9시/아홉 시) 지나면은 치킨이 원 플러스 원하니까.
- *And in KFC, if it's past (9 o'clock/nine o'clock), you can buy one and get one free.*

Figure 1. Examples for transcription rules: (**a**) dual transcription, (**b**) filler-word, (**c**) repeated-word, (**d**) ambiguous pronunciation, (**e**) non-speech event, and (**f**) numeric notation. Italics indicate examples translated into English.

2.2.1. Dual Transcription

When deviating from the standard pronunciation, or when two or more pronunciations were possible for the same transcription, we used both orthographic transcription and phonetic transcription in parallel. We call this dual transcription. The orthographic transcription is written according to Korean standard orthographic rules; the phonetic transcription is written as close to the original sound as possible. They were initially prepared for training the acoustic model and the language model of the conventional ASR system. In the end-to-end ASR system, both notations can be used depending on the purpose. When using the dual transcription, parentheses are only used to indicate the range of the dual transcription. Figure 1a shows an example of dual transcription.

2.2.2. Disfluent Speech Transcription

Disfluent speech is marked with forward slash (/) and addition (+) symbols with their transcription. Here, the/symbol mainly indicates filler words, which generally contain little to no lexical content, such as "uh" and "um" in English. Figure 1b shows an example of a filler word. The + symbol mainly indicates repeated words or word fragments and self-corrections. Figure 1c shows an example of a repeated word.

2.2.3. Ambiguous Pronunciation

Words that were difficult to understand or whose pronunciation was ambiguous were transcribed with an asterisk (*) at the end. This symbol is attached to words that could not be recognized as a single word but could be estimated according to the surrounding circumstances. Figure 1d shows an example of the word with ambiguous pronunciation. The example seems to be fine, but their actual sounds are abbreviated. The * symbol was not added if pronounced clearly. Words that were difficult to estimate despite considering the surroundings are marked with u/for unknown words.

2.2.4. Non-Speech and Noise Notation

For non-speech event notation, we use b/, l/, o/, and n/ symbols, which represent breath sound, laughter, utterance overlapped with other participant's speech and is written at the beginning of the transcription, and other noises, respectively. Figure 1e shows an example of non-speech event notations.

2.2.5. Numeric Notation

Numbers are doubly transcribed to reflect their pronunciation. They are composed of numbers and units and their pronunciation. Korean uses a word-phrase unit, unlike English, consisting of one or more words as the basic unit. Therefore, we needed a criterion for whether to connect or space between numbers and units. For this reason, we separated numbers from the units such as "year", "month", "day", and "minute". The pronunciations of numbers are written with spaces in decimal units. Figure 1f shows an example of numeric notation.

2.2.6. Abbreviation Notation

When abbreviations and foreign language words are pronounced differently from the standard pronunciation, we double transcribed the abbreviations and the actual pronunciation. Frequently used abbreviations, such as "KBS" for Korean Broadcasting System or "FIFA" for the International Federation of Association Football, and foreign language words were indicated as they are if they were spoken in general pronunciation.

2.2.7. Word Spacing and Punctuation

As mentioned earlier, Korean uses word-phrase units, and the spaces between them are sometimes ambiguous. Therefore, we tried to ensure that the space between words was written correctly according to the Korean standard orthographic rules. If we could not clearly determine whether to use spaces even after following the rules, we added a space between words. Punctuation marks such as periods, question marks, and exclamation marks were placed at the end of sentences. Commas are used to indicate the context in the middle.

2.3. Corpus Partitions for Speech Recognition

Distributing a large-scaled corpus as a single large archive may be impractical and inconvenient. Thus, KsponSpeech was compressed by dividing it into five archives for training and one archive for evaluation. First, the training portion consists of a total of 622,545 utterances. Among them, 620,000 utterances are divided by 124,000 and stored in five compressed files. The remaining 2545 utterances are additionally stored in the last (fifth) compressed file. Here, the first 620,000 utterances were designated as "Train", and the last 2545 utterances were designated as "Dev". Here, Dev is used to find the optimal parameters of an end-to-end model, and their speakers were included in Train.

The evaluation data were stored as an archive containing 6000 utterances. This was built by selecting 100 utterances per person from conversations of 60 speakers, which were not included in the training data. Subsequently, the evaluation data were divided into two subsets according to the perplexity [31]. To do this, we first built a three-gram language model [31] with the transcription of the training data. Then, the utterances of each speaker were ranked according to the perplexity, and were divided roughly into 50 utterances each, with the lower-perplexity being designated as "eval-clean", and higher-perplexity designated as "eval-other". In the evaluation data, eval-clean has a perplexity of 444 and eval-other has a perplexity of 3106. Eval-clean has a context similar to the training data and consists of utterances that are shorter and easier to recognize than eval-other. Note that perplexity is measured in word-phrase unit, not in word unit. Table 2 shows the data subsets in KsponSpeech.

Table 2. Data subsets in KsponSpeech.

Subset	Hours	No. Utterances	No. Speakers (Male/Female)	Filenames
Train	965.2	620,000	2000 (923/1077)	KsponSpeech_000001–620000
Dev	3.9	2545	1348 (619/729)	KsponSpeech_620001–622545
Eval-clean	2.6	3000	60 (30/30)	KsponSpeech_E00001–E03000
Eval-other	3.8	3000	60 (30/30)	KsponSpeech_E03001–E06000

3. Speech Recognition Results

To validate the effectiveness of KsponSpeech, we first demonstrate the performance of the two end-to-end models of the RNN [24] and the Transformer [26] used as the standard in the ESPnet toolkit [23]. We then investigate the number of sub-word units [30,32] suitable for Korean and present the performance of the large Transformer architecture. We finally explore the methods for generating clean transcriptions in spontaneous speech.

3.1. Experimental Setups

All speech recognition experiments were performed using the ESPnet toolkit [23]. Most of the hyper-parameters followed the default settings provided by the toolkit, especially the recipe of LibriSpeech [33,34]. Each model was jointly trained with the CTC objective function as an auxiliary task [24,26]. We used four 1080Ti GPUs for the training stage and did not use external language models for the decoding stage. All experiments employed 83-dimensional features, including 80-dimensional log-Mel filterbank coefficients with 3-dimensional pitch features per frame. The features were normalized by the mean and variance for the training set. We removed utterances having more than 3000 frames or more than 400 syllables due to GPU memory efficiency.

The RNN and Transformer were mostly configured by following the settings previously provided [27,34]. For RNN, the encoder was composed of 2 VGG blocks [27] followed by 5 layers of bidirectional long short-term memory (BLSTM) with 1024 cells in each layer and direction. The attention layer used a location-aware attention mechanism [35] with 1024 units. The decoder was 2 layers of unidirectional long short-term memory (LSTM) with 1024 cells. The training was performed using AdaDelta [36] with early stopping and dropout regularization [27]. For the decoding, we used a beam search algorithm with a beam size of 20 and CTC weight of 0.5.

For Transformer, the encoder used 12 self-attention blocks stacked on the 2 VGG blocks, and the decoder used 6 self-attention blocks. For every Transformer layer, we used 2048-dimensional feedforward networks. For multi-head attention, we employed two configurations: 4 attention heads with 256 dimensions (“small Transformer”) and 8 attention heads with 512 dimensions (“large Transformer”). The training was performed using Noam [6] without early stopping. For regularization, we also adopted warmup-steps, label smoothing, gradient clipping, and accumulating gradients described previously [27]. For the decoding, we used a beam search algorithm with beam sizes of 10 and 60 and CTC weights of 0.5 and 0.4 for small and large Transformer, respectively. More detailed configurations are described in Appendix A.

Transcriptions for training and evaluation were generated by the following text processing: first, we used the orthographic transcription in the original transcript consisting of the dual transcription. After that, all punctuation marks such as period and question marks and non-speech symbols such as overlap, breath, and laughter, except for unknown words, are removed from the transcription. Finally, we generate a transcription by removing the “/” symbol for the filler-words and the “+” symbol for repeated-words and self-correction.

3.2. Evaluation Metrics

As evaluation metrics, we used character error rate (CER), word error rate (WER), and space-normalized WER (sWER), which are described below. All metrics were measured using the Score Lite toolkit [37], which is a tool for scoring and evaluating the output of speech recognition systems. This toolkit compares the hypothesis text (HYP) output by the speech recognizer and the reference text (REF). Here, the hypothesis text and the reference text are composed of three units: (1) characters (syllables in Korean), (2) words (word-phrases in Korean), and (3) space-normalized words. Figure 2 provides examples of calculations for metrics, and Appendix B shows their pseudocode.

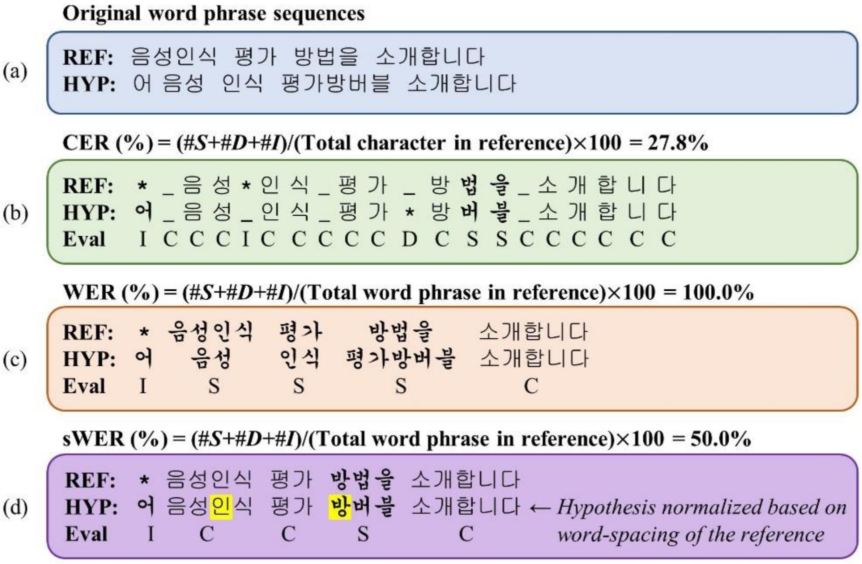

Original word phrase sequences
(a)
REF: 음성인식 평가 방법을 소개합니다
HYP: 어 음성 인식 평가방버블 소개합니다
CER (%) = (#S+#D+#I)/(Total character in reference)×100 = 27.8%
(b)
REF: * _ 음 성 * 인 식 _ 평 가 _ 방 법 을 _ 소 개 합 니 다
HYP: 어 _ 음 성 _ 인 식 _ 평 가 * 방 버 블 _ 소 개 합 니 다
Eval I C C C I C C C C C D C S S C C C C C C
WER (%) = (#S+#D+#I)/(Total word phrase in reference)×100 = 100.0%
(c)
REF: * 음성인식 평가 방법을 소개합니다
HYP: 어 음성 인식 평가방버블 소개합니다
Eval I S S S C
sWER (%) = (#S+#D+#I)/(Total word phrase in reference)×100 = 50.0%
(d)
REF: * 음성인식 평가 방법을 소개합니다
HYP: 어 음성인식 평가 방버블 소개합니다 ← Hypothesis normalized based on word-spacing of the reference
Eval I C C S C

Figure 2. Examples of units and metrics: (**a**) the human transcribed reference text (REF) and the hypothesis text (HYP) obtained from recognizer, (**b**) character units and character error rate (CER), (**c**) word-phrase units and word error rate (WER), and (**d**) space-normalized word-phrase units and space-normalized word error rate (sWER). In the evaluation results (*Eval*), *C*, *S*, *I*, and *D* denote the correct, substituted, inserted, and deleted words, respectively. A yellow box denotes space-normalized words, underscore denotes whitespace, bold text denotes incorrect words, and asterisk denotes inserted words.

The CER was measured by converting character or sub-word units predicted from the end-to-end model into character units, as shown in Figure 2b. Here, the CER was calculated using both character units and spaces. The WER was measured after restoring the sub-word units predicted from the end-to-end model to the original word units, as shown in Figure 2c. This is an evaluation metric commonly used in speech recognition and may present incorrect performance depending on whether or not spaces are used in Korean, as described in Appendix B.

Finally, we propose sWER as a new evaluation metric. In Korean, space rules are flexible; inconsistent spacing is frequently seen in spontaneous speech transcriptions, like in KsponSpeech. However, this causes a problem in the evaluation of speech recognition because correct results are classified as errors due to this spacing variation. Thus, we used sWER, which gives a more valid word error rate by excluding the effects of inconsistent spaces. This metric was measured from space-normalized texts, which was performed only on the hypothesis text, based on spaces in the reference text, as shown in Figure 2d. The detailed algorithm is described in Appendix B. In all evaluation processes, words with the same definition but different forms such as “2” and “two”, were still regarded as incorrect.

3.3. Comparison of RNN and Transformer Architectures

We demonstrate the performance differences between the RNN and Transformer models and then examine the performance by applying SpecAugment [29], which is one of the data augmentation methods. The Transformer experiments used the configuration of the small Transformer. Each model used 2306 Korean syllables (including a space symbol) observed in the training data as output nodes. Table 3 shows the performance of the RNN and Transformer models according to whether or not the SpecAugment is applied.

Table 3. Performance of recurrent neural network (RNN) and Transformer models.

Models	SpecAugment	Metric (%)	Dev	Eval-Clean	Eval-Other
RNN	No	CER	7.7	9.2	10.6
		WER	19.7	24.5	30.0
		sWER	15.3	16.9	20.2
	Yes	CER	7.5	8.6	9.8
		WER	19.1	23.2	28.3
		sWER	14.7	15.6	18.5
Transformer	No	CER	7.2	8.7	9.7
		WER	18.5	23.5	28.3
		sWER	14.1	15.9	18.3
	Yes	CER	**6.5**	**8.0**	**9.0**
		WER	**17.1**	**21.9**	**26.6**
		sWER	**12.7**	**14.2**	**16.5**

The bold number is the best performance.

We first compared the performance of each model on CER. The evaluation data consisted of the three datasets, Dev, Eval-clean, and Eval-other, summarized in Table 2. Here, the Transformer model outperformed the RNN model in all evaluation datasets. Among each dataset, Dev showed the lowest error rate of 7.2% because it consists of the speakers included in the training data, unlike the other evaluation datasets. Eval-clean and Eval-other showed CERs of 8.7% and 9.7%, respectively.

We also observed a large performance gap between WER and sWER. Here, WER indicated a large difference in performance between the evaluation datasets, whereas sWER indicated a slightly smaller difference. When using sWER, mismatched words, which were classified as incorrect words by spacing, were classified as matched words. Most of the spacing differences occurred due to the inconsistent use of spacing in the training data. Note that both metrics used the same reference text, and the hypothesis text differed only in spaces. Thus, the spacing is important issue in the evaluation for Korean, and sWER is a better metric than WER.

In terms of data augmentation, SpecAugment helped improve the performance of both models. The Transformer model using data augmentation had lower relative character error rates of 9.7%, 8.0%, and 7.2% for Dev, Eval-clean, and Eval-other, respectively. As a result, we confirmed that the Transformer model using SpecAugment performs the best. Thus, we continued to use this model in all subsequent experiments.

3.4. The Number of Sub-Word Units

We investigated the number of sub-word units suitable for Korean. The sub-word unit is the result of concatenating several letters to form a new sub-word unit [24]. Here, we connected letters using the unigram algorithm [30] used by default in the ESPnet toolkit. Table 4 describes the performance according to the number of sub-word units. In the table, the experiment with 2306 sub-word units is the same model that uses 2306 Korean syllables, which performed best in the previous experiments. Additionally, the CER was measured by converting sub-word units predicted from the end-to-end model into character units with whitespace, as shown in Figure 2b.

Table 4. Performance according to the number of sub-word units.

Eval Sets	**Metric (%)**	**No. Sub-Words**				
		2306	**5000**	**8000**	**10,000**	**12,000**
Dev	CER	**6.5**	6.9	6.8	6.8	7.2
	WER	**17.1**	17.7	17.6	17.4	18.4
	sWER	**12.7**	13.4	13.2	13.2	14.0
Eval-clean	CER	**8.0**	8.1	8.2	8.3	8.5
	WER	**21.9**	22.0	22.1	22.3	22.7
	sWER	**14.2**	14.4	14.6	14.6	15.1
Eval-other	CER	**9.0**	9.1	9.3	9.0	9.7
	WER	26.6	26.5	27.1	**26.5**	27.8
	sWER	**16.5**	16.7	17.2	16.6	18.0

We observed that performance declined as the 2306 syllable units were increased. We observed that LibriSpeech [33], an English speech data corpus of similar size to KsponSpeech, showed improved performance when using sub-word units with the same algorithm in our preliminary experiments. The reason for showing different results despite using a similarly sized dataset is probably that the basic units between Korean and English are different. Note that Korean uses syllables and English uses characters as the basic unit. These results are also observed in Mandarin, which uses syllable units as the basic unit, as does Korean. Some researchers [38] reported that character units produce better performance than the sub-word units in Mandarin. In our experiment, syllable units may have already been sufficiently concatenated, unlike English character units. Conversely, syllable units may also be unsuitable for expansion into sub-word units as they are already concatenated units of two or more Korean alphabets [39]. We will conduct an experiment starting with the Korean alphabet in the future work.

3.5. *Size of Transformer Model*

We compared the performance difference between the small and large Transformer architectures. Here, the small Transformer consists of four attention heads with 256 dimensions, and the large Transformer consists of eight attention heads with 512 dimensions. The detailed configuration of both models is described in Appendix A. In addition, they used the 2306 syllable units that produced the best performance in the previous experiments as the sub-word unit. Table 5 displays the performance of two Transformer models with different sizes.

Table 5. Performance of the large and small Transformer models.

Models	**Metric (%)**	**Dev**	**Eval-Clean**	**Eval-Other**
Small Transformer	CER WER sWER	6.5 17.1 12.7	8.0 21.9 14.2	9.0 26.6 16.5
Large Transformer	CER WER sWER	**6.1** **16.4** **11.9**	**7.6** **21.1** **13.4**	**8.5** **25.5** **15.4**

In the experimental results, we observed that the large Transformer architecture performed better than the small Transformer architecture. The large Transformer model showed a reduction in relative sWER of 6.3%, 5.6%, and 6.7% over the small Transformer model for Dev, Eval-clean, and Eval-other, respectively. The actual sizes of the two models were 116 and 297 MB, respectively, with the small Transformer model being 2.6 times larger than the large Transformer model. We present the performance of the Large Transformer model in Table 5 as the baseline performance of the KsponSpeech corpus.

3.6. Clean Transcription Generation

KsponSpeech can contribute to the task of generating clean transcriptions because the disfluency tags can be used for the disfluent words provided by the corpus to produce a clean transcription. In spontaneous speech, disfluent words, such as filler words, self-corrections, and repeated words, reduce the readability of automatically-generated transcriptions. Therefore, we demonstrated the feasibility of approaches generating the fluent transcription directly from disfluent speech using the end-to-end ASR model.

We can attempt two possible approaches. First, fluent transcription can be obtained by detecting all disfluent words and then removing them. In this case, the end-to-end model should be trained to output disfluent words along with their disfluency tag. Then, the disfluency-tagged words are removed through post-processing. Second, fluent transcription can be obtained directly from the end-to-end model. In this case, the model should be trained to generate fluent transcription from disfluent speech.

To perform this experiment, we used end-to-end models trained with two types of transcriptions, as shown in Table 6: (1) disfluent transcription with disfluency tag ("disfluent w/tag") and (2) fluent transcription ("fluent"). The disfluent w/tag type is a transcription containing the disfluency tags and the fluent type is a transcription with the disfluent words completely removed.

Table 6. Examples of transcription types: (a) original transcription, (b) disfluent transcription with disfluency tag, and (c) fluent transcription.

Types	Example of Transcription [1]
(a) Original	나중에 내+ 내 목소리랑 똑같은 (AI/에이아이) 막/나오는 거 아니야? I/
(b) Disfluent w/tag	나중에 내+ 내 목소리랑 똑같은 AI 막/나오는 거 아니야
(c) Fluent	나중에 내 목소리랑 똑같은 AI 나오는 거 아니야

[1] Translated into English, "In the future, won't AI speakers like my voice be created?".

Table 7 below shows the performance for each transcription type. Here, we use the transcription of the fluent type as reference text in evaluation. For the disfluent w/tag model, we first removed all disfluent words from their recognition results and then measured their sWER. For the fluent model, we calculated sWER without additional processing. Table 7 shows that both models performed similarly to each other. For the fluent model, we observed that many disfluent words were automatically removed in the hypothesis text. In the predicted clean transcription, there were also words that were incorrectly inserted or deleted. The possible reason is that many people may miss some disfluency tags on the disfluent words in the process of building a large corpus.

Table 7. Space-normalized word error rate (%) for clean transcription generation.

Eval Sets	Models	Cor	Sub	Del	Ins	sWER
Dev	Disfluent w/tag	88.1	9.4	2.5	2.5	**14.4**
	Fluent	87.8	9.4	2.8	2.2	14.4
Eval-clean	Disfluent w/tag	85.8	10.6	3.6	2.6	16.7
	Fluent	85.5	10.4	4.4	2.0	**16.5**
Eval-other	Disfluent w/tag	84.9	12.1	3.0	2.6	**17.6**
	Fluent	84.6	12.0	3.4	2.4	17.8

Cor, Sub, Del, and Ins denote the correct, substituted, deleted, and inserted words, respectively.

As a result, we demonstrated the feasibility of generating the fluent transcription from disfluent speech using the end-to-end ASR model. KsponSpeech can contribute to clean transcription generation. We think this task will be helpful for a variety of applications that require readability and clarity, such as automatic minutes generation and machine translation.

4. Conclusions

This paper introduced a large-scale Korean spontaneous speech corpus for speech recognition. We call this KsponSpeech, which contains 969 h of 622,545 utterances spoken by 2000 Korean native speakers. We present the baseline performance of an end-to-end model trained with KsponSpeech. Here, we proposed a new evaluation metric, space-normalized word error rate, handling the Korean word-spacing variation. To validate the effectiveness of our corpus, we investigated the performance of standard end-to-end models, the effectiveness of the data augmentation technique, and the number of sub-word units suitable for Korean. As a result, we confirmed that the syllable-based Transformer model trained using the data augmentation technique showed the best performance and presented it as the baseline performance. We also explored approaches to generating clean transcription from disfluent speech. We confirmed that an end-to-end model trained with KsponSpeech can be used to generate the clean transcriptions.

We are releasing the KsponSpeech corpus on the AIHub open data hub site. This corpus can contribute to building a high-quality end-to-end ASR model for Korean spontaneous speech, which can be applied to various AI fields, such as AI assistants, dialog robots, and AI tutors. We expect that KsponSpeech will be widely used as a benchmark corpus for Korean speech recognition.

Author Contributions: Data curation, S.Y., J.-U.B., Y.-J.L., and M.-Y.C.; software, J.-U.B., M.-K.L., and S.-H.K. (Seung-Hi Kim); validation, S.-H.K. (Seung-Hi Kim), Y.-J.K., and S.Y.; investigation, J.-U.B. and D.-H.K.; resources, S.Y.; writing—original draft preparation, J.-U.B.; writing—review and editing, J.P., Y.-J.L., and S.Y.; project administration, S.-H.K (Sang-Hun Kim); all of the authors participated in the project. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korea government (20ZS1140 Development of Core Conversational AI Technologies).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1 shows experimental configurations for RNN and Transformer models. Most of the hyper-parameters follow the default settings provided by the ESPnet toolkit [23]. In Table A1, the Transformer has two architectures according to the number of attention heads and their dimensions.

Table A1. Experimental configurations for RNN and Transformer models.

Models	Configurations	Hyperparameters	
RNN	Model	Encoder type	VGGBLSTM
		No. input layers	2 VGG (subsampling = 4)
		No. encoder layers × cells	5 × 1024
		Decoder type	LSTM
		No. decoder layers × cells	2 × 1024
		Attention type	Location-aware
		No. feature maps	10
		Window size	100
	Training	Optimizer	AdaDelta
		CTC weight	0.5
		Epochs	20
		Early stop (patience)	3
		Dropout	0.2
	Decoding	Beam size	20
		CTC weight	0.5
Transformer	Model	Encoder and decoder types	Transformer
		No. input layers	2 VGG (subsampling = 4)
		No. encoder layers × dim	12 × 2048
		No. decoder layers × dim	6 × 2048
		No. attention heads × dim	4 × 256 or 8 × 512 [1]
	Training	Optimizer	Noam
		CTC weight	0.3
		Label smoothing	0.1
		Epochs	100 or 120 [1]
		Early stop (patience)	0
		Dropout	0.1
		Accumulating gradients	2 or 5 [1]
		Gradient clipping	5
		Warmup-steps	25,000
	Decoding	Beam size	10 or 60 [1]
		CTC weight	0.5 or 0.4 [1]

[1] Configurations of small (left) and large (right) Transformer models.

Appendix B

Algorithm A1 shows the pseudocode for character error rate (CER), word error rate (WER), and space-normalized word error rate (sWER), which we propose as the new evaluation metric for Korean. In Korean, space rules are flexible. This means that even though they hear the same utterance, spaces may be used differently for each person writing the transcription. However, this causes a problem in the evaluation of speech recognition because the hypothesis text with inconsistent spaces is not a problem for humans to read, but it is considered an invalid word in the evaluation metric. For this reason, we generated a space-normalized hypothesis text and then calculated the evaluation metric from it. Algorithm A2 provides the pseudocode for generating the space-normalized word units-based hypothesis text. Here, Algorithm A2 was created by modifying the Levenshtein distance provided by the Kaldi toolkit [40], and we did not consider memory efficiency or code simplification.

Algorithm A1. Calculation of CER, WER, & sWER (%)

```
Input: Character-level reference and hypothesis texts (ref_char,hyp_char)
Output: Character, word, & space-normalized word error rates
function calculate_scores(ref_char,hyp_char)
    //check the number of lines in two texts
    if length(ref_char) != length(hyp_char) then
        return report_error

    //get word-level texts
    ref_word ← char_to_word(ref_char))
    hyp_word ← char_to_word(hyp_char))

    //get word-spacing normalized texts
    $ref\_word_{norm}$, $hyp\_word_{norm}$ = [], []
    for each ref, hyp in (ref_char, hyp_char) do
        $ref\_char_{norm}$, $hyp\_char_{norm}$ = get_norm_text(ref, hyp)
        $ref\_word_{norm}$ ← push(char_to_word($ref\_char_{norm}$))
        $hyp\_word_{norm}$ ← push(char_to_word($hyp\_char_{norm}$))

    //compute CER, WER, and sWER with the sclite toolkit
    CER ← sclite(ref_char, hyp_char)
    WER ← sclite(ref_word, hyp_word)
    sWER ← sclite($ref\_word_{norm}$, $hyp\_word_{norm}$)
    return (CER, WER, sWER);
```

Algorithm A2. Word-spacing normalization

```
Input: Reference and hypothesis texts ($ref$, $hyp$)
Output: Word-spacing normalized texts ($ref_{norm}$, $hyp_{norm}$)
    function $get\_norm\_text(ref, hyp)$
        //do character-level tokenizing with space symbol (‘_’)
        //$ref$: “_ $C_1$ $C_2$ _ $C_3$” → $refs$: [‘_$C_1$’, ‘$C_2$’, ‘_$C_3$’]
        $refs$, $hyps$ ← char_tokenizing($ref$), char_tokenizing($hyp$)
        $rlen$, $hlen$ ← length of $refs$ and $hyps$

        //initialization
        $scores$ ← zeros($hlen$+1, $rlen$+1)
        for $r = 0$ to $rlen$ do $scores_{0,r} = r$;
         for $h = 1$ to $hlen$ do
            $scores_{h,0} = scores_{h-1,0} + 1$
            for $j = 1$ to $rlen$ do
                $hyp_{nosp}, ref_{nosp}$ = remove_space_symbol($hyp_{h-1}, ref_{r-1}$)
                $sub\_or\_cor = scores_{h-1,r-1} + (hyp_{nosp} == ref_{nosp}$ ? 0:1)
                $ins, del = scores_{h-1,r} + 1, scores_{h,r-1} + 1$
                $scores_{h,r}$ = min($sub\_or\_cor, ins, del$)

         //traceback and compute alignment
         $h, r = hlen, rlen$
         $ref_{norm}, hyp_{norm}$ = [], []
        while $h > 0$ or $r > 0$ do
            if $h == 0$ then $= h$; $last\_r = r - 1$;
            else if $r == 0$ then $last\_h = h - 1$; $last\_r = r$;
            else
                $hyp_{nosp}, ref_{nosp}$ = remove_space_symbol($hyp_{h-1}, ref_{r-1}$)
                $sub\_or\_cor = scores_{h-1,r-1} + (hyp_{nosp} == ref_{nosp}$. ? 0:1)
                $ins, del = scores_{h-1,r} + 1, scores_{h,r-1} + 1$
                if $sub\_or\_cor \leq$ min($ins, del$) then
                    $last\_h, last\_r = h - 1, r - 1$
                else
                    $last\_h, last\_r = (ins < del$ ? $h - 1, r{:}h, r - 1)$
                $c_{hyp} = (last\_h =$ ? $null$: $hyp_{last\_h})$
                $c_{ref} = (last\_r =$ ? $null$: $ref_{last\_r})$
                $h, r = last\_h, last\_r$

                //do word-spacing normalization
                if $c_{hyp}$ and $c_{ref}$ are the same after removing space symbol then
                    $c_{hyp} = c_{ref}$
            $ref_{norm}, hyp_{norm}$ ← push($c_{ref}, c_{hyp}$)
         ($ref_{norm}, hyp_{norm}$) ← reverse_list($ref_{norm}, hyp_{norm}$)
            return list_to_string($ref_{norm}, hyp_{norm}$
```

References

1. Wang, D.; Wang, X.; Lv, S. An overview of end-to-end automatic speech recognition. *Symmetry* **2019**, *11*, 1018. [CrossRef]
2. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
3. Dahl, G.E.; Yu, D.; Deng, L.; Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *20*, 30–42. [CrossRef]

4. Chan, W.; Jaitly, N.; Le, Q.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–26 March 2016; pp. 4960–4964.
5. Zhang, Q.; Lu, H.; Sak, H.; Tripathi, A.; McDermott, E.; Koo, S.; Kumar, S. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7829–7833.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
7. Amodei, D.; Ananthanarayanan, S.; Anubhai, R.; Bai, J.; Battenberg, E.; Case, C.; Casper, J.; Catanzaro, B.; Cheng, Q.; Chen, G.; et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In Proceedings of the 33rd International Conference on Machine Learning (ICML), New York City, NY, USA, 19–24 June 2016; pp. 173–182.
8. Chiu, C.C.; Sainath, T.N.; Wu, Y.; Prabhavalkar, R.; Nguyen, P.; Chen, Z.; Kannan, A.; Weiss, R.J.; Rao, K.; Gonina, E.; et al. State-of-the-art speech recognition with sequence-to-sequence models. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4774–4778.
9. Deng, Y.C.; Wang, Y.R.; Chen, S.H.; Chiang, C.Y. Recent progress of mandarin spontaneous speech recognition on mandarin conversation dialogue corpus. In Proceedings of the 22nd Conference of the Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), Cebu, Philippines, 25–27 October 2019; pp. 1–6.
10. Bang, J.-U.; Kim, S.-H.; Kwon, O.-W. Acoustic data-driven subword units obtained through segment embedding and clustering for spontaneous speech recognition. *Appl. Sci.* **2020**, *10*, 2079. [CrossRef]
11. Zayats, V.; Ostendorf, M.; Hajishirzi, H. Disfluency detection using a bidirectional LSTM. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), San Francisco, CA, USA, 8–12 September 2016; pp. 2523–2527.
12. Salesky, E.; Sperber, M.; Waibel, A. Fluent translations from disfluent speech in end-to-end speech translation. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, MN, USA, 2–7 June 2019; pp. 2786–2792.
13. Chen, X.W.; Lin, X. Big data deep learning: Challenges and perspectives. *IEEE Access* **2014**, *2*, 514–525. [CrossRef]
14. Cieri, C.; Miller, D.; Walker, K. The Fisher corpus: A resource for the next generations of speech-to-text. In Proceedings of the 4th International Conference of the Language Resources and Evaluation (LREC), Baltimore, MD, USA, 24–30 May 2004; pp. 69–71.
15. Rousseau, A.; Deléglise, P.; Esteve, Y. Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. In Proceedings of the 9th International Conference of the Language Resources and Evaluation (LREC), Reykjavik, Iceland, 26–31 May 2014; pp. 3935–3939.
16. Hernandez, F.; Nguyen, V.; Ghannay, S.; Tomashenko, N.; Estève, Y. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In Proceedings of the International Conference on Speech and Computer (SPECOM), Leipzig, Germany, 18–22 September 2018; pp. 198–208.
17. Liu, Y.; Fung, P.; Yang, Y.; Cieri, C.; Huang, S.; Graff, D. *HKUST/MTS: A Very Large Scale Mandarin Telephone Speech Corpus*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4274, pp. 724–735.
18. Maekawa, K.; Koiso, H.; Furui, S.; Isahara, H. Spontaneous Speech Corpus of Japanese. In Proceedings of the 2nd International Conference of the Language Resources and Evaluation (LREC), Athens, Greece, 31 May–2 June 2000; pp. 947–9520.
19. Pratap, V.; Sriram, A.; Tomasello, P.; Hannun, A.; Liptchinsky, V.; Synnaeve, G.; Collobert, R. Massively Multilingual ASR: 50 languages, 1 model, 1 billion parameters. *arXiv* **2020**, arXiv:2007.03001.
20. Zeroth Project Homepage. Available online: https://github.com/goodatlas/zeroth (accessed on 5 August 2020).
21. Ha, J.W.; Nam, K.; Kang, J.G.; Lee, S.W.; Yang, S.; Jung, H.; Kim, E.; Kim, H.; Kim, S.; Kim, H.A.; et al. ClovaCall: Korean Goal-Oriented Dialog Speech Corpus for Automatic Speech Recognition of Contact Centers. *arXiv* **2020**, arXiv:2004.09367.

22. AIHub Homepage. Available online: http://www.aihub.or.kr/aidata/105 (accessed on 5 August 2020).
23. Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J.; Unno, Y.; Yalta, N.; Heymann, J.; Wiesner, M.; Chen, N.; et al. ESPnet: End-to-end speech processing toolkit. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), Hyderabad, India, 2–6 September 2018; pp. 2207–2211.
24. Kim, S.; Hori, T.; Watanabe, S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 4835–4839.
25. Watanabe, S.; Hori, T.; Kim, S.; Hershey, J.R.; Hayashi, T. Hybrid CTC/attention architecture for end-to-end speech recognition. *JSTSP* **2017**, *11*, 1240–1253. [CrossRef]
26. Karita, S.; Soplin, N.E.; Watanabe, S.; Delcroix, M.; Ogawa, A.; Nakatani, T. Improving Transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), Graz, Austria, 15–19 September 2019; pp. 1408–1412.
27. Karita, S.; Chen, N.; Hayashi, T.; Hori, T.; Inaguma, H.; Jiang, Z.; Someki, M.; Soplin, N.E.; Yamamoto, R.; Wang, X.; et al. A comparative study on Transformer vs RNN in speech applications. In Proceedings of the 2019 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Singapore, 14–18 December 2019; pp. 449–456.
28. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning (ICML), Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
29. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. Specaugment: A simple data augmentation method for automatic speech recognition. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), Graz, Austria, 15–19 September 2019; pp. 2613–2617.
30. Kudo, T. Subword regularization: Improving neural network translation models with multiple subword candidates. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, 15–20 July 2018; pp. 66–75.
31. Stolcke, A. SRILM—An extensible language modeling toolkit. In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP), Warsaw, Poland, 13–14 September 2002; pp. 901–904.
32. Kudo, T.; Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP), Brussels, Belgium, 31 October–4 November 2018; pp. 66–71.
33. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. Librispeech: An asr corpus based on public domain audio books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 19–24 April 2015; pp. 5206–5210.
34. Librispeech Recipe. Available online: https://github.com/espnet/espnet/tree/master/egs/librispeech (accessed on 5 August 2020).
35. Chorowski, J.K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montréal, QC, Canada, 7–12 December 2015; pp. 577–585.
36. Zeiler, M.D. ADADELTA: An adaptive learning rate method. *arXiv* **2012**, arXiv:1212.5701.
37. Score Lite Toolkit. Available online: https://github.com/usnistgov/SCTK (accessed on 5 August 2020).
38. Zhou, S.; Dong, L.; Xu, S.; Xu, B. A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese. *arXiv* **2018**, arXiv:1805.06239.

39. Kwon, O.-W.; Park, J. Korean large vocabulary continuous speech recognition with morpheme-based recognition units. *Speech Commun.* **2003**, *39*, 287–300. [CrossRef]
40. Levenshtein Distance Provided on the Kaldi Toolkit. Available online: https://github.com/kaldi-asr/kaldi/blob/master/src/util/edit-distance-inl.h (accessed on 5 August 2020).