KIPS Trans. Softw. and Data Eng.
Vol.8, No.3 pp.115~122 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2019.8.3.115

CRNN-Based Korean Phoneme Recognition Model with CTC Algorithm

Hong Yoonseok† · Ki Kyungseo†† · Gweon Gahgene†††

ABSTRACT

For Korean phoneme recognition, Hidden Markov-Gaussian Mixture model(HMM-GMM) or hybrid models which combine artificial neural network with HMM have been mainly used. However, current approach has limitations in that such models require force-aligned corpus training data that is manually annotated by experts. Recently, researchers used neural network based phoneme recognition model which combines recurrent neural network(RNN)-based structure with connectionist temporal classification(CTC) algorithm to overcome the problem of obtaining manually annotated training data. Yet, in terms of implementation, these RNN-based models have another difficulty in that the amount of data gets larger as the structure gets more sophisticated. This problem of large data size is particularly problematic in the Korean language, which lacks refined corpora. In this study, we introduce CTC algorithm that does not require force-alignment to create a Korean phoneme recognition model. Specifically, the phoneme recognition model is based on convolutional neural network(CNN) which requires relatively small amount of data and can be trained faster when compared to RNN based models. We present the results from two different experiments and a resulting best performing phoneme recognition model which distinguishes 49 Korean phonemes. The best performing phoneme recognition model combines CNN with 3hop Bidirectional LSTM with the final Phoneme Error Rate(PER) at 3.26. The PER is a considerable improvement compared to existing Korean phoneme recognition models that report PER ranging from 10 to 12.

Keywords : Phoneme Recognition, CTC Algorithm, Convolutional Neural Network, Recurrent Neural Network

CTC를 적용한 CRNN 기반 한국어 음소인식 모델 연구

홍 윤 석† · 기 경 서†† · 권 가 진†††

요 약

지금까지의 한국어 음소 인식에는 은닉 마르코프-가우시안 믹스처 모델(HMM-GMM)이나 인공신경망-HMM을 결합한 하이브리드 시스템이 주로 사용되어 왔다. 하지만 이 방법은 성능 개선 여지가 적으며, 전문가에 의해 제작된 강제정렬(force-alignment) 코퍼스 없이는 학습이 불가능하다는 단점이 있다. 이 모델의 문제로 인해 타 언어를 대상으로 한 음소 인식 연구에서는 이 단점을 보완하기 위해 순환 신경망(RNN) 계열 구조와 Connectionist Temporal Classification(CTC) 알고리즘을 결합한 신경망 기반 음소 인식 모델이 연구된 바 있다. 그러나 RNN 계열 모델을 학습시키기 위해 많은 음성 말뭉치가 필요하고 구조가 복잡해질 경우 학습이 까다로워, 정제된 말뭉치가 부족하고 기반 연구가 비교적 부족한 한국어의 경우 사용에 제약이 있었다. 이에 본 연구는 강제정렬이 불필요한 CTC 알고리즘을 도입하되, RNN에 비해 더 학습 속도가 빠르고 더 적은 말뭉치로도 학습이 가능한 합성곱 신경망(CNN)을 기반으로 한국어 음소 인식 모델을 구축하여 보고자 시도하였다. 총 2가지의 비교 실험을 통해 본 연구에서는 한국어에 존재하는 49가지의 음소를 판별하는 음소 인식기 모델을 제작하였으며, 실험 결과 최종적으로 선정된 음소 인식 모델은 CNN과 3층의 Bidirectional LSTM을 결합한 구조로, 이 모델의 최종 PER(Phoneme Error Rate)은 3.26으로 나타났다. 이는 한국어 음소 인식 분야에서 보고된 기존 선행 연구들의 PER인 10~12와 비교하면 상당한 성능 향상이라고 할 수 있다.

키워드 : 음소 인식, CTC 알고리즘, 합성곱 신경망, 순환 신경망

1. 서 론

음소 인식은 음성인식(Automation Speech Recognition) 분야에서의 주 요소 기술에 해당하는 작업으로, 주어진 음성으로부터 발음의 기본 단위에 해당하는 음소(phoneme)를 판별하는 작업이다. 음소 인식을 수행하기 위해 기존에는 인풋 자질(Feature)로 MFCC(Mel-Frequency Cepstrum Coefficient)를 사용한 은닉 마르코프 모델(HMM)과 가우시안 믹스처 모델(GMM)과 같은 확률 함수 모델이 일반적으로 사용되었다

※ 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2017R1D1A1B03034511).
※ 이 논문은 2018년도 한국정보처리학회 춘계학술발표대회에서 'Deep CNN 기반의 한국어 음소 인식 모델 연구'의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 서울대학교 융합과학부 인지컴퓨팅연구실 석사과정
†† 비 회 원 : 서울대학교 융합과학부 인지컴퓨팅연구실 박사과정
††† 종신회원 : 서울대학교 융합과학부 부교수
Manuscript Received : July 6, 2018
First Revision : August 13, 2018
Second Revision : September 11, 2018
Accepted : October 21, 2018
* Corresponding Author : Gweon Gahgene(ggweon@snu.ac.kr)

[1-4]. 하지만 인공 신경망이 최근 들어 다양한 분야에서 성능을 내기 시작하면서, 점차 심층 신경망 구조를 기반으로 음소 인식에서 더 나은 성능을 내고자 하는 시도가 계속해서 이루어지고 있다[5-7].

하지만 타 언어에서 신경망 모델을 사용하여 음소 인식을 수행하는 경우가 점차 증가하고 있는 반면, 최근의 한국어 음소 인식 연구 대부분은 기존의 HMM-GMM을 사용하여 모델을 구성하였으며, 신경망 구조를 도입한 연구는 찾아보기 어려웠다. 한국어는 음소 인식 태스크가 가장 많이 수행되고 있는 영어와 비교하였을 때 음소의 개수는 49개로 44개인 영어보다 많다. 또한 일상 언어에서 한국어의 발화 음소 분포는 모음이 40%, 자음이 60% 정도인 영어[8]와 달리 자음이 53%, 모음이 47% 정도인 것으로 알려져 있다[9]. 따라서 영어에서 상당한 수준의 성능이 나오는 것으로 알려져 있는 신경망 모델을 한국어에 적용하였을 때, 어느 정도의 성능이 나타나는지를 확인하기 위해서는 별도의 연구가 필요하다. 또한 HMM-GMM을 사용하여 음소 인식을 수행하기 위해서는 학습을 위해 전문가에 의해 제작된 강제정렬(force-alignment) 코퍼스가 요구되나, 공개되어 있는 강제정렬된 코퍼스는 한국어의 경우 매우 부족한 상황이다. 이런 상황을 인지하여, 본 연구에서는 (1) 해외 사례에서 보고된 바 있는 강제정렬 없이 학습을 수행할 수 있도록 제안된 음소 인식 모델에 CTC-알고리즘을 적용하였으며, (2) 영어와 다른 특성을 가진 한국어에서의 성능 개선 효과를 기대하며 합성곱 신경망(CNN)을 기반으로 순환 신경망(RNN)을 결합한 형태의 음소 인식 모델을 개발하였다. 또한 본 연구에서는 (3) CNN과 RNN을 어떤 구조로 결합할 때, 몇 층의 RNN을 사용할 때 성능이 가장 좋은가에 대한 비교 실험을 진행하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 음소 인식 태스크를 수행한 선행 연구 사례를 살펴본다. 다음으로 3장에서는 본 연구를 수행하기 위해 사용한 음성 말뭉치와 실험 방법 및 절차를 제시한다. 이어서 4, 5장에서는 2차에 걸쳐 수행된 7가지 모델에 대한 실험을 토대로 DeepCNN을 기반으로 하여 RNN을 결합한 형태의 음소 인식 모델을 제안한다. 구체적으로 본 연구에서는 크게 두 가지의 비교 실험을 진행하였다. 먼저, 실험 1에서는 CTC 알고리즘을 바탕으로 CNN을 중점적으로 사용한 3가지 종류의 DeepCNN 모델을 구성하여 성능을 비교하였다. 다음으로 실험 2에서는 실험 1의 결과를 참고하여 RNN 계열 신경망의 성능 기여를 확인하기 위해 6개의 모델을 제작하여 성능을 비교하였다. 6장에서는 본 연구에서 제안한 모델의 가능성을 검토하고 기존 모델과의 비교를 수행하며, 마지막으로 7장에서는 본 연구에서 제안한 모델의 의의와 향후 연구 방향을 논의한다.

2. 관련 연구

2.1 선행 연구에서의 음소 인식 기법

전통적으로 90년대 이래 음소 인식 분야에서는 HMM-GMM을 결합한 모델이 널리 사용되어 왔다[1-4]. 하지만 HMM-GMM 모델의 입력 자질로 사용되는 MFCC 자질은 비선형적인 멜 주파수 대역폭을 선형 변환에 해당하는 이산 코사인 변환(DST)을 통해 압축하는 과정에서 미세한 주파수의 변화들을 모두 사상시킨다. 이렇게 미세한 주파수 변화가 모두 사라지면 화자별로 나타나는 고유한 음색의 특성들을 반영하기 어렵게 되어 학습 과정에서 활용할 수 있는 정보량이 줄어들게 되므로 성능 향상의 여지가 제한된다. 이런 이유로 인해 해외에서는 90년대부터 MFCC가 아닌 정보 손실이 적은 다른 자질로도 학습이 가능한 인공 신경망 기반의 음소 인식이 HMM-GMM 모델의 대안 후보로 30년이 넘는 기간 동안 활발하게 연구되어 왔다[5]. 더욱이 해외 연구 사례에 따르면 단순히 인공 신경망을 사용하는 것 뿐 아니라 심층 신경망(DNN)과 HMM을 결합한 하이브리드 모델을 구성하는 것도 시도된 적이 있으며, 이 하이브리드 모델은 HMM만 사용한 모델보다 높은 인식률을 보였던 것으로 알려져 있다[10, 11]. 특히 가장 활발하게 연구 결과가 보고된 영어의 경우, HMM 기반 음소 인식기의 Phoneme Error Rate (PER)는 24.25 정도였으며[3], 하이브리드 모델의 경우 20.25의 결과가 보고되고 있다[10]. 최근 들어서는 단순히 DNN뿐만 아니라 시계열 정보를 처리하는데 특화된 인공 신경망인 순환 신경망(RNN) 계열의 모델을 사용하여, PER을 17.7 까지 낮추어 당대 최고 성능을 보였던 사례도 보고된 바 있다[12]. 위의 선행 연구들은 대부분 영어 음소 말뭉치인 TIMIT을 사용하였으며 한국어로 된 음소 인식 결과는 찾아보기 어려웠다. 다만 [13] 연구가 HMM-GMM 모델을 기반으로 수행된 적이 있으며, 10-12정도의 PER 수치를 보였던 것으로 보고되고 있다.

하지만 최근 널리 사용되고 있는 순환 신경망은 시계열 데이터에 대해 현재 상태를 지속적으로 반영하면서 학습이 수행되기 때문에 속도가 느리고, 다른 신경망 구조에 비해 더 많은 말뭉치를 요구하기 때문에 학습에 많은 제약이 있어 왔다. 특히 많은 양의 말뭉치가 요구된다는 제약은 한국어와 같은 양질의 데이터가 부족한 언어에서 많은 제약이 될 수 있다. 이런 이유로 인해 음성 인식 분야에서는 최근 들어 RNN에 대한 의존도를 낮추고 학습이 비교적 빠르고 쉬운 것으로 알려져 있는 CNN을 활용하고자 하는 연구가 많이 이루어지고 있다[14-17].

2.2 CTC 알고리즘

기존의 음소 인식에서 많이 사용하던 HMM-GMM 모델이나 하이브리드 모델(DNN-HMM/GMM 모델), 그리고 순환 신경망 기반의 모델들은 강제 정렬된 음성 말뭉치가 있어야 학습이 가능했다. 하지만 강제정렬 말뭉치를 만들기 위해서는 모든 오디오 파일들의 구간을 나눠 음소 레이블을 붙이는 전문가의 수작업이 요구되므로, 제작에 오랜 시간과 비용이 들어 데이터 수집에 많은 어려움이 있었다. CTC는 이런 어려움을 해결하기 위해 강제정렬 되지 않은 말뭉치로도 시계열 학습을 수행할 수 있도록 고안된 알고리즘이다[18]. CTC를 사용하게 되면 학습 과정에서 나뉘어진 시간 레이블(Temporal Label) 각각에 대해 확률분포에 기반한 음소 예측

이 가능해지게 된다. 이렇듯 각각의 시간 레이블마다 가장 등장 확률이 높은 음소를 할당하는 식으로 학습이 이루어지게 되면, 기존처럼 미리 강제정렬된 음소 정보를 가지고 학습을 진행하지 않게 된다. 따라서 음성 데이터와 전사 자료만 있으면 강제정렬 데이터가 없어 학습에 사용하지 못했던 음성 데이터로도 학습을 수행할 수 있다.

CTC의 구조는 입력 신호 X에 대해 신호 배열 X로부터 체인룰(Chain Rule) 로 연결된 배열 Y를 찾는 형태로, 수식은 다음과 같다. 음소 인식의 경우 입력 신호 X는 오디오 데이터가 되며, 배열 Y는 오디오 데이터에 대한 음소 인식 결과가 된다.

$$P(Y|X) = \prod_{i} P(y_i|X, y_{<i}) \tag{1}$$

CTC는 입력 신호 배열 X와 길이가 같은 시간 레이블(Temporal Label)을 가진다. 이때, Y의 레이블의 총 유형 개수는 K개의 유형에 1개의 공백 레이블이 더해진 (K+1)개이다. 여기서 공백 레이블은 특정 시간 레이블에 대한 확률값이 주어진 한계점(threshold)에 이르지 못했을 경우 부여되게 된다. CTC가 진행된 뒤에는 CTC-디코딩이 수행된다. CTC-디코딩은 CTC를 통해 얻은 시간 레이블 배열을 음소 배열 Y로 변환시켜 주는 과정을 의미하며, CTC-디코딩 과정은 1) 시간 레이블에 포함되어 있는 중복 레이블을 제거한 뒤, 2) 공백 레이블을 제거하는 순서로 진행된다.

CTC 알고리즘을 음성 인식 분야에서 활용한 연구들로는 기존에 RNN과 함께 사용하던 CTC 알고리즘을 CNN 과 결합하여 사용한 Palaz (2015)의 연구 사례[15], 그리고 CTC 알고리즘에 어텐션 메커니즘(Attention Mechanism)을 합친 CTC-attention에 대한 연구를 수행한 Hori (2017)의 사례[19] 등이 있다. 본래 CTC는 RNN에서 각각의 시간마다 확률분포값을 구하는 형태로 계산이 이루어졌는데, [15] 연구는 이를 CNN에서의 시간 프레임에 적용해도 동일한 결과를 얻을 수 있다는 것을 밝힌 바 있다. 하지만 한국어 음소 인식과 관련하여 CNN-CTC를 사용한 연구는 아직 보고된 바 없다.

3. 실험 환경

본 연구에서는 제안된 알고리즘의 타당성을 증명하기 위해 두 가지 비교 실험을 진행하였으며, 이 두 실험에서 사용한 음성 말뭉치와 입력 자질, 전처리 과정은 Fig. 1과 같이 모두 동일했다. 먼저 본 연구에서 음소 인식 학습을 위해 사용한 음성 말뭉치는 국립국어원에서 제작한 '서울말 낭독체' 말뭉치이다[20]. 서울말 낭독체 말뭉치는 3대째 이상 서울에 살고 있는 20~60대 남, 여 서울말 사용자 80명의 음성 녹음 및 전사 파일로 구성되어 있다. 음성 파일의 개수는 총 71,216개이며, 분량은 약 180시간으로 모든 음성 파일은 한 문장 단위로 실험실 환경에서 녹음되어 있다. 각 문장의 발화 시간은 약 3초에서 10초 사이였으며 평균 시간은 약 5.3초였다. 각 71,216개의 음성 파일에서 추출한 벡터는 평균 176.5개의 음소로 구성되었으며, 벡터 하나에서 추출된 최대 음소 개수는

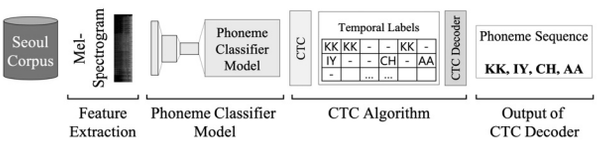

Fig. 1. Phoneme Classifier Overview

416개였다. 샘플링 주파수는 모두 16,000Hz이다. 전사 자료는 19개의 수필 및 단편소설을 한 문장으로 나누어 놓은 형태로, 대략 2,000문장으로 구성되어 있다. 하지만 검수를 진행한 결과 전사 자료를 잘못 읽은 음성 파일과 명확하지 않은 발음으로 읽은 음성 파일이 다소 섞여 있는 것이 확인되어, 문제가 있는 파일은 실험에 사용하지 않았다.

다음으로 두 실험의 입력 자질로 Mel-Spectrogram을 사용하였다. Mel 주파수 대역폭은 일반적으로 사용하는 대역폭인 40으로 설정하였으며, 프레임은 5ms 길이로 각각 나누었다. 하지만 CTC 알고리즘을 사용하였으므로 시간 레이블에서의 예측을 용이하게 하기 위하여 일반적으로 하는 방식인 프레임 간의 겹침(overlapping)을 설정하지는 않았다.

끝으로 두 실험에서 공통된 전처리 과정으로 실제 발음을 반영하는 음소 형태가 아니라 국어 표기법에 따라 기록되어 있는 전사 자료를 한국어 표준 발음에 따르는 음소 배열 Y*로 바꾸는 변환 작업을 진행하였다. 강제정렬된 말뭉치 없이 CTC 알고리즘으로 학습을 진행한다 하더라도, 음소 인식 학습을 위해서는 말뭉치의 각 문장들에 대해 정확한 음소 배열이 무엇인지를 알려 주는 정답지가 필요하였기 때문이다. 이에 정답지의 제작을 위해 공개 소프트웨어인 Grapheme to Phoneme(G2P) 프로그램인 KoG2P를 사용하여 전사-음소 변환 작업을 수행하였다[21]. 이어서 변환된 음성 말뭉치를 Train 6, Validation 1, Test 1의 비율로 무작위로 나누어 학습에 사용하였다. 학습을 통해 1 프레임당 1개의 시간 레이블 C가 추출되도록 하였으며, 이렇게 추출된 시간 레이블 C는 총 50개의 유형으로 구성되어 있다. 각 유형들은 49개의 한국어 음소와 1개의 공백 레이블(blank)로 구성된다.

$$C \in \{AA, AX, CH, EH, EY, \ldots, blank\} \tag{2}$$

위의 세 가지 실험 환경을 바탕으로 하여, 먼저 실험 1에서는 CTC 알고리즘을 바탕으로 CNN을 중점적으로 사용한 3가지 종류의 DeepCNN 모델을 구성하여 성능을 비교하였다. 다음으로 실험 2에서는 실험 1의 결과를 참고하여 RNN 계열 신경망의 성능 기여를 확인하기 위해 4개의 모델을 제작하여 성능을 비교하였다.

두 실험에서 사용된 CNN 구조는 이미지 인식 분야에서 좋은 성과를 보인 바 있는 심층 CNN 기반 모델인 VGG16 모델[16, 17, 22]의 구조를 변형한 것이다. VGG16 모델은 (224 × 224) 크기의 이미지를 입력 자질로 사용하도록 고안된 모델로, 16개~19개의 CNN과 연결 신경망(Fully Connected Neuron)을 깊게 쌓는 것이 특징이다. 하지만 신경망을 깊이 쌓아 추상적인 수준의 자질을 뽑아내야 하는 이미지와는 달리, 본 연구에서 사용하고자 하는 Mel-Spectrogram은 시간

에 따라 주파수 대역폭별로 어떤 소리가 나타나는지, 그리고 각 대역폭에 따라 소리의 강도가 어떠한지를 담고 있는 자질이므로 이미지에 비해 뽑아내고자 하는 정보가 비교적 명확한 편이다. 따라서 이에 맞춰 본 연구에서는 VGG16 구조를 깊게 쌓지 않고 간략화하는 방식으로 모델을 구성하기로 하였다. 구체적인 실험 과정은 아래와 같다.

4. 실험 1: CNN 단일 - RNN 결합 모델 비교

실험 1에서는 우선 VGG16 구조를 변형한 DeepCNN 모델을 구성하여 한국어에서 CNN 기반 모델의 음소 인식 성능을 확인하여 보고자 하였다, 또한 이 모델에 RNN 계열 신경망인 BiLSTM과 BiGRU를 붙여, RNN계열 신경망에 의한 성능 개선이 나타나는지를 확인하고자 하였으며, 추가적으로 BiLSTM과 BiGRU 중에 어느 신경망의 성능 향상이 더 컸는지 확인해보고자 하였다.

4.1 학습 방법

실험 1에서는 Fig. 2와 같이 VGG16 구조를 간략화한 DeepCNN 기반 3 가지의 모델 구조를 구성하여 비교를 수행하였다. 비교에 활용된 모델들은 (a) CNN만 사용한 모델, (b) CNN과 BiLSTM 층을 사용한 모델, 그리고 (c) CNN과 BiGRU 층을 사용한 모델의 세 종류이다.

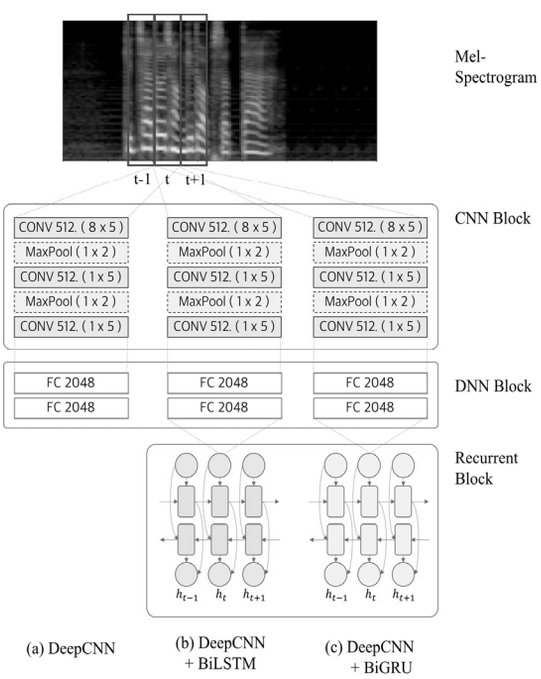

Fig. 2. VGG Networks based DeepCNN, DeepCNN + BiLSTM, DeepCNN + BiGRU Models

실험 1에서 공통으로 사용된 DeepCNN 모델의 구조는 VGG16 구조에서 세 겹의 같은 필터 사이즈의 CNN으로 구성된 컨벌루션 층(Convolution Layer)을 CNN 1층으로 대체하는 방식으로 구조의 깊이를 줄였다. 또한 각 CNN 층마다 Max Pooling을 진행하였다.

학습을 위해 입력 자질로 사용한 Mel-Spectrogram을 시간 레이블 단위로 작게 잘라 사용하였으며, 입력 자질의 크기인 (8 X 40)은 (시간 레이블의 시간 X 주파수 대역)을 의미한다. 본 실험에서 사용하는 음소 단위 입력 자질의 크기에 맞춰, CNN 층에서 사용하는 커널은 주파수 대역폭이 더 긴 직사각형 모양의 커널을 사용하였다.

한편, 모델 (b), (c)에서는 CNN 층를 통과한 출력 배열을 입력 자질로 받는 BiLSTM, BiGRU 층을 CNN 층의 최하단부에 추가하여, CNN만으로 구성된 모델에 비해 성능 차이가 발생하는지를 확인하고자 하였다. 모델에 사용한 활성 함수(Activation Function)로는 CNN에는 clipped rectified-linear unit(clipped-ReLU)를, RNN에는 하이퍼볼릭 탄젠트(Hyperbolic Tangent)를 사용하였다. Clipped-ReLU의 경우 수식은 아래와 같다.

$$Clipped-ReLU: g(z) = \min\{\max\{0, x\}, 20\} \quad (3)$$

또한 Fig. 1에 있는 세 모델에서 각 레이어를 모두 통과한 최종값은 CTC로 전달되어 시간 레이블이 예측된 뒤, 예측 결과를 바탕으로 손실값이 산출되었다. 모델 훈련에는 Adam 최적화 함수(Optimizer)를 사용하였으며, 과적합(overfitting)을 막기 위해 드랍아웃(dropout) 수치로 20%를 설정하였고, 또한 음성 자료에 무작위로 백색 소음과 분홍색 소음을 추가하였다. 세 모델 모두 15번째 주기(epoch)에서 손실값(loss)의 변화가 매우 작아, 조기 중단(Early Stopping)을 통해 학습을 마무리하였다.

4.2 결과

본 연구에서는 음소 인식기의 정확도를 측정하기 위해 예측한 레이블 배열 $\hat{Y}$와 정답지 Y 사이의 PER을 구하여 음소 인식기의 최종 성능을 평가하였다[18]. PER 수치는 자료 집합 S에 대하여 $S \subseteq (X, Y)$일 때, $\hat{y}$를 음소 인식기의 출력으로 정의하면, 아래 수식을 통하여 구할 수 있다. 수식에서 ED는 Edit Distance를 의미한다. PER 수치는 음소 레이블의 오류 정도를 의미하므로, PER 수치가 낮을수록 성능이 좋은 모델을 의미한다.

$$PER(S) = \frac{1}{S}\sum \frac{ED(y, \hat{y})}{len(y)} \quad (4)$$

각각의 세 가지 종류의 모델에 대해 8800개의 자료 집합을 대상으로 실험을 수행하고, PER 수치를 산출하여 표 1과 같은 결과를 얻었다.

Table 1. Results for Phoneme Recognition

Model	PER	Time
(a) DeepCNN	32.66	9h
(b) DeepCNN + BiLSTM	29.94	9h 54m
(c) DeepCNN + BiGRU	35.48	9h 45m

실험을 통하여 CNN만을 사용한 모델이나 BiGRU를 하단에 추가한 모델에 비해 BiLSTM을 하단에 추가한 모델이 가장 나은 PER 결과를 보이는 것을 확인할 수 있었다. 이 결과를 통해 CNN모델에 RNN층을 추가하는 것이 성능 향상에 도움이 된다는 것을 알 수 있었으며, BiGRU보다 BiLSTM의 성능 개선이 더 크다는 것을 알 수 있었다. 하지만 세 모델 모두 선행 연구에 비해 만족스러운 결과를 내지 못했다. 우선 모델 (a) 만으로는 음소 인식이 만족할 만큼 이루어지지 않는다는 것을 확인하였으며, 모델 (b)와 (c)에서는 DeepCNN 의 출력값이 BiLSTM과 BiGRU의 학습에 상당한 영향을 미칠 것이라 기대하였으나 성능 향상폭이 크지 못했다. 이러한 결과는 DeepCNN의 출력값만으로는 음소에 대한 충분한 정보가 주어지지 못해서인 것으로 추정된다. 이에 본 연구에서는 이를 개선하고자 DeepCNN 모델의 중간층에 RNN계열 신경망을 여러 겹 추가함으로써, DeepCNN으로부터 최종 출력값을 받아 학습하는 것이 아니라 중간 출력값을 받아 학습을 수행할 수 있도록 모델을 변경하기로 하였다.

이에 본 연구에서는 실험 2를 추가로 설계하여 RNN이 음소 인식 태스크에서 최종적으로 어느 정도 수준까지 성능에 기여할 수 있는지 확인하고, 몇 개의 RNN 층을 추가해야 유의미한 성능 향상이 나타나는지를 확인하기로 하였다. 이에 본 연구에서는 RNN 층의 개수를 서로 다르게 설정한 6개의 DeepCNN+RNN 모델을 추가로 구성하여 학습을 진행하였다.

5. 실험 2: CNN/RNN 기반의 음소 인식 모델 개선

실험 1에서 RNN 계열 신경망이 DeepCNN 모델의 성능 향상에 기여한다는 점을 확인하기는 했지만, 세 모델 모두 기대에 미치는 성능을 보여주지 못했다. 이에 실험 2에서는 먼저 성능 개선을 위해 RNN 계열 신경망 층을 실험 1과는 달리 맨 끝단이 아닌 중간에 결합하는 방식으로 모델을 재구성하였다. 이어서 RNN 계열 신경망 층의 최적 개수를 찾기 위하여 DeepCNN+RNN 모델에서 RNN 계열 신경망 층의 개수를 다르게 설정한 여섯 가지 모델을 비교하여, 몇 개 층의 RNN 계열 신경망을 쌓는 것이 성능 향상에 가장 효과적인지에 대해 알아보고자 하였다.

5.1 실험 방법

실험 2에서는 CNN을 사용한 구조에 RNN 층을 실험 1보다 2개, 4개 더 추가한 형태의 음소 인식 모델을 구성하였다. 실험 1에서는 DeepCNN 모델과 1겹의 RNN 층을 DeepCNN 모델의 하단에 추가한 모델을 사용하여 실험을 진행하였으며, BiLSTM 층이 포함된 모델이 DeepCNN 모델보다 더 나은 성능을 내는 것을 관찰할 수 있었다. 하지만 세 모델 모두 만족스러운 수준의 PER수치를 보이지는 못하였다. 이에 실험 2에서는 성능 개선을 위해 영어 음성 인식 분야에서 높은 성능을 보고한 바 있는 Amodei[23]의 DeepSpeech2 모델의 구조를 참고하여 모델을 재설계하였다.

Fig. 3과 같이 실험 1에서 구성하였던 DeepCNN 모델에 RNN 층을 2개 더 추가하여 총 세 겹(3hop)의 RNN 층을 사용하는 형태의 개선된 구조 (e), (h)를 제작하였다. 또한 음성 인식 영역에서 RNN 계열의 신경망을 6층 이상 쌓는다 하더라도 성능 향상에 크게 기여하지 못한다는 선행 연구에서의 보고를 감안하여[24], RNN 층을 얼마나 더 추가해야 유의미한 성능 개선 효과가 있는지를 확인하고자 RNN 층 2개를 더 쌓은(5hop) 2개의 비교 모델 (f)와 (i)를 추가로 제작하여 성능을 비교하였다. 이렇게 실험에 사용된 모델은 (d) Deep CNN에 BiGRU 층을 한 겹(1hop) 사용한 모델, (e) Deep CNN에 BiGRU을 세 겹(3hop) 사용한 모델, (f) Deep CNN에 BiGRU 층을 다섯 겹(5hop) 사용한 모델, (g) Deep CNN에 BiLSTM을 한 겹(1hop) 사용한 모델, (h) Deep CNN에 BiLSTM을 세 겹(3hop) 사용한 모델, (i) Deep CNN에 BiLSTM을 다섯 겹(5hop) 사용한 모델의 총 6가지이다. 한편 DeepCNN 층은 실험 1의 DeepCNN 모델과 마찬가지로 CNN을 각각 1층씩 쌓는 구조로 구성하였으며, 각각의 CNN 층마다 Max Pooling 을 진행하였다. 활성 함수와 손실값의 산출은 실험 1과 동일하게 설정하였다.

실험 2에서 사용된 6가지 모델 학습에서도 실험 1과 같이 Mel-Spectrogram을 입력 자질로 사용하였으며, CTC-디코딩을 통해 최종 음소 출력 배열을 얻은 후, 정답지와 비교하여 PER을 산출하는 식으로 평가를 수행하였다. 모델 훈련에는 실험 1과 같이 Adam 최적화 함수를 사용하였으며, 과적합을 막기 위해 음성 자료에 무작위로 백색 소음과 분홍색 소음을

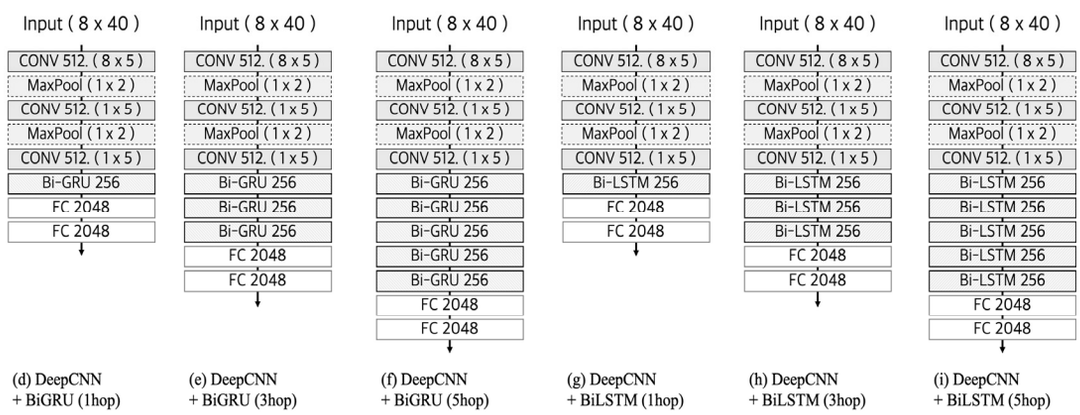

Fig. 3. DeepCNN + BiGRU(1hop ~ 5hop), DeepCNN + BiLSTM(1hop ~ 5hop) Models with Improved DeepCNN Models

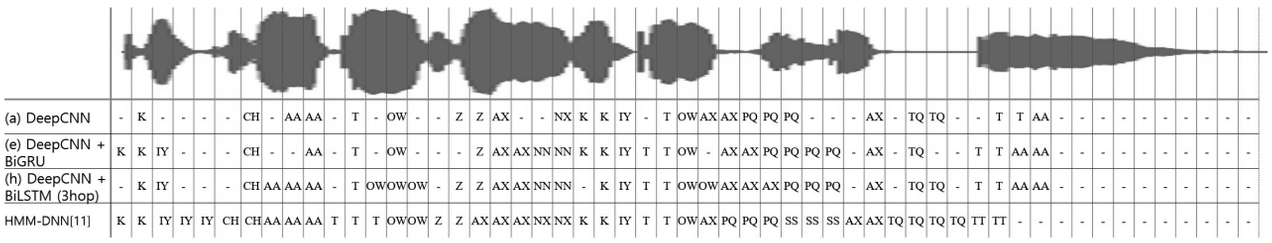

Fig. 4. Temporal Labels for Phoneme Recognition Models. One Phoneme Label per 30ms is Displayed in Each Cell

추가하였다. 하지만 실험 1과 달리 드랍아웃은 설정하지 않았다. 또한 빠른 학습을 위해 모델의 입력 데이터에 배치 정규화(Batch Normalization)를 수행하였으며, 여섯 모델 모두 10 주기에서 손실 값의 변화폭이 작아 조기 중단을 통해 학습을 마무리하였다.

5.2 결과

실험 2에서도 실험 1과 마찬가지로 8800개의 음성 자료 집합을 사용하여 PER을 산출함으로써 모델 평가를 진행하였으며, 그 결과는 표 2와 같다.

Table 2. Results for Phoneme Recognition

Model	PER	Time
(d) DeepCNN + BiGRU (1hop)	5.13	9h 39m
(e) DeepCNN + BiGRU (3hop)	4.15	13h 14m
(f) DeepCNN + BiGRU (5hop)	3.82	18h 52m
(g) DeepCNN + BiLSTM (1hop)	4.47	10h 51m
(h) DeepCNN + BiLSTM (3hop)	3.26	13h 8m
(i) DeepCNN + BiLSTM (5hop)	3.52	17h 23m

실험 1과 유사하게 DeepCNN과 BiLSTM을 함께 사용한 모델이 가장 나은 PER 수치를 보였다. DeepCNN과 RNN 계열이 함께 사용된 모델 간에서의 성능 차이를 비교하여 보았을 때, 실험 1에 사용된 (b), (c) 모델에 비해 RNN 층의 위치를 변경한 (d), (g) 모델에서 PER이 20 이상 좋아진 것으로 나타났다. 이는 시계열 정보를 반영하는 RNN 층이 맨 하단이 아니라 중간에 추가됨으로써 시간에 따른 음소의 변화 추이를 더 잘 학습할 수 있었기 때문인 것으로 판단된다. 또한 DeepCNN 모델에 BiLSTM 과 BiGRU를 사용한 모델을 서로 비교하였을 때, (g), (h), (i) BiLSTM 모델이 (d), (e), (f) BiGRU 모델보다 전반적으로 성능이 더 우수하였다. 또한 BiGRU 을 사용한 모델은 BiGRU 층을 깊게 쌓을수록 성능이 향상되었으나 BiLSTM의 경우 (e) BiLSTM (3hop)이 (f) BiLSTM (5hop)의 성능보다 높았다. 이는 RNN 계열의 층을 많이 쌓을수록 항상 성능이 향상되는 것은 아님을 시사한다.

6. 강제정렬 비교 분석

2회에 걸친 실험을 통하여 최종적으로 DeepCNN과 BiLSTM을 결합한 모델이 가장 나은 성능을 보인다는 것을 확인한 뒤, 본 연구에서는 최종적으로 실험을 통해 구축한 7가지 모델 중 3개의 모델인 (a) DeepCNN과 (e) DeepCNN+ BiGRU, (h) DeepCNN+BiLSTM과 선행 연구 [13]에서 제안된 모델 간의 음소 강제 정렬 결과를 테스트 집합에서 임의로 1개의 음성 자료를 선택하여 비교하여 보았다. 이러한 비교를 수행한 이유는 본 연구에서 구축된 모델들이 모두 CTC 디코딩을 통해 얻은 음소 아웃풋을 정답지와 비교하는 방식으로 평가를 진행하였기 때문이다. 이러한 평가 방식은 해당 음성 파일에 대하여 음소 인식이 제대로 수행되었는지를 평가할 수는 있지만, 음성 파일에서의 음소 강제정렬 위치에 대한 정확도를 알기는 제한된다는 한계가 있다. 물론 해당 오디오 파일이 음소 강제정렬에 대한 정답지를 가지고 있지는 않으므로 직접적인 평가는 제한되나, 선행 연구와의 차이점이 나타나는지를 살펴보기 위해 이러한 비교 작업을 수행하여 보았다.

선택된 음성 자료의 전사 자료는 '기차도 전기도 없었다.' 이며, 정답 음소 문자열은 {K(/k/), IY(/i/), CH(/ʧʰ/), AA(/a/), T(/t/), OW(/o/), Z(/ʧ/), AX(/ʌ/), NN(/n/), K(/k/), IY(/i/), T(/t/), OW(/o/), AX(/ʌ/), PQ(/p(s)/), AX(/ʌ/), T(/t/), TQ(/t/), T(/t/), AA(/a/)} 이다.

해당 문장에 대해 우선 음소 인식에 대하여 평가를 수행한 결과, (e), (h) 모델의 경우 PER 이 0 으로 나타났지만 (a) DeepCNN 모델의 경우는 PER이 10 으로 나타났다. 하지만 Fig. 3에서도 확인할 수 있듯이, 여섯 모델의 결과가 모두 비교적 비슷한 경향을 나타냈다. 다만 (a) DeepCNN 모델의 결과는 음소 IY(/i/, ㅣ)를 예측하지 못하였고, 음소 NN(/n/, -ㄴ)의 경우 비교적 발음이 비슷한 NX(/ŋ/, -ㅇ)로 오인식한 것을 확인할 수 있었다. 그러나 NN과 NX의 경우는 사람들의 발음 습관에 따라 발음이 혼용되는 경향이 있으므로 두 발음 모두 정답으로 인정되는 경우도 있기 때문에[13], 이는 비교적 경미한 오류로 판단된다.

7. 토 의

본 연구에서 최종적으로 제안한 모델의 최종 PER 수치는 3.26이다. 다른 실험과 테스트 셋의 종류 및 실험 방법 등이 다르기 때문에 직접적인 비교는 불가능하나, 이는 한국어 음소 인식 연구 사례로 가장 최근에 보고된 바 있는 [13]의 음소 인식기 성능보다 비교적 좋은 결과라고 할 수 있다. [13]의 연구 사례는 147,263개의 테스트 셋을 대상으로 19,541개의 불일치를 보여, 12 정도의 PER 값이 나온 것으로 보고되

고 있다. 이를 통해 기존의 HMM-GMM 방식이나 하이브리드 방식을 사용하지 않고 딥 러닝만을 사용해서도 음소 인식기의 학습이 가능하다는 것을 확인할 수 있었다. 또한 결과 외에 본 연구를 통해 추가적으로 알게 된 사실은 다음과 같다.

첫째로, CTC가 강제정렬된 말뭉치 없이도 음소 인식 태스크를 수행할 수 있게 해 준다는 것이 학습 결과를 통해 입증되었다. 더욱이 선행 연구 결과와의 비교를 통해 딥러닝과 CTC 모델을 사용하여 대략의 음소 위치까지 어느 정도 확인할 수 있었다는 것을 고려한다면, 강제정렬 코퍼스를 만드는 태스크에서도 본 연구에서 제안된 방법이 어느 정도 해결책이 될 수 있을 것으로 예상된다.

둘째로, CNN과 RNN을 함께 사용하는 것이 CNN과 RNN을 각각 사용한 것보다 한국어 음소 인식 분야에서 성능을 개선할 수 있는 방안이라는 것을 실험 1을 통해 확인할 수 있었다. 본 실험은 CNN과 RNN을 함께 사용하여 최초로 한국어 음소인식 태스크를 수행하였다는 점에서 의의가 크다고 할 수 있다. 실험 1의 결과를 볼 때, DeepCNN만 사용한 모델의 경우 출력값을 확인한 결과 확인된 대부분의 오류는 발음 시간이 길고 진폭이 작은 자음을 놓치는 경우에 해당하였으며, 다른 오류들은 발음상 비슷한 음소가 잘못 인식되는 경우가 대부분이었다. 이러한 오류 경향은 RNN 층을 DeepCNN 모델의 중간에 추가함으로써 뚜렷이 개선되는 것이 관찰되었는데, 이를 통해 CNN이 음소 각각의 특성에 대해서는 학습하지만 음소의 순간적인 변화를 감지하는 데 있어서만큼은 RNN이 CNN보다 더 나은 구조라는 것을 확인할 수 있었다. 이에 본 연구에서는 음소 인식 영역에서는 CNN, RNN을 함께 사용할 때, CNN의 단점을 해결할 수 있음을 알 수 있었다.

셋째로, DeepCNN과 3층의 BiLSTM을 함께 사용한 모델이 가장 뛰어난 성능을 나타내는 것이 확인되었다. 실험 2에서 BiGRU를 3층 사용할 때보다 5층 사용할 때 성능이 더 좋았으며, BiLSTM의 경우 5층 사용하였을 때보다 3층을 사용하였을 때 성능이 더 높았다. 또한 이는 선행 연구[24]에서 6층 이상의 RNN 모델은 성능 향상이 되지 않는다고 보고한 결과와 같다. 더불어 실험 2에서 RNN 층이 깊어질수록 학습 시간이 매우 늘어났으며, 이에 본 연구에서는 성능 향상과 학습 시간의 균형을 맞추기 위해 3층의 양방향 RNN 층을 사용하는 것이 적합하다는 결론을 얻었다.

넷째, 실험 1과 실험 2에서의 비교 실험에서 BiLSTM을 사용한 모델이 BiGRU를 사용한 모델보다 성능이 더 높았다. 이는 음성 인식 태스크에서 BiGRU보다 BiLSTM을 사용할 때 성능이 더 좋다고 알려진 해외의 연구사례[25]에서의 보고와 같은 결과이며, 일반적으로는 GRU가 LSTM보다 성능이 높다는 시계열 데이터 모델링 분야[26]와는 다른 결과이다. 이는 음소 인식과 같은 태스크에서는 음소가 분절적으로 변하는 것이 아니라 연속적으로 구강 구조의 변화에 의해 음소가 달라지게 되므로, 이러한 시계열 정보를 보다 잘 보존할 수 있는 LSTM이 연산량을 줄이고 간략화한 형태인 GRU보다 더 유리한 구조이기 때문인 것으로 판단된다.

하지만 본 연구에서는 서울말 낭독체 말뭉치의 한계로 인해, 음성 파일의 음소 정답지를 만드는 과정에서 전사 자료와 G2P 프로그램을 사용할 수밖에 없었다. 이로 인해 모델이 음성 파일 개개의 정확한 실제 음소를 파악하지 못하고, 모델 기반으로만 음소를 파악하였을 수도 있다. 하지만 음성 합성처럼 대략적인 음소 인식 정보만 있어도 충분한 태스크에는 본 연구의 방법론이 충분히 효과적으로 활용될 수 있으며, 전문가들이 정밀한 강제정렬 말뭉치를 제작하는데도 본 연구의 결과물이 초벌 작업으로 유용하게 사용될 수 있다. 이러한 가능성들을 감안한다면 본 연구의 결과는 충분한 의의를 가진다고 할 수 있을 것이다. 이에 본 연구에서는 이번 연구 성과를 토대로 하여 대략적인 음소 인식 정보에 기반하여 수행할 수 있는 태스크 중 하나인 발음 실수 감지에 대한 연구를 진행하여 보고자 한다.

References

[1] Gales, Mark JF. “Maximum likelihood linear transformations for HMM-based speech recognition,” *Computer Speech & Language*, Vol.12, No.2, pp.75-98, 1998.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, Vol.86, No.11, pp.2278-2324, 1998.

[3] Glass, James R. “A probabilistic framework for segment-based speech recognition,” Computer Speech & Language , Vol.17, No.2-3, pp.137-152, 2003.

[4] Schwarz, Petr, Pavel Matějka, and Jan Černocký. “Towards lower error rates in phoneme recognition,” International Conference on Text, Speech and Dialogue. Springer, Berlin, Heidelberg, 2004.

[5] Waibel, Alexander, et al., “Phoneme recognition using time-delay neural networks,” Readings in Speech Recognition, 1990. 393-404.

[6] Bengio, Yoshua. “A connectionist approach to speech recognition,” Advances in Pattern Recognition Systems Using Neural Network Technologies, pp.3-23. 1993.

[7] Mohamed, Abdel-rahman, George E. Dahl, and Geoffrey Hinton. “Acoustic modeling using deep belief networks,” *IEEE Transactions on Audio, Speech, and Language Processing*, Vol.20, No.1, pp.14-22, 2012.

[8] Ardussi Mines, M., Hanson, B. F., & Shoup, J. E. “Frequency of Occurrence of Phonemes in Conversational English,” *Language and Speech*, Vol.21, No.3, pp.221-241, 1978.

[9] Ji-Young Shin. “Phoneme and Syllable Frequencies of Korean Based on the Analysis of Spontaneous Speech Data,” *Communication Sciences and Disorders*, Vol.13, No.2, pp.193-215, 2008.

[10] Seltzer, Michael L., and Jasha Droppo. “Multi-task learning in deep neural networks for improved phoneme recognition,” *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013.

[11] Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. “Hybrid speech recognition with deep bidirectional LSTM,”

Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013.

[12] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013.

[13] Minsoo Na and Minhwa Chung, "Assistive Program for Automatic Speech Transcription based on G2P Conversion and Speech Recognition," *Proc. Conference on Korean Society of Speech Sciences*, pp.131-132, 2016.

[14] Palaz, Dimitri, Ronan Collobert, and Mathew Magimai Doss. "End-to-end phoneme sequence recognition using convolutional neural networks," arXiv preprint arXiv: 1312.2137 (2013).

[15] Palaz, Dimitri, Mathew Magimai Doss, and Ronan Collobert. "Convolutional neural networks-based continuous speech recognition using raw speech signal," *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.

[16] Heck, Michael, et al., "Ensembles of Multi-scale VGG Acoustic Models," *Proc. Interspeech 2017* (2017): 1616-1620.

[17] Zhang, Ying, et al., "Towards end-to-end speech recognition with deep convolutional neural networks," arXiv preprint arXiv:1701.02720 (2017).

[18] Graves, Alex, et al., "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006.

[19] Hori, Takaaki, et al., "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," arXiv preprint arXiv:1706.02737 (2017).

[20] National Institute of the Korean Language (NIKL), Seoul Reading Speech Corpus("서울말 낭독체 발화 말뭉치"), 2003. URL: https://ithub.korean.go.kr

[21] Yejin Cho, Korean Grapheme-to-Phoneme Analyzer (KoG2P), 2017.
GitHub repository : https://github.com/scarletcho/KoG2P

[22] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556 (2014).

[23] Amodei, Dario, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," International Conference on Machine Learning. 2016.

[24] Xiong, Wayne, et al., "The Microsoft 2016 conversational speech recognition system," *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017.

[25] Sainath, Tara N., et al., "Convolutional, long short-term memory, fully connected deep neural networks," *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.

[26] Chung, Junyoung, et al., "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555 (2014).

[27] Xiong, Wayne, et al., "The Microsoft 2016 conversational speech recognition system," *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017.

홍 윤 석

https://orcid.org/0000-0002-8644-8636
e-mail : yshong93@snu.ac.kr
2017년 숭실대학교 글로벌미디어학부 (학사)
2017년～현 재 서울대학교 융합과학부 인지컴퓨팅연구실 석사과정
관심분야 : Speech Recognition, Natural Language Processing, Human-Computer Interaction

기 경 서

https://orcid.org/0000-0002-9866-0052
e-mail : kskee88@snu.ac.kr
2013년 서울대학교 미학과(학사)
2016년 서울대학교 미학과(석사)
2017년～현 재 서울대학교 융합과학부 인지컴퓨팅연구실 박사과정
관심분야 : Speech Recognition, Natural Language Processing, Human-Computer Interaction

권 가 진

https://orcid.org/0000-0003-3268-477X
e-mail : ggweon@snu.ac.kr
2002년 University of California, Berkeley, Economics / Computer Science(B.A. 학사)
2004년 Carnegie Mellon University, Human Computer Interaction (M.S., 석사)
2012년 Carnegie Mellon University, Human Computer Interaction(Ph.D., 박사)
2012년～2016년 KAIST 지식서비스공학과 조교수
2016년～현 재 서울대학교 융합과학부 부교수
관심분야 : Human-Computer Interaction, Learning Science, Multimedia Educational Technology, Natural Language Processing