Interspeech 2018
2-6 September 2018, Hyderabad

# Automatic Miscue Detection using RNN Based Models With Data Augmentation

*Yoon Seok Hong, Kyung Seo Ki, and Gahgene Gweon*

Graduate School of Convergence Science and Technology,
Seoul National University, South Korea

yshong93@snu.ac.kr, kskee88@snu.ac.kr, ggweon@snu.ac.kr

## Abstract

This study proposes a method of using data augmentation to address the problem of data shortages in miscue detection tasks. Three main steps were taken. First, a phoneme classifier was developed to acquire force-aligned data, which would be used for miscue classification and data augmentation. In order to create the phoneme classifier, phonetic features of "Seoul Reading Speech" (SRS) corpus were extracted by using grapheme-to-phoneme (G2P) to train CNN-based models. Second, to obtain miscue labeled corpus, we performed data augmentation using the phoneme classifier output, which is artificially generated miscue corpus of SRS (modified-SRS). This miscue corpus was created by randomly deleting or modifying sound sections according to three miscue categories; extension (EXT), pause (PAU), and pre-correction (PRE). Third, the performance of the miscue classifier was tested after training three types of RNN based models (LSTM, BiLSTM, BiGRU) with the modified-SRS corpus. The results show that the BiGRU model performed best at 0.819 in F1-score on augmented data, while BiLSTM model performed best at 0.512 on real data.

**Index Terms**: automatic miscue detection, phoneme classification, data augmentation, recurrent neural network

## 1. Introduction

Miscue analysis, which identifies and analyzes pronunciation errors while reading aloud, has long been used as an effective indicator for students' reading skills [1, 2]. Despite its effectiveness, providing feedback on miscues takes a lot of resource and time. Although developing automatic miscue detection technology can contribute to reading fluency instructions, one of the main barriers in conducting the miscue detection task is difficulty in obtaining miscue data that can be used for training in a speech recognition system. To address such problem of data scarcity, this study proposes to use data augmentation to develop automatic miscue detection method.

Data augmentation has been applied in solving the data shortage problem in various fields of speech recognition tasks. Specifically, Ko et al. [3] showed that manipulating speed of audio data can yield performance improvements in speech recognition. Hartmann et al. [4] also proposed a data augmentation method using noise and speed perturbation for languages that lack speech data, such as Amharic, Guarani, Igbo, Pashto. Both of these studies have shown the effectiveness of data augmentation by manipulating existing real audio data. Furthermore, attempts have also been made to create new artificial speech data. Alharbi et al. [5], artificially created sentences by combining word-pronounced audio data and stuttering speech data, which were manipulation of the original audio. Fainberg et al. [6] also proposed a data augmentation method for children's speech recognition by modulating adult speech data. Similarly, we propose applying data augmentation techniques for miscue detection by creating artificial audio data.

To create artificial data with miscues, we inspected existing literature on the different types of miscues. Unlike existing works on binary miscue detection task [7, 8], we need miscue types that occur in natural speech of reading. Since our goal is to classify miscue types for reading corpus at the phoneme level, we selected miscue types [9] that are applicable at the phoneme level. The three selected types are extension, pause, and pre-correction. These miscues were chosen because they have salient phonetical characteristics and they are considered to be detectable by audio, and without reference to canonical pronunciation.

Prior to creating artificial data for miscues, we examined previously published datasets that can be used as an input. In the Korean language, which is our target language, only two audio datasets are publicly available: "Korean Corpus of Spontaneous Speech" [10] and "Seoul Reading Speech" (SRS) Corpus [11]. These two datasets correspond to the two types of speech used in miscue detection tasks; spontaneous and reading speech. Neither of these two datasets is labeled with miscues. Since our target task is to detect miscues while reading, we used the SRS corpus for data augmentation.

To perform miscue detection tasks using data augmentation, we implemented a phoneme classifier and a miscue classifier. Section 2 describes the convolutional neural network (CNN) based phoneme classifier which extracts the phonetic features of the acquired reading corpus, along with the data augmentation method. Section 3 describes proposed recurrent neural network (RNN) based miscue classification model. We end the paper with a discussion of the miscue detection task.

## 2. Phoneme Classifier

The input data for miscue classification model is speech data which is aligned with phonemes. Therefore, we first created a phoneme classifier that can provide aligned phoneme labels given just a speech file. Unlike existing research that detects miscues with miscue annotated corpus [9], our approach starts the miscue detection task using speech files with only transcriptions, but no aligned phonemes. Figure 1 shows our phoneme classification pipeline as well as the data augmentation process.



10.21437/Interspeech.2018-1644

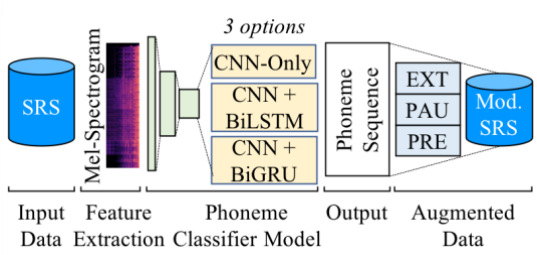


Figure 1: *Phoneme classification pipeline and data augmentation process.*

To build the phoneme classification model, we applied and compared three types of CNN-based deep learning architecture, CNN-Only, CNN+bidirectional long-short term memory (BiLSTM) and CNN+bidirectional gated recurrent unit (BiGRU). The advantage of applying deep learning is that we can make use of features such as mel-spectrogram, which requires low reduction of raw data rather than features such as Mel-frequency cepstral coefficients (MFCC). Using less reduced data contributes to performance improvement in speech recognition tasks [13]. Therefore, we expect similar performance improvement in classifying phonemes compared to existing models based on hidden Markov model or the maximum entropy model that use MFCC [9, 12].

### 2.1. Phoneme Classifier Dataset

We used the Seoul Reading Speech (SRS) corpus provided by the National Institute of the Korean Language (NIKL) [11]. The SRS corpus consists of transcriptions without phoneme alignment and 71,216 raw speech audio files, totaling about 180-hours in length. All Sampling rates are at 16000 Hz. The files were recorded by 80 people, equally distributed among the age range of twenties to forties. A total of nineteen different reading texts, each consisting of about 50 sentences were recorded. All audio files were recorded without noise in a laboratory environment. We divided the corpus into train, validation, and test sets at a ratio of 6: 1: 1. All the data used in this study were checked for correctness of transcription. During the checking process, we identified 151 data files which contained miscues. These 151 files were not used for training, but later used as a gold set for the miscue classification step described in section 3.3.

### 2.2. Training of Phoneme Classifier

For the phoneme classification task, three types of CNN-based models were compared. As the input for the models, we used Mel spectrogram with 40 Mel bandwidth. The length of the time window was 5ms, without any overlap between the windows.

The three types of CNN-based classifier architectures are shown in Figure 2. All three architectures have three convolution layers, which are modified convolution networks of the VGG16 architecture [14, 15]. Max-pooling layers were added between convolution layers. Architecture (b) and (c) each added BiLSTM and BiGRU layers between CNN and fully-connected (FC) layers. For all three architectures, clipped rectified-linear unit (clipped-ReLU) was used as an activation function. Equation 1 shows the definition of the ReLU function used.

$$clipped\_ReLU(x) = min\{max\{0, x\}, 20\} \quad (1)$$

We trained our models with an Adam optimizer. To prevent overfitting, we set dropout rate as 20% for each layer to models and randomly added white and pink noises to audio data. Early stopping conditions were 15 epochs.

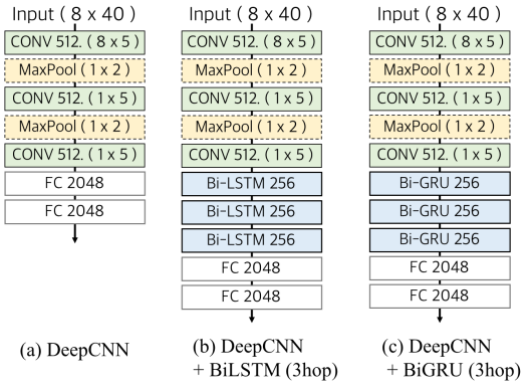


Figure 2: *VGG-based DeepCNN, DeepCNN+BiLSTM and DeepCNN+BiGRU model architecture*

The input of the three CNN-based phoneme models is a 8 x 40 mel-spectrogram that consists of time durations by Mel-bandwidths information. The output of each models is a list of temporal labels of phonemes, which are 5ms time intervals. Although a common time window for phoneme classification task is between 10 to 20ms [18], we used a higher resolution at 5ms since the duration of miscues can be shorter than phonemes, which is the target unit for our classification task. If a phoneme lasts for more than 5ms, a time interval in which the first instance of the phoneme appears, is labeled with the phoneme, as well as the following 5ms time intervals. For example, for the word "기차도", the temporal label is, "'KK', '-', 'IY', '-', '-', 'CH', 'CH', '-', 'AA', '-', 'T', '-', 'OW', '-', '-', '-'." Since we are interested in the first occurrence of a phoneme along with the duration of the phoneme, we want the output to remove duplicates of consecutive phonemes, as shown in the following format, "'KK', 'IY', 'CH', 'AA', 'T', 'OW'." To obtain this desired format without duplicate phonemes, we applied Connectionist Temporal Classification (CTC) decoding [16]. For the application of CTC, we used the notations where $C = \{\text{AA}, \text{AX}, \text{CH}, \text{EH}, \text{EY}, \ldots, \text{blank}\}$ is the set of temporal phoneme labels, and $\text{P} = \{ \cdots \}$ is the set of desired phoneme labels. For each feature vector $c \in C^*$ whose entry $c_t$ denotes the temporal label at time $t$, we applied CTC decoding to output the desired phoneme label $\hat{p} \in P^*$. Note that the length of $\hat{p}$, $l$ is usually less than that of c.

### 2.3. Accuracy of Phoneme Classifier

To measure the accuracy of the phoneme classifier, we measured the Label Error Rate (LER) between the labeled phonemes ($P_i$) and the gold standard set. The phonemes for the gold standard set was obtained by applying the open-source-based Korean grapheme-to-phoneme (G2P) [17] software to the original SRS transcription files. For evaluation, we computed LER score on the set S as follows in Equation 2. Here, let $S \subseteq X \times P^*$ be the set of test examples $(x, p)$ and $\hat{p}$ be the result of phoneme classifier. In addition, $ED(p, q)$ is the edit distance between two sequence p and q. Edit distance is a minimum number of insertions, substitutions and deletions required to change p into q. LER using edit distance is a popular measurement for speech or handwriting recognition tasks. [16]

$$LER(S) = \frac{1}{S}\sum \frac{ED(p, \hat{p})}{len(p)} \quad (2)$$

The accuracies of phoneme classification results for the three types of classifier architectures with corresponding LER scores are shown in table 1. A smaller LER score denotes a higher performance. The experiments on 8,800 test sets show that the model using only DeepCNN yields the best LER. Although, we cannot provide a direct comparison of accuracy between our results and other researchers due to the difference in the datasets, our numbers seem comparable to the state-of-the art performance. Our LER score is about 9% better than the existing state-of-the art performance on the Korean phoneme classification task, which is reported by Minsoo Na et al. (LER: 12) [18].

Table 1: *Phoneme classification results. DeepCNN-BiLSTM / Deep CNN-BiGRU are measured 5 times. Best scores are bolded ± means standard error.*

| | DeepCNN (VGG) | DeepCNN-BiLSTM | DeepCNN-BiGRU |
|---|---|---|---|
| LER | 32.66 | 7.79 ± 1.75 | 3.71 ± 1.68 |

### 2.4. Modified-SRS Corpus using Data Augmentation

Since miscue annotated reading data is required as an input to our design of the miscue classifier, we created an artificial miscue corpus using the SRS corpus. Artificial miscues were made by modifying or deleting a sound between aligned phonemes. Since the phonetic characteristics differ according to the type of miscue, we selected three miscue types and applied corresponding algorithms. Miscue algorithms were applied at the border of the aligned phonemes in the SRS corpus. Figure 3 shows the original waveform without any manipulations of miscue. The following subsections will each describe the applications of the three different miscue types used in our study. These miscues were introduced in Proença et al. [9, 19, 20].

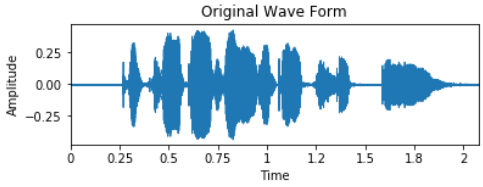


Figure 3*: Original waveform before manipulation*

#### 2.4.1. *EXT (Extension of a Word)*

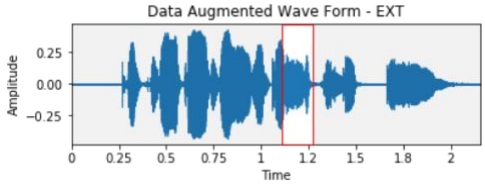


Figure 4: *Manipulated waveform - EXT*

EXT is a miscue in which one phoneme is abnormally extended when reading a text. Given an audio segment, we randomly selected one vowel from the audio segment to create EXT-like data. Using a phase vocoder, the selected vowel was time-stretched. The stretched duration was determined by randomly selecting a number between 1 and 2, at 0.1 interval.

#### 2.4.2. *PAU (Pause in a Word)*

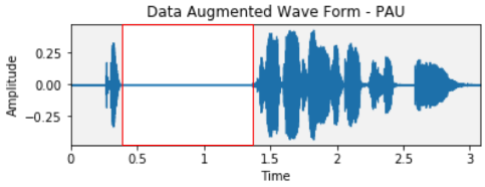


Figure 5: *Manipulated waveform - PAU*

PAU refers to a miscue that stops pronouncing in a very short period of time while reading a text. We have added a silence interval of 0.2 to 1.0 seconds between arbitrary phonemes to create a PAU dataset. Silence interval was generated by mixing zeros and some noise. As a result, a pattern very similar to PAU was produced.

#### 2.4.3. *PRE (Pre-Correction of Pronunciation)*

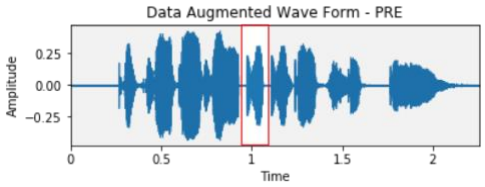


Figure 6: *Manipulated waveform - PRE*

PRE is a miscue in the reading process where the reader recognizes a mistake in pronunciation of a word, then stops reading the word, and starts re-reading the word. To simulate PRE miscue, we transformed an audio file as if pronouncing same syllable twice. Because the consonants were not well detected due to the nature of the sound, we decided to create a PRE, based on a vowel in which the formant was clearly formed and detected relatively well. The artificial miscue was generated by selecting the position of the vowel at random with force-aligned boundaries and generating the corresponding pronunciation again.

## 3. Miscue Classifier

To address our goal of developing a model that detects the three types of miscues (EXT, PAU, PRE) at the phoneme level, we built a miscue classifier that uses phoneme aligned speech data files as an input. Three types of RNN (Recurrent Neural Network) based miscue classifier models have been compared; LSTM, BiLSTM, BiGRU. Figure 7 shows the structure of our miscue classification model.

To train the miscue classifier, we used artificial miscue corpus which is modified from the SRS corpus (modified-SRS) as described in section 2.4. The modified-SRS corpus was divided into training, validation, and test data sets with a 6: 1: 1 ratio, same ratio as in the training process of the phoneme classifier.

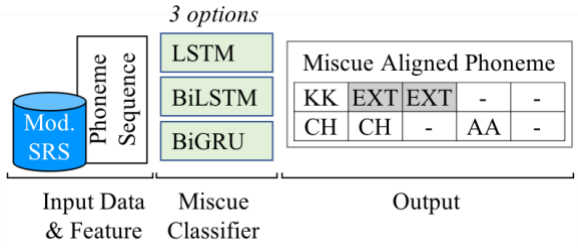


Figure 7: *Miscue classification model pipeline*

For testing the accuracy of miscue classifier, we used two types of data as the test sets; augmented and real. For the augmented data, 12.5% of modified-SRS data saved for testing was used. The real miscue data was obtained during the data cleaning process of the SRS corpus as described in section 2.1. The resulting real miscue data consists of a total of 151 files, including 9 EXTs, 67 PAUs, and 27 PREs. The amount of real data is about 0.002% of the training data.

### 3.1. Training of Miscue Classifier

For the training of miscue classifier, we built three miscue classifiers using different types of RNN models; LSTM, BiLSTM, and BiGRU. The input for the miscue classifiers were the phoneme aligned data of modified-SRS corpus. For all the RNN models, we set the model depth as 2 and set 512 hidden dimensions for each layer. We used RMSProp optimizer for the miscue classifier, and dropout rate was set to 20%. All three models were trained for 3 epochs (approximately 420 hours of speech data).

When the miscue classifier detects a phoneme as a miscue, the classifier replaces the miscue phoneme with a corresponding miscue label as an output. The miscue phoneme can be assigned one of these following four labels; PAU, EXT, PRE and <EOS> (end of string). For example, let's assume that given a temporal label $c_i$ = ('KK', **'-', 'IY'**, '-', '-', 'CH', 'CH', '-', 'AA', '-', 'T', '-', 'OW', '-', '-', '-'.) as an input, a miscue of EXT occurred at location $c_1$ and $c_2$. Therefore, the prediction value $\hat{Y}_i$ of the miscue classifier is as follows; $\hat{Y}_i$ = ('KK', **'EXT', 'EXT',** '-', '-', 'CH', 'CH', '-', 'AA', '-', 'T', '-', 'OW'). After training and building the miscue classification model, two types of test sets were examined; augmented data and real miscue data.

### 3.2. Accuracy of Miscue Classifier

To evaluate the performance of the miscue classifier, we used F1 scores for both augmented and real data types. Of the three types of RNN based models, BiGRU showed the best performance for augmented data, while BiLSTM showed the best performance for real data. Table 2 shows the F1 score of each model.

Table 2: *Overall F1 score overview. Best scores for real and augmented data are bolded.*

| Model | Data | Precision | Recall | Overall F1 |
|---|---|---|---|---|
| LSTM | Aug. | 0.608 | 0.592 | 0.600 |
| | Real | 0.558 | 0.535 | 0.533 |
| BiLSTM | Aug. | 0.627 | 0.627 | 0.618 |
| | **Real** | 0.519 | 0.512 | **0.512** |
| BiGRU | **Aug.** | 0.824 | 0.809 | **0.819** |
| | Real | 0.519 | 0.512 | 0.511 |

Although the amount of real data is too small to make generalizations about the classification results, we wanted to compare the miscue classification results of augmented data results and real data results. Experimental results showed that even though it is a model trained with augmented data, our miscue classifier yields an F1 score of 0.5 or more in real miscue.

In addition, Table 3 shows the F1 score on each miscue types. EXT in real miscue all showed the same value. This seems to be caused by too few samples of the miscue type EXT (9 samples). BiGRU showed the best overall performance for augmented data. On the other hand, the differences in performance were similar across all three models for real data.

Table 3: *Classification results on each miscue. Best scores for each miscue types are bolded.*

| Model | Data | EXT | PAU | PRE |
|---|---|---|---|---|
| LSTM | Aug. | 0.498 | 0.875 | 0.601 |
| | Real | 0.497 | 0.550 | 0.493 |
| BiLSTM | Aug. | **0.613** | 0.903 | 0.751 |
| | Real | 0.497 | 0.540 | 0.495 |
| BiGRU | Aug. | 0.516 | **0.948** | **0.782** |
| | Real | 0.429 | 0.548 | 0.520 |

## 4. Discussion

In this paper, we proposed a method for automatic miscue detection in the absence of miscue labeled data by using data augmentation. Despite the fact that we only trained with artificially generated data, our miscue detection model yields F1 score of about 0.512 for real data and 0.819 for augmented data. A direct comparison of our F1 score with existing work is not possible due to the difference in audio data characteristics. However, as a rough comparison, the current state-of-the art on miscue detection tasks [7, 8] report an F1 score of about 0.6 to 0.65, which is comparable to our score of 0.819 on augmented data. However, for real data we obtained a slightly lower F1 score of 0.512.

Our three proposed directions for increasing the F1 score for real data are as follows. First, since the number of real data available for testing was too small, we would like to collect more real data in the future for further validation on our miscue classifier. Second, increasing the accuracy of miscue types with low classification accuracy is needed. Our miscue classification results show that in all three RNN models, the accuracy of miscue type PAU and PRE are relatively higher than EXT. This trend was observed with the real miscue data as well as with the augmented data. One possible reason for the relatively easy detection of PAU and PRE may be due to the silence in which there is an absence of amplitude volume. Therefore, additional effort for building miscue types without silence is required. Finally, considering that we only generated three types of miscues, we plan to generate additional miscue types using additional methods such as generative adversarial network (GAN).

## 5. Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2017R1D1A1B03034511).

# 6. References


[1] L. S. Fuchs, et al. “Oral Reading Fluency as an Indicator of Reading Competence: A Theoretical, Empirical, and Historical Analysis,” *Scientific Studies of Reading*, vol. 5, no. 3, 2001, pp. 239–56.

[2] Yetta M. Goodman, Dorothy J. Watson, and Carolyn L. Burke, *Reading Miscue inventory: from Evaluation to Instruction*. Katonah, N.Y.: Richard C. Owen Publishers, 2005.

[3] T. Ko et al. “Audio Augmentation for Speech Recognition,*” Proc. Interspeech*, 2015.

[4] W. Hartmann, et al. "Two-Stage Data Augmentation for Low-Resourced Speech Recognition," *Proc. Interspeech*. 2016. pp. 2378-2382.

[5] S. Alharbi, et al. "Detecting Stuttering Events in Transcripts of Children’s Speech." *International Conference on Statistical Language and Speech Processing*. Springer, Cham, 2017. p. 217-228.

[6] Fainberg, Joachim, et al. "Improving Children's Speech Recognition through Out-of-Domain Data Augmentation." INTERSPEECH. 2016. p. 1598-1602.

[7] Yao-Chi Hsu et al. “Mispronunciation Detection Leveraging Maximum Performance Criterion Training of Acoustic Models and Decision Functions,” Proc. Interspeech 2016, pp. 2646-2650.

[8] Hu, Wenping, et al. “Improved Mispronunciation Detection with Deep Neural Network Trained Acoustic Models and Transfer Learning based Logistic Regression Classifiers,” *Speech Communication*, 2015, 67. pp. 154-166.

[9] J. Proença et al. “Detection of Mispronunciations and Disfluencies in Children Reading Aloud,” *Proc. Interspeech* 2017, pp. 1437–1441.

[10] W. Yun et al. “The Korean Corpus of Spontaneous Speech,” *Phonetics and Speech Sciences*, 7(2), 2015, pp. 103-109.

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

[12] Y. Liu et al. “Comparing HMM, maximum entropy, and conditional random fields for disfluency detection,” *Proc. Interspeech* 2005, pp. 3313–3316.

[13] Deng, Li, Geoffrey Hinton, and Brian Kingsbury. "New Types of Deep Neural Network Learning for Speech Recognition and Related Applications: An Overview," *Proc. ICASSP* 2013, pp. 8599-8603.

[14] Heck, Michael, et al. "Ensembles of Multi-scale VGG Acoustic Models." *Proc. Interspeech 2017*, pp. 1616-1620.

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

[16] Graves, Alex, et al. "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," *Proc. ICML*, 2006.

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

[18] 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*, 2016. pp. 131-132.

[19] J. Proença et al. “The LetsRead Corpus of Portuguese Children Reading Aloud for Performance Evaluation,” *Proc. LREC* 2016, pp. 781–785.

[20] J. Proença et al. “Automatic Annotation of Disfluent Speech in Children’s Reading Tasks,” *Proc. IberSpeech* 2016, pp. 172-181.