Article

Acoustic Data-Driven Subword Units Obtained through Segment Embedding and Clustering for Spontaneous Speech Recognition

Jeong-Uk Bang ¹, Sang-Hun Kim ² and Oh-Wook Kwon ^{1,*}

¹ School of Electronics Engineering, Chungbuk National University, 1 Chungdae-ro, Seowon-gu, Cheongju, Chungbuk 28644, Korea; jubang@cbnu.ac.kr

² Artificial Intelligence Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), 218 Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea; ksh@etri.re.kr

* Correspondence: owkwon@cbnu.ac.kr; Tel.: +82-43-261-3374

Received: 8 February 2020; Accepted: 17 March 2020; Published: 19 March 2020

Abstract: We propose a method to extend a phoneme set by using a large amount of broadcast data to improve the performance of Korean spontaneous speech recognition. In the proposed method, we first extract variable-length phoneme-level segments from broadcast data and then convert them into fixed-length embedding vectors based on a long short-term memory architecture. We use decision tree-based clustering to find acoustically similar embedding vectors and then build new acoustic subword units by gathering the clustered vectors. To update the lexicon of a speech recognizer, we build a lookup table between the tri-phone units and the units derived from the decision tree. Finally, the proposed lexicon is obtained by updating the original phoneme-based lexicon by referencing the lookup table. To verify the performance of the proposed unit, we compare the proposed unit with the previous units obtained by using the segment-based k-means clustering method or the frame-based decision-tree clustering method. As a result, the proposed unit is shown to produce better performance than the previous units in both spontaneous, and read Korean speech recognition tasks.

Keywords: acoustic subword unit; phoneme set; spontaneous speech recognition

1. Introduction

The phoneme unit has been long used as the acoustic subword unit [1] for automatic speech recognition (ASR) systems. Here, a phoneme is considered the smallest contrastive linguistic unit that may produce a change of meaning [2]. In recent years, the use of a grapheme unit that does not require grapheme-to-phoneme conversion for speech recognition has been attracting interest, as end-to-end speech recognition systems are becoming more popular. However, the grapheme unit has lower speech recognition accuracy than the phoneme unit because grapheme units are irregularly assigned to speech signals with various spectral patterns, while phoneme units are assigned to speech signals with similar spectral patterns. These results are commonly observed in both conventional speech recognition systems and recent end-to-end ASR systems [3].

In large vocabulary continuous speech recognition, there is a problem that the number of phoneme units is too small to express various acoustic changes. Previous works have confronted this problem using the implicit method [4,5] or the explicit method [6]. Here, we focus on the implicit method using a decision tree [4]. The implicit method extends a context-dependent model according to adjacent phonemes and then uses a decision tree to share the parameters of models with similar acoustic characteristics. This method is helpful in terms of addressing the data sparsity issue and handling unseen contexts. Both the implicit method and explicit method are used commonly in conventional speech recognition systems [4].

In spontaneous speech recognition, the phoneme unit has a problem of acoustically low discrimination. In more detail, the phoneme unit in spontaneous speech has a smaller inter-unit distance and a larger variance than the phoneme unit in read speech, which is one of the major factors contributing to the decrease in recognition accuracy [7,8]. In general, using a decision tree in the implicit method shows improved speech recognition accuracy when segmented from acoustically discriminative units. This is also confirmed by the fact that a speech recognizer for read speech has shown better performance when segmented based on the phoneme unit instead of the grapheme unit. Thus, if we build an acoustically discriminative unit by clustering common spectral patterns from spontaneous speech, we can expect an improvement in the performance of spontaneous speech recognition.

We propose a method to improve the performance of spontaneous speech recognition by extending the phoneme set with a large amount of Korean broadcast data. The proposed unit is extracted in three steps. We first extract variable-length phoneme-level segments and then convert them into fixed-length latent vectors based on a long short-term memory (LSTM) architecture [9]. Finally, we use the decision tree-based clustering algorithm [4,10] to cluster acoustically similar latent vectors and then build a new acoustic subword unit by gathering the clustered vectors. In the unit derivation experiments, we compare the proposed and previous approaches [9,11] in terms of the fixed-length vector extraction and the clustering algorithm. The proposed unit is shown to produce better performance than the acoustic subword units obtained by previous methods, in both spontaneous and read speech recognition tasks.

This paper is an extension of our previous conference paper [9] that improves the clustering method from a k-means clustering algorithm [12] to a decision tree-based clustering algorithm [4]. In addition, we compare our method with another approach [11] to verify the superiority of the proposed method and compare the performance, in detail, by increasing the scope of the evaluation data. This paper is structured as follows. First, in Section 2, we describe related works. In Section 3, we describe a method for building the proposed unit and a method for updating the lexicon. Then, in Section 4, we explain the units derived by each method and show the results of our experiments. Finally, in Section 5, we present our conclusions.

2. Related Works

This paper aims to find a subword unit suitable for spontaneous speech recognition. Similar to our approach, some studies [13–17] have attempted to overcome the limitations of the phoneme unit. These studies focused on automatically deriving subword units from speech signals and constructing a lexicon based on them; this was done to build the subword unit using a data-driven, rather than hand-crafted approach. However, most of these studies were performed decades ago and did not focus on improving the performance of spontaneous speech recognizers.

In recent years, there has been a renewed interest in handling phoneme-like units for low-resource languages. Thus, most of the previous studies [11,18–22] aimed to convert grapheme units to phoneme-like units, rather than focusing on spontaneous speech [9]. Here, the phoneme-like unit is a subword unit, automatically derived from training speech data, not a phoneme unit selected by experts. In general, this subword unit-based speech recognizer uses a lexicon that maps each word to a sequence of subword units, typically phonemes. Here, developing a lexicon [11] for each language requires linguistic knowledge, as well as human effort, which may not be always readily available, particularly for low-resource languages. For this reason, grapheme-based subword units or phoneme-like subword units have been explored in the literature as an alternative to the absence of a phoneme-based lexicon.

Phoneme-like units tend to be data-dependent, as they are typically obtained through the optimization of an objective function using training speech data [11]. Moreover, phoneme-like units may facilitate handling pronunciation variations [1]. For this reason (although different in purpose), we can derive a method for building subword units suitable for the spontaneous speech by applying the methods used to build phoneme-like units. To take advantage of existing studies, we compared the pros and cons between the subword unit extension methods, proposed in our previous study [9], and the most recently studied phoneme-like unit discovery method [11].

In the previous study [9], we extended the existing phoneme, set using a large amount of broadcast speech data, in order to build a subword unit suitable for spontaneous speech recognition. This method first extracts variable-length phoneme-level segments from broadcast data and then converts them into fixed-length latent vectors based on a long short-term memory architecture. Then, we used the k-means algorithm to cluster acoustically similar latent vectors to build a new phoneme set by gathering the clustered vectors. In the visualization and clustering experiments, we confirmed that the acoustic information of each segment is better represented by the latent vector, obtained from the LSTM architecture, rather than by the fixed-length feature vector obtained by simple linear interpolation. In the results, the proposed unit is shown to produce better performance than the phoneme-based and grapheme-based units, in both spontaneous and read speech recognition tasks.

In a previous study [11], a phoneme-like unit was derived by clustering context-dependent (CD) graphemes in Gaussian mixture model (GMM) and hidden Markov model (HMM) frameworks using maximum likelihood criteria. To do this, CD grapheme-based HMM/GMM systems were trained with 39-dimensional cepstral features extracted using the HTK toolkit [23]. Here, each CD grapheme was modeled with a single HMM state. Finally, the phoneme-like units were derived through likelihood-based decision tree-based clustering using singleton questions [4]. As a result, the phoneme-like unit, which is obtained from clustered CD grapheme units, is shown to significantly yield better performance compared to the grapheme unit in the GMM/HMM system. If we use CD phoneme units instead of CD grapheme units, we can build a unit suitable for spontaneous speech, which is the purpose of this paper.

The above two approaches [9,11] can be compared in terms of segment embedding and clustering algorithms. First, the two approaches entail different segment embedding methods. In a previous study [9], phoneme-level segments were represented as fixed-length vectors using the LSTM architecture. In contrast, the previous study [11] extracted grapheme-level segments obtained using a single HMM state with GMM and then applied a clustering algorithm to each frame. Here, processing each frame may be insufficient to represent spontaneous speech with various spectral patterns. On the other hand, the LSTM architecture, which converts a segment consisting of various frames into a latent vector, can better represent spontaneous speech by using long-term dependency. Second, these two approaches used different clustering algorithms: k-means clustering [12] and decision tree-based clustering [4]. K-means clustering is the most common clustering algorithm using Euclidean distance. On the other hand, decision tree-based clustering splits the data by using top-down greedy splitting, which considers the left phoneme, the right phoneme, and the central phoneme. We did not search the literature to compare these two clustering algorithms to derive phoneme-like units. However, we expect that decision tree-based clustering performs better than k-means clustering as the acoustic model training process usually uses a decision tree-based clustering algorithm.

The proposed method builds new subword units by using the segment embedding method of the previous study [9] and a decision tree-based clustering method performed on phoneme-level segments. Our approach in this paper differs from that in [11] because we use segment-based clustering instead of frame-based clustering, and our approach differs from that in [9] because we apply decision tree-based clustering instead of k-means clustering. In addition, the decision tree-based clustering method differs from the approach in [11], which uses the HTK toolkit [23], in that the questions for node splitting are automatically generated by top-down binary data clustering rather than by hand.

3. Proposed Method

This section describes a method for building new acoustic subword units suitable for spontaneous speech recognition as shown in Figure 1. Here, we build new subword units in three steps: Segment extraction, segment embedding, and decision tree-based segment clustering. Next, the pronunciation lexicon for speech recognition experiments is updated by converting the phoneme-based lexicon into the proposed unit-based lexicon.

Figure 1. Framework for building the proposed unit using segment embedding and clustering.

3.1. Segment Extraction

First, we extract phoneme-level segments from the utterance-level speech data. Each segment is extracted using the forced alignment algorithm [24], which is commonly applied to acoustic model training. Here, we used an acoustic model with a deep neural network (DNN) structure, as applied in the previous work [25], and 40-dimensional log-Mel filter-bank (FBank) features are spliced over time, with a context size of 15 frames (± 7 frames). The extracted segments, which total about 50 million, consist of different numbers of frames, even if they have the same phoneme label. Hereafter, we refer to the phoneme-level segment as a variable-length segment.

3.2. Segment Embedding

Next, we convert the variable-length segments to fixed-length vectors. In a previous study [9], we compared the methods of linear interpolation, LSTM auto-encoders [26], and LSTM classifiers to convert variable-length segments to fixed-length vectors. According to [9], the fixed-length vectors obtained using linear interpolation do not accurately represent the acoustic differences of each phoneme. The fixed-length vectors, obtained from the LSTM auto-encoder architecture, were shown to cluster with frame length information as the frame length information is the most important information for restoring a latent vector to the original feature vectors of the segment. Finally, the fixed-length vectors obtained from the LSTM classifier architecture are more suitable for fixed-length vector extraction to derive subword units.

Based on these previous results, we decided to convert the variable-length segments to fixed-length vectors by using the LSTM classifier structure [9], shown in Figure 2. The fixed-length vectors are extracted at the point between the LSTM encoder (blue) and the DNN decoder (red), where the acoustic properties of each segment are projected into the latent space.

Figure 2. LSTM structure for fixed-length vector extraction.

s_i^l represents the i -th segment with the label l . Here, the label l consists of 40 phoneme symbols commonly used for Korean speech recognition. f_k represents the k -th 40-dimensional log-Mel filter-bank feature spliced in time with adjacent ± 1 frame for a segment of length n . x_i^l represents the i -th fixed-length vector with the label l . y^l represents the one-hot vector of label l . Similar to the model structure, we used conventional LSTM consisting of an input gate, a forget gate, and an output gate, and the peephole connections were not used. As the model parameter, we used one hidden layer with 80 LSTM units with the tanh activation function and the Adam optimization algorithm with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, a batch size of 64, and 10 epochs with a learning rate of 0.01.

Here, we did not optimize parameters such as the dimension size or the number of layers at the encoder and decoder stages because, whenever the model parameters change, we have to perform all of the following steps again: Segment embedding, segment clustering, and acoustic model training for a speech recognition experiment. However, in light of previous studies [26–28], the 80-dimensional latent vector used in this study is of an appropriate size to represent the segment. A previous study [27] argued that speech segments perform best when they are represented with 50–200 dimensions in nonlinear embedding using Laplacian eigenmaps. In addition, other studies [26,28] also used 60–100 dimensional latent vectors to represent speech segments or acoustic features.

3.3. Decision Tree-Based Segment Clustering

We then perform decision tree-based clustering from fixed-length vectors to find subword units consisting of acoustically similar vectors. Decision tree-based clustering, in HMM-based ASR, is generally used for HMM state clustering and testing methods to address the data sparsity issue and address unfamiliar contexts [4]. In addition, a previous study [11] used this clustering algorithm to derive phoneme-like units from context-dependent grapheme units. Similarly, we use decision tree-based clustering to derive new subword units from the fixed-length vectors to obtain phoneme-level segments. The proposed method, as shown in Figure 3a, works in three sub-steps: obtaining statistics, building questions, and constructing a decision tree. The clustering algorithm was implemented by modifying a clustering method provided by the Kaldi toolkit [29] for our purposes (e.g., obtaining statistics on the fixed-length vectors instead of the feature vectors).

Figure 3. Block diagram and a simple example of decision tree-based clustering. (a) Block diagram of the decision tree-based clustering; (b) statistical table obtained at point A; (c) questions obtained at point B; (d) decision tree obtained at point C.

The decision tree was implemented via top-down greedy splitting. Each question for a split is automatically generated by top-down binary clustering of the data to ensure the highest likelihood under the node in the corresponding branches. To build the decision tree, we first collect cumulative statistics for each CD phoneme unit into a statistical table as shown in Figure 3a. Here, the CD phoneme unit uses a tri-phoneme that considers both, the left and the right neighboring phonemes. Our method generates a statistical table using the segment-level fixed-length vectors, unlike the previous method [11] using frame-level feature vectors. The statistical table consists of the number of samples (*count*), the sum of the fixed-length vectors (x), and the sum of the squares of the fixed-length vectors (x^2). The statistical table is then used to estimate the weight, mean, and covariance for calculating the log-likelihood of a single Gaussian value at each node.

The next steps of building questions and building a decision tree are performed in the same way as the decision tree-based clustering method using the Kaldi toolkit [29]. In the step involving building questions, we generate questions by using phoneme groups and the original phoneme set, as shown

in Figure 3b. Here, for simplicity, we assume that only four phonemes, a , b , c , and d , exist. To obtain the phoneme groups, we first build a decision tree with a single root node and put all the phonemes in the root node ($Q0$). Then, we select a node with the highest log-likelihood improvement in the decision tree and find the question with the highest log-likelihood difference when dividing the node into the left branch and the right branch. In Figure 3c, the phoneme group tree shows the maximum log-likelihood difference when the node $Q0 \{a, b, c, d\}$ is split into the left branch $Q1 \{a, b\}$ and the right branch $Q2 \{c, d\}$. We repeat this process until the total number of leaf nodes is the number of phonemes. As a result, we generate nine questions ($Q0 \sim Q8$) consisting of five phoneme groups ($Q0 \sim Q4$) and four original phonemes ($Q5 \sim Q8$).

Equations (1) and (2) show the mean and covariance of each node, where N_p is the total count of data belonging to node p . Equation (3) represents a single Gaussian value of the D -dimensional fixed-length vectors belonging to node p . Finally, Equations (4) and (5) show the log-likelihood L_p for node p and the log-likelihood difference ΔL . Here, L_{old} is the log-likelihood of a node before splitting, and L_{left} and L_{right} are the log-likelihood of the left, and right, nodes after splitting, respectively:

$$\mu_p = \frac{1}{N_p} \sum_{c=1}^{N_p} x_c \quad (1)$$

$$\Sigma_p = \frac{1}{N_p} \sum_{c=1}^{N_p} x_c^2 - \mu_p^2 \quad (2)$$

$$f_p(x) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_p|}} \exp\left(-\frac{1}{2}(x - \mu_p)^T \Sigma_p^{-1} (x - \mu_p)\right) \quad (3)$$

$$L_p = E \left[\log \left(\prod_{c=1}^{N_p} f(x_c) \right) \right] = N_p \left(-0.5 (\log |\Sigma_p| + D \log 2\pi) \right) \quad (4)$$

$$\Delta L = L_{left} + L_{right} - L_{old} \quad (5)$$

Finally, a decision tree is built using the statistical table and questions, as shown in Figure 3d. To build a decision tree, we first build a decision tree using a single root node and then create leaf nodes consisting of each original phoneme. Next, we calculate the log-likelihood difference before the separation and the log-likelihood after the separation according to the context of each phoneme and find the optimal question with the highest difference. Finally, we repeat the process of splitting the leaf nodes with the highest log-likelihood difference for all leaf nodes. The leaf nodes of the decision tree have acoustically similar context phonemes, which are used as new subword units instead of the phoneme units for spontaneous speech recognition.

As the parameter for building the decision tree, we used 40 questions, which equals the number of CI phonemes, following the default setting of the Kaldi toolkit [29]. The decision tree was constructed to derive 120 leaf nodes according to the left and the right neighboring phonemes. Here, the number of leaf nodes indicates the number of subword units to be extended.

3.4. Lexicon Update

We next update the phoneme-based lexicon to the proposed unit-based lexicon. The mapping from the phoneme sequence of the word to the corresponding proposed unit sequence is carried out using the decision tree. To build the proposed lexicon, we select the leaf node of the decision tree corresponding to the CD phoneme in question. To do this, the new pronunciation sequence of each word is generated by examining the word from left-to-right one CD phoneme symbol at a time. Based on the context information, the CD phoneme starts at the root node of the decision tree and goes down until the left node is found. The index of the leaf node corresponding to the CD phoneme is

output in the new subword unit. The process then moves onto the next CD phoneme unit, and the corresponding CD phoneme unit is found in a similar way. The subword sequence of each word is constructed by concatenating the proposed unit that has been found.

Figure 4a shows a block diagram of the lexicon update, and Figure 4b shows an example of a phoneme-based lexicon consisting of N words. Figure 4c shows the updated lexicon obtained by using the decision tree. In Figure 4b, the first word W_1 has $[a\ b]$ as the pronunciation sequence. Here, the tri-phoneme of the first phoneme 'a' has a word boundary symbol '\$' on the left and the phoneme symbol 'b' on the right. As shown in Figure 4d, we can express this CD phoneme unit as "\$ - a + b". Here, the "\$ - a + b" symbol denotes that the '\$' symbol (i.e., the left-hand symbol) precedes the 'a' phoneme, and the 'b' phoneme (i.e., the right-hand phoneme) follows it. The decision tree first asks what the center phoneme of the unit is and then goes to the left node corresponding to phoneme 'a' (red line). After that, the decision tree checks whether the left symbol '\$' belongs to the question $\{a, c\}$. In this case, we seek out a node corresponding to 'No', since the symbol does not belong to the question. The decision tree then checks again whether the right symbol 'b' belongs to the question $\{a, b\}$. In this case, we go to a node corresponding to $\{a, b\}$, which means 'Yes'. Finally, we reach the leaf node 'U2'. The leaf node 'U2' is then output as a new subword unit for the CD phoneme "\$ - a + b". This process then moves on to the next CD phoneme " $a - b + \$$ ", and the CD phoneme that corresponds to it is found in a similar way. As a result, we constructed the proposed lexicon consisting of 120 subword units from the original lexicon consisting of 1.8 million words and 40 phoneme units. Note that 64,000 ($= 40 \times 40 \times 40$) CD phonemes, rather than 40 CI phonemes, are mapped to 120 subword units. Further, this is a crude computation; in most ASR systems, the left and right contexts are binned into, for example, 10 broader classes, leaving $10 \times 40 \times 10$ CD phonemes. We use the proposed lexicon for the acoustic model training and speech recognition experiments.

Figure 4. Block diagram and an example of the lexicon update. (a) Block diagram of the lexicon update; (b) phoneme-based lexicon given at point D; (c) proposed lexicon obtained at point E; (d) example of the lexicon update using CD phoneme "\$ - a + b".

4. Experiments and Results

This section compares the speech recognition performance of the units derived by the previous method and the proposed method. We first compare the subword units obtained by various methods and then verify the speech recognition performance using lexicons composed of those units.

4.1. Experimental Setup

All speech recognition experiments were performed using the Kaldi toolkit [29]. The input features were equivalent to the global mean-variance normalized 40-dimensional log-Mel filter-bank feature spliced across ± 2 frames. The acoustic model used was LSTM-projection (LSTMP) trained by

layer-wise back-propagation supervised with a 3-state left-to-right hidden Markov model. The LSTMP model has 3 layers, and each layer has 1,024 memory cells and 256 hidden nodes in the projection and the output layer, with about 8,033 nodes using the softmax activation function. The language model was trained via Kneser-Ney discounting (cut-off 0-3-3) using the SRILM toolkit [30] and included 1.8 million unigrams, 32 million bigrams, and 43 million trigrams from various corpora, including broadcast subtitles (excluding the evaluation datasets). The decoder was set to have an acoustic weight of 0.125, a beam-size of 10.0, and a lattice-beam of 5.0. The recognition results were compared by calculating the word error rate (WER) in the decoding unit.

We used about 1000 h of Korean broadcast data [25,31] for the subword unit derivation and the acoustic model training. This database was automatically constructed from broadcast audio and their subtitle text using a lightly supervised approach [32]. The collected data contain some incorrectly transcribed text that does not match the speech signal and has a mixture of background noise and music. The broadcast data were extracted from broadcasts recorded on South Korean broadcasting channels from March to June 2016. The broadcast data are largely composed of seven genres: News, current affairs, documentary, culture, drama, children, and entertainment. Here, we note that the news, documentary, and current affairs genres, which include many reading-style utterances, comprise 25%, 12%, and 5%, respectively (42%), of the total broadcast data. On the other hand, the culture, drama, entertainment, and children's genres, which include many spontaneous-style utterances, comprise 22%, 16%, 12%, and 3%, respectively (54%) of the total broadcast data. The remaining 4% consists of sports broadcasts and music programs.

The evaluation datasets featured 1.5 h of broadcast news data ('News'), 3.5 h of web news data ('Web'), and 7 h of spontaneous data ('SPT'). Here, the News dataset may contain the same speakers and similar sentences because it was collected during a similar time to the training data. The Web dataset, on the other hand, consists of sentences on subjects different from those in the training dataset, since the Web dataset was collected from web sites in 2019. The SPT dataset contains unwanted pauses, word fragments, filler words, self-corrections, and repeated words. All evaluation datasets were not used for training the acoustic model or building the proposed unit.

4.2. Acoustic Data-Driven Subword Units

This section describes the seven subword units obtained in the previous and proposed methods. Table 1 shows the based-units, clustering samples, and clustering methods for deriving each unit.

Table 1. Summary of the subword units obtained in the various methods.

Subword Unit	Based-Unit	Clustering Samples	Clustering Method
(a) Grapheme-52	Grapheme	-	-
(b) Phoneme-40	Phoneme	-	-
(c) Pho-LTV-Kmeans-120 [9]	Phoneme	Latent vectors	k-Means
(d) Gra-MFCC-Tree-120 [11]	Grapheme	MFCC vectors	Decision tree
(e) Pho-MFCC-Tree-120	Phoneme	MFCC vectors	Decision tree
(f) Pho-LTV-Tree-120 (<i>proposed</i>)	Phoneme	Latent vectors	Decision tree

As the subword unit, we first used the grapheme units of Table 1(a). The Korean language is written in Hangul, a syllabic unit character that consists of initial, medial, and final sounds. Here, Hangul creates 11,172 ($= 19 \times 21 \times (27 + 1)$) syllables by combining 19 initial consonants, 21 medial vowels, and 27 final consonants (including no final consonant). The total number of unique letter units that have eliminated duplicate consonants from the initial and final consonants is 52 (including silence). In this paper, we call the unique letter units the grapheme units. The phoneme units of Table 1(b), on the other hand, typically consist of 40 phonemes, including silence. Here, the phoneme unit is derived from a grapheme-to-phoneme converter [33,34] and has acoustically high discrimination despite having a smaller number than the grapheme units. However, the phonemes in spontaneous

speech still possess acoustically ambiguous units. In Korean consonants, the phonemes /b/, /d/, and /g/ are a typical set of similar consonants, and their corresponding double-consonants, /b^h/, /t^h/, and /k^h/, also have similar sounds. In Korean vowels, the phonemes /ε/ and /e/ are the two most similar vowels. Note that ordinary Koreans usually do not distinguish the two phonemes /ε/ and /e/ in daily life.

Next, we extracted the Pho-LTV-Kmeans-120 units of Table 1(c) using the method proposed in a previous study [9]. This previous study first extracted fixed-length latent vectors from broadcast data in the same way we proposed in Section 3. Then, the authors used the k-means clustering algorithm and built a new phoneme set by gathering the clustered vectors. On the other hand, our study used decision tree-based clustering instead of k-means clustering. This clustering method is the main difference between the proposed method and this previous study [9], where the number of clusters (*k*) was determined by speech recognition experiments. On the other hand, in this paper, we selected the optimal number of clusters using the Davies–Bouldin (DB) index [35], which is an evaluation measure of the clustering algorithm. Here, the DB index is calculated using Equations (6–8) below:

$$DB = \frac{1}{N} \sum_{i=1}^N \max_{j \neq i} \frac{S_i + S_j}{M_{i,j}} \quad (6)$$

$$S_i = \left(\frac{1}{T_i} \sum_{j=1}^{T_i} |x_j - \mu_i|^2 \right)^{\frac{1}{2}} \quad (7)$$

$$M_{i,j} = \|\mu_i - \mu_j\|_2 = \left(\sum_{k=1}^n |\mu_{k,i} - \mu_{k,j}|^2 \right)^{\frac{1}{2}} \quad (8)$$

In Equation (6), S_i is a measure of scatter within the i -th cluster, and $M_{i,j}$ is a measure of cluster separation between the n -dimensional i -th cluster and the j -th cluster. Here, the measure of scatter (S_i) is denoted as in Equation (7) and indicates the average distance between the j -th latent vector x_j and the centroid A_i in the i -th cluster with T_i latent vectors. The measure of separation ($M_{i,j}$) is denoted as in Equation (8) and indicates the distance between centroid A_i of the i -th cluster and centroid A_j of the j -th cluster. Finally, the DB index is obtained by calculating the average ratio of separation measure and the scatter measure for all N clusters. Table 2 shows the results of measuring the DB index by increasing the number of clusters from 40 to 160 (the lower the better). As a result, the best performance of the DB index is 2.65 when using 120 clusters ($k = 120$).

Table 2. Comparison of the DB index according to the number of clusters.

#Clusters	40	80	120	160
DB index	2.68	2.67	2.65	2.67

Next, we extracted the Gra-MFCC-Tree-120 units of Table 1(d), mimicking the method proposed in a previous study [11]. First, we trained the CD grapheme-based HMM/GMM systems to derive the phoneme-like units. Then, the phoneme-like units were derived through likelihood-based decision tree-based clustering by using a single HMM state and 39-dimensional Mel-frequency cepstral coefficient (MFCC) feature vectors ($c_{0.12}$ + first derivative + second derivative) per frame. Here, the previous study differs from our method in that the previous study’s statistical table for building the decision tree was generated with the acoustic features obtained from each frame. Our method generates the statistical table using segment-level fixed-length vectors unlike the previous method [11] that used frame-level feature vectors.

We next build the Pho-MFCC-Tree-120 units in Table 1(e). We aim to build subword units suitable for spontaneous speech from phoneme units. On the other hand, the previous study aimed to build phoneme-like units from grapheme units. We can simply build a subword unit by using the CD

phoneme units instead of the CD grapheme units used in the previous study [11]. The decision tree clustering method, used to build these new units, may be slightly different from the method used in the previous study [11]. Note that the proposed method automatically generates questions by top-down binary data clustering, which means that it is not necessary to supply hand-generated questions, unlike in previous studies [11].

Finally, we build the Pho-LTV-Tree-120 units of Table 1(f) by using the method proposed in Section 3. The proposed unit is obtained by using the fixed-length vector extraction method in a previous study [9] and the decision tree-based clustering method also used in a previous study [11]. As a result, we upgraded k-means clustering (applied to the unit extension method used in the previous study [9]) to decision tree-based clustering.

4.3. Results of the Grapheme and Phoneme Units

We first compare the performance of the lexicons consisting of 52 graphemes and 40 phonemes. Table 3 shows the grapheme-based and the phoneme-based speech recognition performance of each evaluation dataset. Here, CI-GMM-HMM is an acoustic model (AM) using a context-independent (CI) unit, and CD-GMM-HMM is an acoustic model using CD units obtained by applying an implicit method [4]. CD-LSTM-HMM is an acoustic model using the LSTM classifier instead of GMM in CD-GMM-HMM. In experiments with the CI unit, the AM consists of a different number of parameters because each unit has a different number. However, in experiments with the CD unit, the AM has 8,033 equal parameters for each unit.

Table 3. Word error rate (%) of the grapheme unit and the phoneme unit.

Units	AM	News	Web	SPT
Grapheme-52	CI-GMM-HMM	39.6	59.1	92.9
	CD-GMM-HMM	23.4	35.3	74.9
	CD-LSTM-HMM	17.7	22.1	53.6
Phoneme-40	CI-GMM-HMM	36.8	57.1	92.2
	CD-GMM-HMM	22.1	34.2	74.6
	CD-LSTM-HMM	15.7	20.3	52.6

In the News, Web, and SPT evaluation data, the phoneme unit performs 11.3%, 8.1%, and 1.9% better than the grapheme unit. Compared to other languages [19], there is little difference between Korean graphemes and phonemes. This is because Korean is a phonogramic language that basically represents a speech signal as a symbol. On the other hand, in the SPT evaluation data using spontaneous speech, the phoneme unit has similar performance to the grapheme unit since the inter-unit distance becomes close and the intra-unit variance increases [7].

4.4. Results of Acoustic Data-Driven Subword Units

We first confirmed the performance of the Pho-LTV-Kmeans units obtained by using the previous method [9]. Table 4 shows the performance of the units after increasing the number of clusters from 40 to 160. As a result, we confirmed the best performance in 120 clusters, as expected in the DB index comparison experiment. In the News, Web, and SPT evaluation data, the Pho-LTV-Kmeans-120 unit showed 15.9%, 8.9%, and 7.0% better performance than the phoneme unit. The performance presented in Table 4 has a higher overall WER than the performance presented in the previous study [9]. This is because, here, the number of decoding units in the language model is 1.8 million words, which is much larger than the 81 thousand words used in the previous study [9].

Table 4. Word error rate (%) of the Pho-LTV-Kmeans units according the number of clusters.

#Clusters	40	80	120	160
News	14.4	14.1	13.2	13.7
Web	20.2	19.0	18.5	18.9
SPT	52.9	51.0	48.9	50.0

Table 5 shows the performance of the subword units derived from each method. Here, the number of subword units derived by each method was 120, which presented the best performance in the Pho-LTV-Kmeans experiments. As argued in a previous study [11], the Gra-MFCC-Tree-120 offers significantly better performance improvement than the grapheme unit. In addition, we observed that units derived from large amounts of speech data can achieve similar performance to a phoneme unit. In a previous study [11], the phoneme-like units performed more poorly than the phoneme units, but in our experiments, the Gra-MFCC-Tree-120 performed better than the phoneme units. This is due to our use of an LSTM-based AM, which was trained by about 1000 h of speech data, unlike the previous study, which used a GMM-based AM, trained by a small amounts of speech data. On the other hand, the Pho-MFCC-Tree-120, which is derived from the phoneme unit rather than the grapheme unit, showed higher performance than the Gra-MFCC-Tree-120. However, the Pho-MFCC-Tree-120 unit showed lower performance than the Pho-LTV-Kmeans-120 unit obtained by using segment-level latent vectors instead of the frame-level feature vectors.

Table 5. Word error rate (%) of the subword units obtained by various methods.

Units	AM	News	Web	SPT
Pho-LTV-Kmeans-120 [9]	CI-GMM-HMM	37.5	54.7	90.0
	CD-GMM-HMM	22.3	33.6	73.9
	CD-LSTM-HMM	13.2	18.5	48.9
Gra-MFCC-Tree-120 [11]	CI-GMM-HMM	35.5	54.4	89.7
	CD-GMM-HMM	22.9	34.2	73.4
	CD-LSTM-HMM	15.6	20.0	50.2
Pho-MFCC-Tree-120	CI-GMM-HMM	34.4	51.4	88.7
	CD-GMM-HMM	21.1	32.6	72.7
	CD-LSTM-HMM	13.7	18.6	49.3
Pho-LTV-Tree-120 (proposed)	CI-GMM-HMM	33.6	51.2	88.4
	CD-GMM-HMM	20.7	32.2	72.3
	CD-LSTM-HMM	12.9	18.0	47.9

We finally confirmed the performance of the Pho-LTV-Tree-120 derived by using the proposed method. The proposed unit showed a relative performance improvement of 17.8% in the News evaluation data—similar to the AM training data—and a significant improvement of 11.1% and 8.9% in the Web and SPT evaluation data with different domains. Moreover, the proposed unit showed higher performance than the subword units derived by other methods. This is because, unlike previous studies [9,11], we used segment-level vectors instead of frame-level vectors as samples for clustering and performed the decision tree-based clustering instead of k-means clustering. Table 5 shows the performance difference between the latent vectors and the feature vectors by comparing the performance between Pho-LTV-Tree-120 and Pho-MFCC-Tree-120. We can also observe the performance difference between k-means clustering and decision tree-based clustering by comparing the performance between Pho-LTV-Kmeans-120 and Pho-LTV-Tree-120.

In the matched-pairs sentence segment word-error test using the National Institute of Standards and Technology Scoring Toolkit (SCTK) [36], the proposed method yielded a p-value less than 0.001 compared with other methods, including the ‘Pho-LTV-Kmeans-120’ method [9]. This justifies the statistical significance of the proposed method. As a result, we confirmed that the subword unit,

obtained by the proposed method, is more suitable for spontaneous speech recognition than the phoneme unit, in order to handle the acoustically low discrimination problem in spontaneous speech.

The proposed method can also be applied to other languages. We confirmed that the proposed unit shows the best performance in Korean spontaneous speech. In Korean, Hangul characters are closely related to their phonetic realizations; a similar language in this regard is Spanish. In other words, Korean grapheme-to-phoneme mapping is close to one-to-one. On the other hand, some languages have a poor correspondence between their graphemes and phonemes, such as English [19]. If the lexicon of the target language exists, we can build an acoustically discriminating subword unit using the proposed method. In addition, even if the lexicon does not exist, the proposed method can be utilized to build a phoneme-like unit from the grapheme units, similar to the previous study [11].

5. Conclusions

This paper proposed a method for extending a phoneme set using a large amount of broadcast data to improve the performance of a Korean spontaneous speech recognizer. In this process, we described a method to extract the fixed-length latent vectors from the variable-length phoneme-level segments, as well as a method to derive subword units using decision tree-based clustering. As a result, the proposed unit shows about 17.8% relative performance improvement over the phoneme unit in the News data. Here, the News evaluation data consist of read speech and are similar to the speech data used for the subword unit derivation and the acoustic model training. On the other hand, in the Web data consisting of read speech and different domains, the proposed unit showed a performance improvement of about 11.1%. Finally, in the SPT data consisting of spontaneous speech, the proposed unit showed a performance improvement of about 11.1%. As a result, we confirmed that the proposed unit is more suitable for spontaneous speech recognition than the phoneme unit with the acoustically low discrimination problem in spontaneous speech.

We expect that the proposed method for deriving new acoustic units from speech data will contribute to building a decoding unit with high discrimination. To do this, we plan to use all the broadcast speech data to train the acoustic model and verify the performance of the subword unit, derived from the grapheme unit, rather than the phoneme unit. We then plan to use the proposed unit as a decoding unit for end-to-end speech recognition systems, after which we will compare our results with those of other decoding units, such as byte-pair encoding and word piece model, to confirm the performance of the spontaneous speech system. In addition, although this study focused on subword units, in the future, we plan to study how speech recognition performance can be improved by applying discriminative training to the proposed unit.

Author Contributions: Data curation, J.-U.B. and S.-H.K.; conceptualization, J.-U.B.; Methodology, J.-U.B. and O.-W.K.; writing—original draft preparation, J.-U.B.; resources, S.-H.K.; writing—review and editing, O.-W.K.; supervision, O.-W.K.; all of the authors participated in the project. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: 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.

References

1. Livescu, K.; Fosler-Lussier, E.; Metze, F. Subword modeling for automatic speech recognition: Past, present, and emerging approaches. *IEEE Signal Process. Mag.* **2012**, *29*, 44–57. [[CrossRef](#)]
2. Chomsky, N.; Halle, M. *The Sound Pattern of English*; Harper and Row: New York, NY, USA; Evanston, IL, USA; London, UK, 1968; pp. 12–19.

3. Sainath, T.N.; Prabhavalkar, R.; Kumar, S.; Lee, S.; Kannan, A.; Rybach, D.; Schoglo, V.; Nguyen, P.; Li, B.; Wu, Y.; et al. No need for a lexicon? Evaluating the value of the pronunciation lexica in end-to-end models. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5859–5863.
4. Young, S.J.; Odell, J.J.; Woodland, P.C. Tree-based state tying for high accuracy acoustic modelling. In Proceedings of the Workshop on Human Language Technology, Plainsboro, NJ, USA, 8–11 March 1994; pp. 307–312.
5. Hain, T. Implicit modelling of pronunciation variation in automatic speech recognition. *Speech Commun.* **2005**, *46*, 171–188. [[CrossRef](#)]
6. Lee, K.-N.; Chung, M. Modeling cross-morpheme pronunciation variations for Korean large vocabulary continuous speech recognition. In Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH), Geneva, Switzerland, 1–4 September 2003; pp. 261–264.
7. Nakamura, M.; Iwano, K.; Furui, S. Differences between acoustic characteristics of spontaneous and read speech and their effects on speech recognition performance. *Comput. Speech Lang.* **2008**, *22*, 171–184. [[CrossRef](#)]
8. Nakamura, M.; Furui, S.; Iwano, K. Acoustic and linguistic characterization of spontaneous speech. In Proceedings of the Workshop on Speech Recognition and Intrinsic Variation (SRIV), Toulouse, France, 20 May 2006; pp. 3–8.
9. Bang, J.-U.; Choi, M.-Y.; Kim, S.-H.; Kwon, O.-W. Extending an acoustic data-driven phone set for spontaneous speech recognition. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), Graz, Austria, 15–19 September 2019; pp. 4405–4409.
10. Ljolje, A. High accuracy phone recognition using context clustering and quasi-triphonic models. *Comput. Speech Lang.* **1994**, *8*, 129–151. [[CrossRef](#)]
11. Razavi, M.; Rasipuram, R.; Doss, M.M. Towards weakly supervised acoustic subword unit discovery and lexicon development using hidden Markov models. *Speech Commun.* **2018**, *96*, 168–183. [[CrossRef](#)]
12. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 18–21 June 1967; pp. 281–297.
13. Lee, C.H.; Soong, F.K.; Juang, B.H. A segment model based approach to speech recognition. In Proceedings of the 1988 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New York, NY, USA, 11–14 April 1988; pp. 501–541.
14. Svendsen, T.; Paliwal, K.K.; Harborg, E.; Husoy, P.O. An improved sub-word based speech recognizer. In Proceedings of the 1989 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Glasgow, Scotland, 23–26 May 1989; pp. 108–111.
15. Paliwal, K.K. Lexicon-building methods for an acoustic sub-word based speech recognizer. In Proceedings of the 1990 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Albuquerque, NM, USA, 3–6 April 1990; pp. 729–732.
16. Bacchiani, M.; Ostendorf, M. Using automatically-derived acoustic sub-word units in large vocabulary speech recognition. In Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP), Sydney, Australia, 30 November–4 December 1998; pp. 1843–1846.
17. Holter, T.; Svendsen, T. Combined optimisation of baseforms and model parameters in speech recognition based on acoustic subword units. In Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Santa Barbara, CA, USA, 14–17 December 1997; pp. 199–206.
18. Kanthak, S.; Ney, H. Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition. In Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Orlando, FL, USA, 13–17 May 2002; pp. 845–848.
19. Killer, M.; Stuker, S.; Schultz, T. Grapheme based speech recognition. In Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH), Geneva, Switzerland, 1–4 September 2003; pp. 3141–3144.
20. Dines, J.; Doss, M.M. A study of phoneme and grapheme based context-dependent ASR systems. In Proceedings of the International Workshop on Machine Learning for Multimodal Interaction (MLMI), Brno, Czech Republic, 28–30 June 2007; pp. 215–226.

21. Magimai-Doss, M.; Rasipuram, R.; Aradilla, G.; Bourlard, H. Grapheme-based automatic speech recognition using KL-HMM. In Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH), Florence, Italy, 27–31 August 2011; pp. 445–448.
22. Rasipuram, R.; Magimai-Doss, M. Acoustic and lexical resource constrained ASR using language-independent acoustic model and language-dependent probabilistic lexical model. *Speech Commun.* **2015**, *68*, 23–40. [[CrossRef](#)]
23. Young, S.; Kershaw, D.; Odell, J.; Ollason, D.; Valtchev, V.; Woodland, P. *The HTK Book Version 3.0*; Cambridge University Press: Cambridge, UK, 2000.
24. Ljolje, A.; Riley, M. Automatic segmentation and labeling of speech. In Proceedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Washington, DC, USA, 19–22 April 1991; pp. 473–476.
25. Bang, J.-U.; Choi, M.-Y.; Kim, S.-H.; Kwon, O.-W. Improving speech recognizers by refining broadcast data with inaccurate subtitle timestamps. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, Sweden, 20–24 August 2017; pp. 2929–2933.
26. Chung, Y.A.; Wu, C.C.; Shen, C.H.; Lee, H.Y.; Lee, L.S. Audio Word2Vec: Unsupervised Learning of audio segment representations using sequence-to-sequence autoencoder. In Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH), San Francisco, CA, USA, 8–12 September 2016; pp. 410–415.
27. Levin, K.; Henry, K.; Jansen, A.; Livescu, K. Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Olomouc, Czech Republic, 8–13 December 2013; pp. 410–415.
28. Mitra, V.; Vergyri, D.; Franco, H. Unsupervised Learning of acoustic units using autoencoders and Kohonen nets. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), San Francisco, CA, USA, 8–12 September 2016; pp. 1300–1304.
29. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The Kaldi speech recognition toolkit. In Proceedings of the 2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Waikoloa, HI, USA, 11–15 December 2011.
30. 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.
31. Bang, J.-U.; Choi, M.-Y.; Kim, S.-H.; Kwon, O.-W. Automatic construction of a large-scale speech recognition database using multi-genre broadcast data with inaccurate subtitle timestamps. *IEICE Trans. Inf. Syst.* **2020**, *103*, 406–415. [[CrossRef](#)]
32. Lamel, L.; Gauvain, J.L.; Adda, G. Lightly supervised and unsupervised acoustic model training. *Comput. Speech Lang.* **2002**, *16*, 115–129. [[CrossRef](#)]
33. Kwon, O.-W.; Park, J. Korean large vocabulary continuous speech recognition with morpheme-based recognition units. *Speech Commun.* **2003**, *39*, 287–300. [[CrossRef](#)]
34. Kim, J.-M. Computer codes for Korean sounds: K-SAMPA. *J. Acoust. Soc. Korea* **2001**, *20*, 1–14.
35. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *1*, 224–227. [[CrossRef](#)] [[PubMed](#)]
36. Gillick, L.; Cox, S.J. Some statistical issues in the comparison of speech recognition algorithms. In Proceedings of the 1989 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Glasgow, UK, 23–26 May 1989; pp. 532–535.