applied sciences

MDPI

Article

Accented Speech Recognition Based on End-to-End Domain Adversarial Training of Neural Networks

Hyeong-Ju Na [1] and Jeong-Sik Park [2,*]

[1] Department of English Linguistics, Hankuk University of Foreign Studies, Seoul 02450, Korea; skgudwn34@gmail.com
[2] Department of English Linguistics and Language Technology, Hankuk University of Foreign Studies, Seoul 02450, Korea
* Correspondence: parkjs@hufs.ac.kr; Tel.: +82-02-2173-8814

Featured Application: This research can be applied for automatic speech recognition systems that handle input speech with two or more accents.

Abstract: The performance of automatic speech recognition (ASR) may be degraded when accented speech is recognized because the speech has some linguistic differences from standard speech. Conventional accented speech recognition studies have utilized the accent embedding method, in which the accent embedding features are directly fed into the ASR network. Although the method improves the performance of accented speech recognition, it has some restrictions, such as increasing the computational costs. This study proposes an efficient method of training the ASR model for accented speech in a domain adversarial way based on the Domain Adversarial Neural Network (DANN). The DANN plays a role as a domain adaptation in which the training data and test data have different distributions. Thus, our approach is expected to construct a reliable ASR model for accented speech by reducing the distribution differences between accented speech and standard speech. DANN has three sub-networks: the feature extractor, the domain classifier, and the label predictor. To adjust the DANN for accented speech recognition, we constructed these three sub-networks independently, considering the characteristics of accented speech. In particular, we used an end-to-end framework based on Connectionist Temporal Classification (CTC) to develop the label predictor, a very important module that directly affects ASR results. To verify the efficiency of the proposed approach, we conducted several experiments of accented speech recognition for four English accents including Australian, Canadian, British (England), and Indian accents. The experimental results showed that the proposed DANN-based model outperformed the baseline model for all accents, indicating that the end-to-end domain adversarial training effectively reduced the distribution differences between accented speech and standard speech.

Keywords: accented speech recognition; speech recognition; end-to-end domain adversarial training; domain adversarial neural network; domain adaptation; connectionist temporal classification

Citation: Na, H.-J.; Park, J.-S. Accented Speech Recognition Based on End-to-End Domain Adversarial Training of Neural Networks. *Appl. Sci.* **2021**, *11*, 8412. https://doi.org/10.3390/app11188412

Academic Editor: Antonio Fernández-Caballero

Received: 17 August 2021
Accepted: 9 September 2021
Published: 10 September 2021

1. Introduction

The performance of automatic speech recognition (ASR) has been continuously improved because of neural network-based technological developments [1,2]. However, ASR performance may be considerably reduced when recognizing abnormal speech such as noisy, emotional, or accented speech. In particular, accented speech is very difficult to recognize because it has some linguistic differences from standard speech in terms of phonetic, morphological, and syntactic differences. Thus, many studies have proposed some methods to improve the performance of accented speech recognition.

The initial approaches for accented speech recognition were focused on adaptations from standard speech to accented speech. Several adaptation techniques such as maximum

a posteriori (MAP) [3] and maximum likelihood linear regression (MLLR) [4] were mainly used up to the early 2010s.

In recent years, neural network-based approaches have been widely used for accented speech recognition. The most representative approach is accent embedding, in which accent embedding features are directly fed into the ASR model. The authors of [5] built a standalone network and a multi-task learning network to perform ASR and accent classification simultaneously. The standalone network classified the type of accent and produced frame-level accent embedding features. The output features of the standalone network were then fed into the multi-task learning network as extra input feature vectors. The authors of [6] advanced the approach by applying an end-to-end method to the ASR model. In the real world, it is difficult to collect a large amount of accented speech data to train deep neural networks, which is one of the challenges of accented speech recognition. The domain adaptation technique can be a solution to this problem [7]. This technique aims to enhance less-trained target domain models that are constructed with a small amount of target domain data by utilizing the information learned from the source domain data that can be collected abundantly. In accented speech recognition, standard speech can be regarded as the source domain, whereas accented speech is considered as the target domain. This study applied the Domain Adversarial Neural Network (DANN) as a domain adaptation technique for accented speech recognition. It has been widely used for computer vision studies [8]. The goal of DANN is to learn domain-invariant features to reduce the distribution differences between the source and target domains with the help of domain adversarial training. In contrast to accent embedding, DANN does not require extra input features because the ASR model can be trained using only filter-bank features, which are the basic features of ASR. In addition, this study proposed an end-to-end domain adversarial training framework for accented speech recognition.

This study is organized as follows. Conventional accented speech recognition studies and their drawbacks are discussed in Section 2. In Section 3, the main method proposed in this study is explained. In Section 4, the experimental setup and results are reported. Finally, the conclusions and further research work are discussed in Section 5.

2. Conventional Studies on Accented Speech Recognition

Up to the early 2010s, several adaptation techniques such as MAP and MLLR were mainly adopted for accented speech recognition [3,4]. These techniques aimed to adapt the standard ASR model to accented speech data. Although the approaches successfully improved accented speech recognition performance, they operated on statistical pattern models such as the Gaussian Mixture Model (GMM)–Hidden Markov Model (HMM), which is a classic approach [9–11]. Thus, they have rarely been used in current studies that focus on neural network modeling.

The authors of [5] proposed a multi-task learning network-based approach that performs ASR and accent classification simultaneously. They built a standalone network to produce accent embedding features, which were fed into the multi-task learning network, expecting that asking (accent classification) and telling (accent embedding) the network would improve the performance. In [6], the authors improved this approach by changing the inner structure of the ASR model from two time delay neural network (TDNN) [12] layers to a combination of two convolutional neural network (CNN) layers and four gated recurrent unit (GRU) [13] layers.

The conventional approaches based on accent embedding build a multi-task learning network and a standalone network. Figure 1 describes the model architecture proposed in [5]. A multi-task learning network has two sub-networks ((A) and (B)), and each sub-network performs different functions. As shown in this figure, the acoustic model (A) is combined with the accent classifier (B), and these sub-networks share two TDNN layers. Based on two sub-networks, the multi-task learning network performs phone recognition and accent classification simultaneously. Next, the standalone network (C) produces accent embedding features, which are fed into the multi-task learning network. Thus,

mel-frequency cepstral coefficients, i-vectors, and accent embedding features are used as input features in this approach.

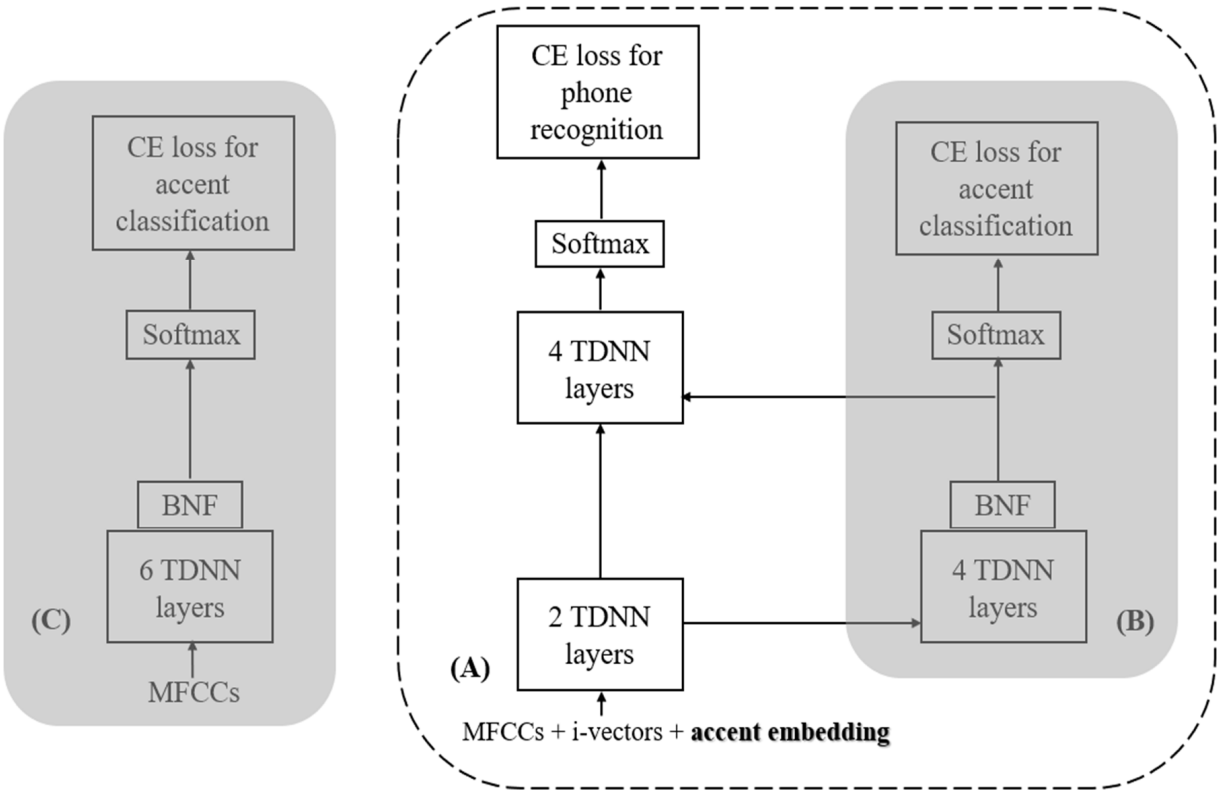

Figure 1. Architecture of the multi-task learning network and the standalone network in the accent embedding method. (**A**) Acoustic model; (**B**) Accent classifier; (**C**) Standalone network.

The accent embedding method is helpful for improving ASR performance; however, it has some limitations. To output the accent embedding features, the standalone network needs to be built and trained independently. Moreover, the standalone network should be constructed sophisticatedly, as ASR performance is considerably affected by the accent classification accuracy of the network.

Accent classification accuracy has been reported according to the model specifications, as shown in Table 1. When the model has more nodes, the accuracy becomes higher. In other words, to achieve stable classification accuracy, a complex model that has high computational costs and large memory space is required.

Table 1. Accent classification accuracy of a standalone network.

Model Specifications	Accent Classification Accuracy
TDNN, 7 layers, 100-d nodes	78.4%
TDNN, 7 layers, 200-d nodes	78.4%
TDNN, 7 layers, 300-d nodes	80.0%
TDNN, 7 layers, 1024-d nodes	82.6%

Furthermore, a number of parameters should be estimated in the multi-task learning network because additional feature vectors are added into the network. To accommodate these parameters, the network should be complex, similar to the standalone network.

3. Accented Speech Recognition Based on End-to-End Domain Adversarial Training

3.1. Domain Adaptation for Accented Speech Recognition

Deep neural networks require a large number of training data to construct the ASR model. In standard speech recognition tasks, a lot of normal speech data are available, while accented speech data are difficult to collect. Thus, it is difficult to construct the ASR model for accented speech recognition using only accented speech data. Furthermore, fully depending on standard speech data (i.e., non-accented data) for model training may degrade the ASR accuracy when accented speech is inputted, as the acoustic characteristics of standard speech data are quite different from those of accented speech data. Domain adaptation can be a solution for this problem.

Domain adaptation is a technique used to train models when the training and test data domains are mismatched [7]. This technique utilizes the well-organized information provided by the source domain data to handle the target domain data. Thus, it aims to obtain models that retain the characteristics of the target domain data from source domain models that are well trained with a large number of source domain data. With the help of domain adaptation, the distribution differences between the source and target domains are reduced. Figure 2 shows the effect of domain adaptation. Through the domain adaptation, the distance between the source and target domains becomes closer. In other words, the districts of the two domains are almost integrated.

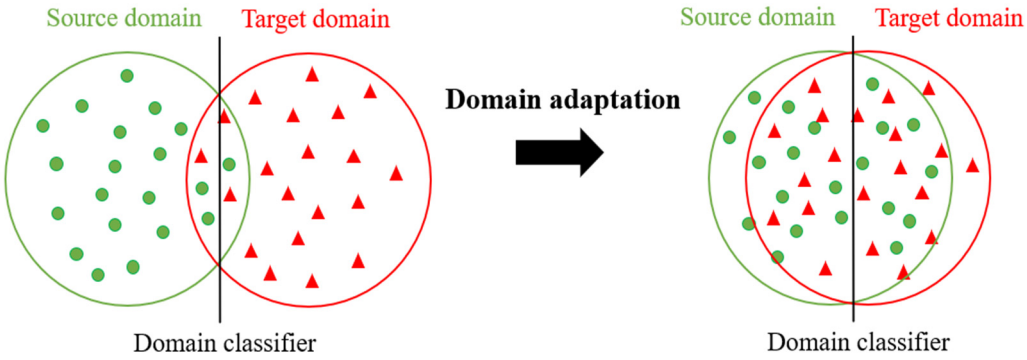

Figure 2. Effect of domain adaptation.

In accented speech recognition, standard speech data can be regarded as the source domain data, whereas accented speech data are considered as the target domain data. The main idea of this study was to construct a reliable model for accented speech recognition with a small number of accented speech data from a standard ASR model by using the domain adaptation technique.

3.2. Domain Adversarial Neural Network

A representative domain adaptation technique in neural network-based model training is DANN, which was first proposed in the field of computer vision [8]. Some speech recognition studies then adopted the technique to handle noisy speech data [14]. Compared with accent embedding, which was addressed in Section 2, DANN requires a lower computational cost, as it can be trained using only filter-bank features without the generation of extra features. In particular, DANN performs more sophisticated training procedures consisting of feature learning, domain adaptation, and label prediction, each of which is jointly unified within the architecture. Figure 3 shows the architecture of DANN, which comprises three sub-networks: the feature extractor, the domain classifier, and the label predictor. In this figure, the green-colored, red-colored, and blue-colored parts are the feature extractor $G_f(x;\theta_f)$, the domain classifier $G_d(f;\theta_d)$, and the label predictor $G_y(f;\theta_y)$, respectively.

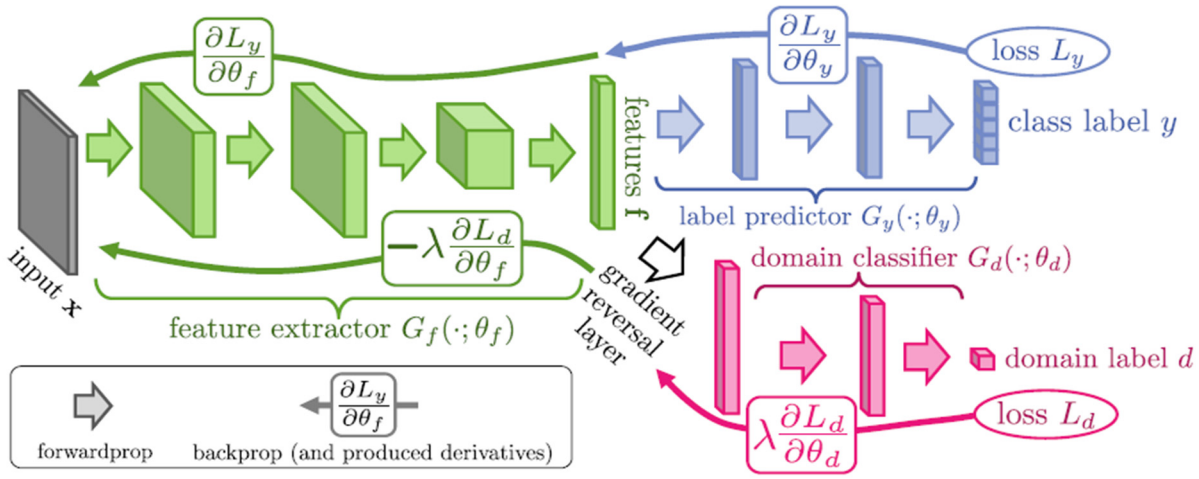

Figure 3. Architecture of the domain adversarial neural network.

First, the feature extractor aims to extract some useful features from the input data x to train the model. These features are then forwarded to the domain classifier and the label predictor. According to the backpropagation mechanism, the extractor reflects the gradient of the domain classifier $\frac{\vartheta L_d}{\vartheta \theta_d}$ and the label predictor $\frac{\vartheta L_y}{\vartheta \theta_y}$. Afterward, the extractor adjusts the features to be domain-invariant in a number of training steps.

The domain classifier determines whether the domain of the input data is the source domain or the target domain. The goal of DANN is to reduce the distribution differences between the source and target domains, and it is achieved by making it difficult to distinguish between the source and target domains. The gradient reversal layer (GRL) plays an important role in this step. The GRL is positioned at the bottom of the domain classifier and reverses the gradient $\frac{\vartheta L_d}{\vartheta \theta_d}$ by multiplying it by a scalar λ during backpropagation. With the help of the GRL, the feature extractor receives the reversed gradient $-\lambda\frac{\vartheta L_d}{\vartheta \theta_f}$. Thus, the feature extractor can make the features become domain-invariant in the subsequent training steps. The GRL is only activated during backpropagation and it does not change any parameters during forward propagation.

Lastly, the label predictor predicts a label $\hat{y}$ and calculates the loss value between the predicted label and the original label y. The gradient $\frac{\vartheta L_y}{\vartheta \theta_y}$ is also backpropagated into the bottom layers of the model. In summary, DANN has two main objects: to predict the label correctly and to reduce the distribution between the source and target domains. These objects are achieved by minimizing the loss of the label predictor (given by (1)) and maximizing the loss of the domain classifier (given by (2)).

$$\theta_y = argmin_{\theta_y} E\left(\theta_f, \theta_y, \theta_d\right) \tag{1}$$

$$\theta_d = argmax_{\theta_d} E\left(\theta_f, \theta_y, \theta_d\right) \tag{2}$$

3.3. End-to-End Domain Adversarial Training Based on DANN for Accented Speech Recognition

This study proposes an efficient accented speech recognition approach using DANN to handle domain adaptation. In particular, we propose an end-to-end domain adversarial training framework targeting accented speech recognition. Figure 4 illustrates the model training framework proposed in this study. In the figure, the green-colored, red-colored, and blue-colored parts are the feature extractor, the domain classifier, and the label predictor, respectively.

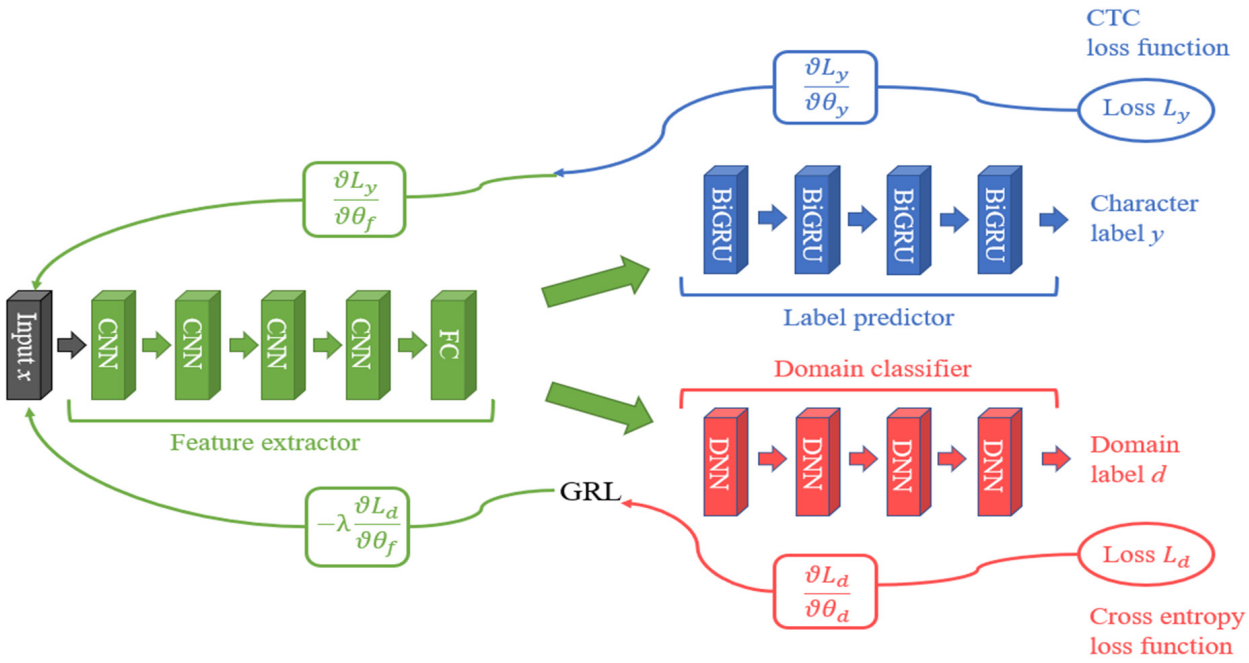

Figure 4. Model training framework for end-to-end domain adversarial training based on DANN.

3.3.1. Feature Extractor

We constructed the feature extractor using CNN, which is a useful neural network for learning two-dimensional data while maintaining information. The general features used in ASR are two-dimensional mel-spectrograms. Thus, in this study, mel-spectrograms extracted from raw speech data were used as input features, then the features were used to train four CNN layers. When the mel-spectrogram features were fed into the feature extractor, a filter skimmed through the features in one stride. The filter moved forward step-by-step and produced a feature map that had the weights of the parameters calculated by the convolution between the mel-spectrogram features and the filter.

The simplified procedure of feature mapping is illustrated in Figure 5. If the CNN has many layers, its output size is drastically reduced because each layer receives the feature map of the previous layer and performs convolution again. To prevent drastic size reduction, zeros are padded into the feature map to maintain the size of the feature map. For each convolutional layer, the Gaussian error linear unit [15] is used as an activation function to receive the parameter weights and produce output. After passing through the four CNN layers, one fully connected layer changes the final two-dimensional CNN outputs into one-dimensional feature vectors. These converted features are fed into the domain classifier and the label predictor. Meanwhile, the feature extractor adjusts the features to be domain-invariant in a number of training steps. Thus, it finally extracts useful features characterizing the target accented speech.

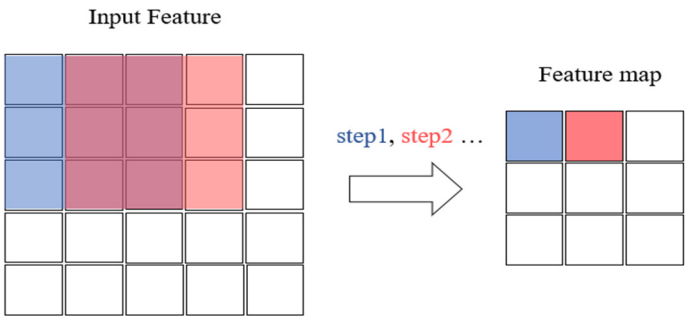

Figure 5. Simplified procedure of CNN feature mapping.

3.3.2. Domain Classifier

The domain classifier plays a role in deciding whether the input data are the target (accented speech) or not. This binary classification problem does not require complex algorithms; hence, we used a simple deep neural network (DNN) architecture with four layers to construct the domain classifier, with the expectation that it would provide sufficient conditions for binary classification. The features learned from the feature extractor passed through four DNN layers, then the domain label d was outputted from the domain classifier. Afterward, the domain loss L_d was calculated by the difference between the predicted domain and original domain using the cross-entropy loss function. The gradient $\frac{\vartheta L_d}{\vartheta \theta_d}$ was backpropagated into the downstream of the model, and the GRL reversed the gradient and multiplied it by a scalar λ, delivering the reversed gradient $-\lambda \frac{\vartheta L_d}{\vartheta \theta_f}$ to the feature extractor. Thus, the feature extractor could learn accent-invariant features in the next training steps.

3.3.3. Label Predictor

The label predictor is a very important module, as it actually recognizes the accented speech data. This module aims to predict the character label among 28 labels (26 English characters <A–Z>, <apostrophe>, and <space>) for features forwarded from the feature extractor. The correctness of predicted labels directly affects the ASR results. In order to simplify the label prediction procedures and improve the accuracy, we used the end-to-end framework.

There are two end-to-end ASR approaches: connectionist Temporal Classification (CTC) [16] and Listen, Attend and Spell (LAS) [17]. Of these two approaches, we used CTC because of two reasons. First, CTC can be trained more efficiently and conveniently than LAS. CTC is a kind of loss function, and no weights are required to train it. Meanwhile, LAS requires weights to train a joint model comprising the encoder, the attention mechanism, and the decoder. Second, LAS has a long delay in decoding the output sequence, as it does not proceed to the alignment and decoding processes until all the encoding results have been generated. Thus, LAS is not suitable for streaming applications. In contrast, CTC has a short delay because it can begin decoding the output sequence once an encoding result has been received. For these reasons, this study concentrates on CTC for constructing the label predictor.

The label loss L_y is calculated by the difference between the predicted label and the original label using the CTC loss function. The gradient $\frac{\vartheta L_y}{\vartheta \theta_y}$ is then backpropagated downstream from the model. In the next training step, the feature extractor uses the gradient information to correctly predict the label. Unlike the domain classifier, we constructed the label predictor using bi-directional GRU (BiGRU) layers because recognizing speech requires a more complex algorithm than classifying the domain.

The Recurrent Neural Network (RNN) is known as useful model for handling time series data such as speech data, but it has a drawback. When the time step becomes longer, RNN cannot remember the information inputted a long time ago, as long-term information fades so that recent information can be learned. In this case, the model cannot be perfectly trained for long time series data and it cannot correctly output the front parts of the label sequence. To solve this problem, some advanced RNN models have been introduced. The most famous models are LSTM [18] and GRU [13]. Both LSTM and GRU can remember long-term memory, but they have a difference in complexity. GRU has a simpler structure than LSTM, so it requires fewer parameters to train the model. LSTM has two kinds of states: the hidden state and the cell state; GRU combines the two states. In addition, GRU has a smaller number of gates than LSTM. GRU has two gates (reset and update), whereas LSTM requires three gates (forget, input, and output). Thus, this study adopted a GRU-based model called BiGRU to build the label predictor. BiGRU is an advanced GRU that considers the context bi-directionally both from past to future and from future to past, thus enhancing speech recognition accuracy.

4. Experiments and Results

We conducted several experiments to verify the efficiency of the proposed accented speech recognition approach. The experimental setup is described first, then the experimental results are reported.

4.1. Experimental Setups

4.1.1. Speech Corpus

In this study, all experiments were conducted using the Common Voice corpus [19], which is an open-source speech database released by Mozilla in 2019. As this corpus consists of a significant amount of well-refined speech data, it is widely used for various speech recognition tasks [20]. In particular, this corpus can be effectively utilized in the field of accented speech recognition because speaker information such as age, gender, language, and accent type are provided. The Common Voice corpus has speech data for 60 languages ranging from widely used languages such as English and Spanish to relatively unfamiliar languages such as Basque and Welsh. This study targeted English accents, using speech data from five English accents including US (US accent), AU (Australian accent), CA (Canadian accent), EN (British English (England) accent) and IN (Indian accent). Hereafter, the accent names are denoted by their respective abbreviations for convenience. For domain adaptation, US was determined as the source domain, while the other four accents were regarded as the target domains, because the quantity of US data is much larger than that of other accents. All the speech files were pre-processed for the experiments. The downloaded speech files (.mp3) were converted into wav format and sampled at 16,000 Hz. The files were then divided into the training set (Table 2), the validation set (Table 3), and the test set (Table 4).

Table 2. Summary of the training set.

Dataset	Region	Number of Files	Hours
US-160k	US	160,000	196
AU-20k	Australia	20,000	26
CA-20k	Canada	20,000	25
CA-32k	Canada	32,000	43
EN-20k	England	20,000	24
EN-40k	England	40,000	53
EN-60k	England	60,000	77
EN-80k	England	80,000	101
EN-100k	England	100,000	126
IN-20k	India	20,000	26
IN-40k	India	40,000	57

Table 3. Summary of the validation set.

Dataset	Region	Number of Files	Hours
AU-val	Australia	2000	3
CA-val	Canada	2000	3
EN-val	England	2000	3
IN-val	India	2000	3

Table 4. Summary of the test set.

Dataset	Region	Number of Files	Hours
AU-test	Australia	2000	3
CA-test	Canada	2000	3
EN-test	England	2000	3
IN-test	India	2000	3

There were no duplicated files in these three sets. The training set was used to train the ASR model. The names of the datasets in Table 2 indicate the accent type and the number of files. For example, AU-20k indicates an Australian accent dataset with 20,000 files. The validation set was used to calibrate the hyperparameters of the ASR model. It played a role in enhancing the credibility of the experimental results. To obtain reliable experimental results, the hyperparameters needed to be calibrated in detail using the validation set. Finally, the ASR model was evaluated using the test set.

4.1.2. Hyperparameters

To obtain the best experimental results, the hyperparameters were calibrated in detail using the validation set. The hyperparameters were heuristically determined by the system developers, whereas the feature parameters were calculated within the model and they can be determined from the data. In deep learning, the representative hyperparameters include the learning rate, epoch, batch size, and dropout rate.

To find the lowest loss L, the weights W were updated with a learning rate η. The weight w_{t+1} in the next time step was updated by subtracting the multiplication of gradient $\frac{\vartheta L}{\vartheta w}$ and η from the current weight w_t, as described in (3). The weights were optimized by the Adam optimizer [21].

$$w_{t+1} = w_t - \eta \times \frac{\vartheta L}{\vartheta w} \tag{3}$$

Setting an appropriate learning rate is very important, as the learning rate affects variations in the loss value and helps to find the global minimum loss value. After conducting several experiments by calibrating the learning rate, this study determined the most appropriate learning rate to be 0.0001.

Epoch means a count of how often the total training samples have been passed forward and backward through the model. As each epoch proceeds, the loss value is expected to gradually decrease. However, there is a period in which the model does not show any significant improvement. Thus, an appropriate number of epochs needs to be determined when investigating the improvement in performance. This study confirmed that 100 epochs were sufficient to achieve the lowest loss.

Batch size refers to the number of training samples that are sequentially entered into the training stage, and it is generally set to a power of two such as 2, 4, 8, 16, 32, etc. In general, a larger batch size makes the training faster but it requires more memory for calculation. Thus, the batch size should be determined considering the memory capacity of the computer. In this study, the batch size was empirically set to 32. Thus, 32 training samples were in a single batch.

Dropout is a type of regularization technique [22]. It helps to prevent the overfitting problem, which arises when the network excessively fits into the training data and thus fails to correctly recognize test data. Overfitting problem often occurs when the network has a large capacity and the amount of training data is too small to meet the capacity. Hence, this issue needs to be handled in the accented speech recognition task, where a smaller amount of accented speech data have been provided. Dropout deactivates some neurons (nodes) to reduce the capacity of the network, making the network simpler. This operation is performed according to a dropout rate, which is the ratio of the deactivated nodes to all nodes. In this study, the dropout rate was empirically set to 0.1.

The last hyperparameter in this study was a domain adaptation parameter λ used in DANN. When the gradient of the domain classifier $\frac{\vartheta L_d}{\vartheta \theta_d}$ is backpropagated into the downstream of the ASR model, the gradient is multiplied by λ, which can be set between 0 and 1. The parameter controls the influence of domain adversarial training. A higher value makes the effect of the domain adversarial training more dominant. However, an excessive dependence on domain adversarial training may lead to performance degradation. Thus, the parameter should be set precisely via experiments. In this study, λ was empirically set to 0.01.

4.2. Experimental Results

To verify the efficiency of the proposed approach, baseline and DANN models were built. The baseline model performed standard end-to-end speech recognition, and it comprised four CNN and four BiGRU layers, as shown in Figure 6. The DANN model was constructed according to Figure 4, illustrated in Section 3.3.

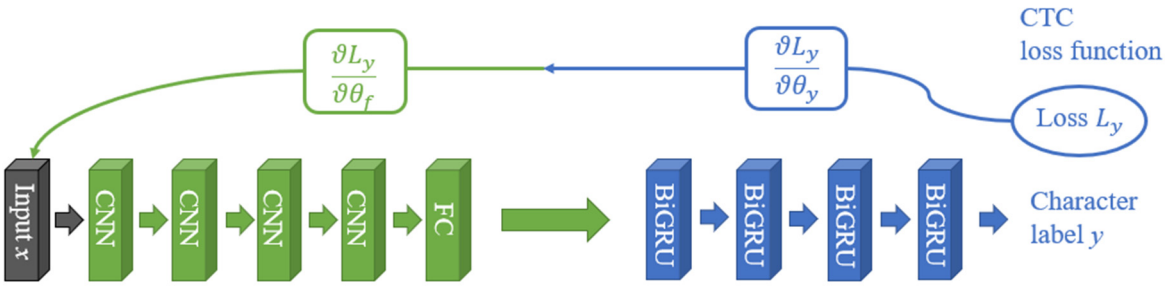

Figure 6. Architecture of the baseline model.

The performance of the two model types was first investigated. As described in Table 5, two baseline models were compared with the DANN model. Baseline-src was trained using only the dataset of the source domain (US-160k), whereas baseline-src-tgt was trained using both the datasets of the source domain (US-160k) and each target domain (AU-20k, CA-20k, EN-20k, and IN-20k). DANN was also trained using the same datasets as baseline-src-tgt. The performance of the models was assessed using two measures: the character error rate (CER) and the word error rate (WER), which were measured by the edit distance based on the Levenshtein algorithm [23] between the predicted labels $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$ and the original labels $Y = \{y_1, y_2, \ldots, y_n\}$.

Table 5. Performance (CER/WER) comparison between the baseline and DANN models.

Accent	Model		
	Baseline-src	**Baseline-src-tgt**	**DANN**
AU	28.85%/65.95%	25.09%/61.06%	24.22%/59.80%
CA	14.76%/43.16%	13.77%/40.68%	13.55%/40.15%
EN	25.49%/61.53%	24.43%/59.78%	21.60%/54.67%
IN	36.41%/76.03%	30.52%/69.41%	28.83%/66.49%

The results are presented in pairs of CER and WER (CER/WER) in Table 5. As shown in the table, baseline-src-tgt achieved a better performance than baseline-src for all accents. Although the amount of target domain data was much smaller than that of the source domain data, the model applying the target domain data was effective for recognizing the test data corresponding to the target domain. Moreover, compared with the two baseline model types, DANN achieved significant performance improvements for all accents. In particular, compared with baseline-src-tgt, the proposed DANN approach showed a notable performance for EN and IN accents, with WER reductions of approximately 5% and 3%, respectively. Meanwhile, it had a very slight performance improvement for the CA accent, with a WER reduction of 0.53%. These results indicated that the linguistic differences between the source and target domain accents affected the performance. The source domain accent (US) has linguistically more similar characteristics to CA than the EN and IN accents. As a result, domain adversarial training provides better conditions for recognizing target accents that are linguistically different from the source domain accents.

Although our proposed approach demonstrated superior performance compared with the baseline, it is necessary to examine whether the results are sufficiently meaningful. In particular, the CER and WER are quite high for practical purposes. In general, standard speech recognition systems providing stable performance use vast amounts of training data. The study in [16], which was based on end-to-end speech recognition, used over 10,000 h

of standard speech data to train acoustic models. However, it is a challenge to construct reliable models for different accent types, as it is difficult to collect a sufficient amount of speech data for each target accent. For this reason, most studies have concentrated on finding efficient methods under the conditions of limited amounts of accented speech data. In this study, we used approximately 200 h of source domain data and 25 h of target domain data to train the models.

Nevertheless, it is necessary to compare the performance of our approach with that of conventional studies. We investigated the results given in [6], in which the accent embedding technique, known as the most representative approach for accented speech recognition, was adopted; the end-to-end method was applied for training acoustic models; and the Common Voice corpus was used for evaluation. We selected the Indian accent for our performance comparison as it was the only target accent for which performance was reported in both studies. In the conventional study, the WER of the Indian accent was 52%, which outperformed our approach (63.56%). However, a direct comparison may not be meaningful, as the two studies conducted experiments on different experimental setups in terms of the training dataset and hyperparameters. For this reason, we concentrated on relative improvements in the baseline system. The baseline performance of the conventional study was 55.2%, providing a relative improvement of 6.15%. On the other hand, our approach achieved a relative improvement of 14.35% compared with the baseline. These results show that our proposed approach improved the baseline system more efficiently with less computational cost compared with the conventional study.

The next experiment was conducted to investigate the effect of the amount of target domain data on performance improvement. The performance was observed when larger amounts of target domain data were applied to the DANN model. The model was called DANN-inc and it was compared with the DANN model (DANN) shown in Table 5. The Common Voice corpus provides different amounts of speech files according to accents. For a fair evaluation, we balanced the amount of data for different accent types when constructing the DANN-inc model. As a result, the amount of data for each target accent was set to about 40k. Table 6 summarizes the results. When more target domain data were used to train the DANN model, the model's performance was significantly improved for most accents. Among the four accent types, the IN accent demonstrated the most significant improvement, with a WER reduction of approximately 3%.

Table 6. Performance (CER/WER) of DANN models by adding more target domain data.

Accent	Model	
	DANN	**DANN-inc**
AU	24.22%/59.80%	22.72%/57.38%
CA	13.55%/40.15%	13.63%/40.02%
EN	21.60%/54.67%	19.96%/52.53%
IN	28.83%/66.49%	26.24%/63.56%

Although Table 6 shows that the amount of target domain data affected the accuracy of DANN, it is difficult to say that the experiments perfectly observe the tendency, due to the limited amount of data for some accents. The final experiment focused on investigating the effect of the amount of target domain data in detail. This experiment was conducted using only the speech files of EN, which is the accent with the largest amount of target domain data among the four target accents. Five models were constructed, ranging from DANN-EN1 to DANN-EN5, while varying the amount of EN accent data (20k, 40k, 60k, 80k, and 100k for EN1, EN2, EN3, EN4, and EN5, respectively). In all the DANN-EN models, the amount of source domain (US) data was 160k. For example, DANN-EN3 was trained with US-160k and EN-60k data. Table 7 presents the performance of the five models. As the amount of target domain data increased, the accuracy of accented speech recognition consistently improved. In particular, DANN-EN5 achieved a WER reduction of about 9% in comparison with DANN-EN1. This result shows the possibility that collecting

a larger amount of target accent data will improve the performance of accented speech recognition, enhancing the correctness of DANN models.

Table 7. Performance (CER/WER) of DANN models according to the amount of EN accent data.

Accent	Model				
	DANN-EN1	**DANN-EN2**	**DANN-EN3**	**DANN-EN4**	**DANN-EN5**
EN	21.60%/54.67%	19.96%/52.53%	18.65%/49.91%	17.15%/47.32%	16.35%/45.68%

5. Conclusions

This study proposed an efficient accented speech recognition approach using end-to-end domain adversarial training of neural networks based on DANN. The goal of DANN is to learn domain-invariant features to reduce the distribution differences between the source and target domains. In this study, we proposed an efficient DANN model architecture to carefully handle accented speech recognition. Each of the three sub-networks of DANN was constructed with appropriate neural networks considering the characteristics of accented speech data. CNN was used for the feature extractor, DNN for the domain classifier, and BiGRU for the label predictor. In particular, we used a CTC-based end-to-end framework to construct the label predictor, which is a very important module in DANN, as the accuracy of the predicted labels directly affects the ASR results.

To verify the efficiency of the proposed approach, we performed several experiments of accented speech recognition using the Common Voice corpus for four English accents (Australian, Canadian, England, and Indian accents). For all accents, the proposed DANN model outperformed the baseline model constructed according to a standard end-to-end speech recognition scheme. In addition, in experiments performed with varying amounts of target accent data, we observed that the accuracy of the proposed model improved significantly as the amount of the target domain data increased.

In a further study, we will investigate an efficient DANN approach for an unsupervised accented speech recognition task in which data labels are not required. We expect that this further study will help to make larger datasets because unlabeled accented speech data are easier to collect. Furthermore, we will utilize the transformer architecture used for end-to-end speech recognition to enhance the end-to-end domain adversarial training. In particular, since the transformer-based decoder is applicable to the label prediction module of DANN, we expect that this further study will greatly improve accented speech recognition performance.

Author Contributions: Conceptualization, H.-J.N. and J.-S.P.; methodology, H.-J.N. and J.-S.P.; software, H.-J.N.; validation, H.-J.N. and J.-S.P.; formal analysis, H.-J.N. and J.-S.P.; writing—original draft preparation, H.-J.N.; writing—review and editing, J.-S.P.; supervision, J.-S.P.; project administration, J.-S.P.; funding acquisition, J.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Hankuk University of Foreign Studies Research Fund, the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2020R1A2C1013162), and the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2016-0-00313) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The speech data used for experiments is publicly available at https://commonvoice.mozilla.org/ (accessed on 16 August 2021).

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

References

1. Oh, D.; Park, J.-S.; Kim, J.-H.; Jang, G.-J. Hierarchical Phoneme Classification for Improved Speech Recognition. *Appl. Sci.* **2021**, *11*, 428. [CrossRef]
2. Lee, D.; Park, J.-S.; Koo, M.-W.; Kim, J.-H. Language Model Using Neural Turing Machine Based on Localized Content-Based Addressing. *Appl. Sci.* **2020**, *10*, 7181. [CrossRef]
3. Gauvain, J.-L.; Lee, C.-H. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. Speech Audio Process.* **1994**, *2*, 291–298. [CrossRef]
4. Leggetter, C.; Woodland, P. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Comput. Speech Lang.* **1995**, *9*, 171–185. [CrossRef]
5. Jain, A.; Upreti, M.; Jyothi, P. Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning. In Proceedings of the Interspeech 2018, Hyderabad, India, 2–6 September 2018; pp. 2454–2458.
6. Viglino, T.; Motlicek, P.; Cernak, M. End-to-End Accented Speech Recognition. In Proceedings of the Interspeech 2019, Graz, Austria, 15–19 September 2019; pp. 2140–2144.
7. Ben-David, S.; Blitzer, J.; Crammer, K.; Pereira, F. Analysis of representations for domain adaptation. In Proceedings of the 19th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2006; pp. 137–144.
8. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *arXiv* **2016**, arXiv:1505.07818.
9. Liu, W.K.; Fung, P.N. MLLR-based accent model adaptation without accented data. In Proceedings of the ICSLP 2000, Beijing, China, 16–20 October 2000.
10. Tomokiyo, L.M.; Waibel, A. Adaptation methods for non-native speech. In Proceedings of the Multilinguality in Spoken Language Processing, Aalborg, Denmark, 8 September 2001.
11. Vergyri, D.; Lamel, L.; Gauvain, J.L. Automatic speech recognition of multiple accented English data. In Proceedings of the Interspeech 2010, Chiba, Japan, 26–30 September; 2010.
12. Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, K. Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech, Signal Process.* **1989**, *37*, 328–339. [CrossRef]
13. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
14. Shinohara, Y. Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition. In Proceedings of the Interspeech 2016, San Francisco, CA, USA, 8–12 September 2016; pp. 2369–2372.
15. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
16. 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 ICML, New York, NY, USA, 19–24 June 2016; pp. 173–182.
17. Chan, W.; Jaitly, N.; Le, Q.V.; Vinyals, O. Listen, attend and spell. *arXiv* **2015**, arXiv:1508.01211.
18. Soltau, H.; Liao, H.; Sak, H. Neural Speech Recognizer: Acoustic-to-Word LSTM Model for Large Vocabulary Speech Recognition. *arXiv* **2016**, arXiv:1610.09975. [CrossRef]
19. Ardila, R.; Branson, M.; Davis, K.; Henretty, M.; Kohler, M.; Meyer, J.; Morais, R.; Saunders, L.; Tyers, F.M.; Weber, G. Common voice: A massively-multilingual speech corpus. *arXiv* **2019**, arXiv:1912.06670.
20. Kim, H.; Park, J.-S. Automatic Language Identification Using Speech Rhythm Features for Multi-Lingual Speech Recognition. *Appl. Sci.* **2020**, *10*, 2225. [CrossRef]
21. Kingma, D.P.; Ba, J.A. A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
22. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* **2014**, *15*, 1929–1958.
23. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.