Speech Emotion Recognition with Text Feature

Korea University COSE461 Final Project

Nokyung Park
Department of Computer Science
Team 14
2018320156

Seongcho Ji
Department of Computer Science
Team 14
2019320005

Jooeun Lim
Department of Computer Science
Team 14
2018320110

Abstract

Knowing the speaker's emotional state through voice is important. Various studies are underway in the Speech Emotion Recognition (SER) field, and in most cases, features are extracted only from audio data. But in this paper, we use text data as well as audio. We propose a model that has three featurizers which respectively receive a Mel-Frequency Cepstral Coefficient (MFCC) vector, text, and Mel-Spectrogram image as input. We set the model using only MFCC features as the baseline, and held a experiment comparing the performance between model variants. We tried adding text features or image features, or both text and image features. The performance appeared to be the best when both text and image features were added. Furthermore, a remarkable point was that the extend of performance improvement was the largest when only text features were added. This paper suggests that text data plays a significant role in SER, and they should not be omitted.

1 Introduction

Capturing the speaker's emotional state from voice is important with broad applications in many tasks. [1] However, SER task remains yet complex to be used in real-life applications. Speech contains emotional information in terms of different forms, which can be categorized into text and non-verbal information. [2] Both parts take account in predicting the emotion, and despite the importance of phonetics that takes the majority part in understanding the emotion of the sound, there is still a possibility that textual cues exist and provide a hint for classifying the emotion.

In this paper, we proposed a model which utilizes both audio and text data. We have raw audio data and corresponding text. MFCC vector and Mel-Spectrogram image data were obtained by applying the MFCC algorithm and Mel-Spectrogram production process to raw audio, respectively. The model receives these preprocessed data as input and those data are fed into three featurizers then combined to create a new feature for emotion prediction. A detailed description of the structure of the model is written in section 3, and a description of data preprocessing in section 4.1.

Our model received text and audio data separately during evaluation. However, in often cases there are no transcriptions provided in the SER datasets, or real-time circumstances. Today, the performance in the field of Automatic Speech Recognition (ASR) reached the level sufficient to be commercialized[3] and text data can be extracted quite accurately without difficulty. Therefore we assumed that text can

be generated by ASR even when there are no given text and that generated text can replace the role of the given one. Thus, our approach is available in various circumstances.

This is the link to our code implementation:
`https://github.com/noparkee/Natural-Language-Process-Team-Project.git`

2 Related Work

2.1 Mel-Frequency Cepstral Coefficient (MFCC)

In SER tasks, MFCC is one of the most popular features used for extracting audio features. [4][5] The concept of MFCC is imitating the behavior of human ears of how they process the incoming sound waves: be more sensitive in lower frequency domain than higher. To generate MFCC, several steps are needed to be taken: Apply the Fourier Transform to the input signal, map the power of the spectrum to the Mel scale, take logs, take Discrete Cosine Transform (DCT), and then finally, convert it back to the time domain. [6]

2.2 Mel-Spectrogram

A spectrogram is a time-frequency representation of an audio signal. In specific, Mel-Spectrogram is a intermediate state of MFCC production before taking DCT. Mel-Spectrograms are commonly represented as images, and these images can be fed to deep learning networks such as CNN or RNN for further procedure. [7] [8]

2.3 Self-Attention and Convolutional Self-Attention(CSA)

Self-Attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. [9] Self-Attention method is popular approach in diverse NLP tasks and now also common in vision tasks. [10] Many models for SER also adopt self-attention mechanism. [5][11][1]

CSA network computes attention map A from original feature y, then produces new feature $\hat{y}$ with learnable scaling factor γ. The original feature y is fed into three convolutional layer with kernel size of 1.

$$j(y) = W_j y,\ k(y) = W_k y,\ l(y) = W_l y, \tag{1}$$

where j, k, l is new feature spaces. Note that transform W_j, W_k reduce the number of channels to one-eighth, while W_l leaves it. The attention energy matrix E is calculated by softmax of dot product between j and k.

$$E = softmax(j(y)^{\mathbf{T}} k(y)) \tag{2}$$

Then attention map A is calculated by multiplication of E and $l(y)$.

$$A = l(y)E \tag{3}$$

This attention map is scaled with γ then added to y. A new feature $\hat{y}$ is calculated as:

$$\hat{y} = y + \gamma A, \tag{4}$$

where γ is initialized with zero.

2.4 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a pre-trained model based on a very large transformer that learns general understanding on the language from large corpus of plain texts, with an unsupervised, and deeply bidirectional manner. BERT can be fine-tuned to be used for task-specific objectives. [12]

3 Approach

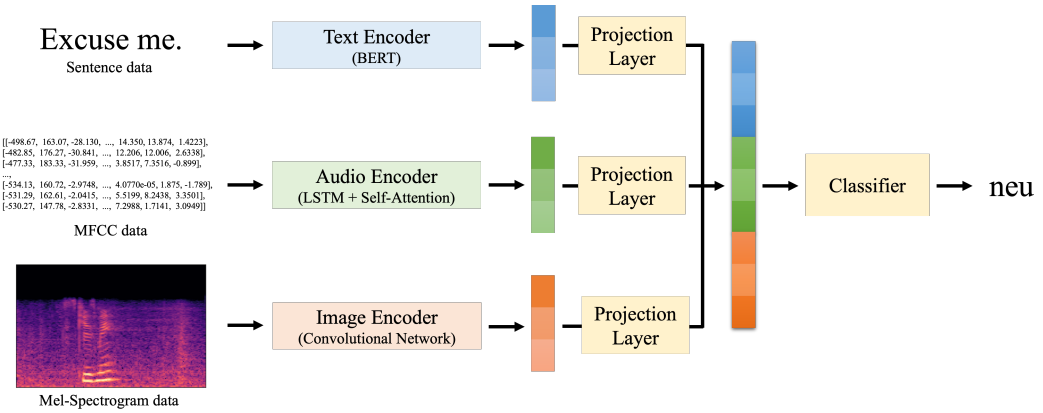

Figure 1: An overview of our proposed model, which consists of three featurizers and one classifier.

Our model uses three types of data: MFCCs, sentences, and Mel-Spectrogram images. Each data pair contains one sentence, one MFCC vector, and one Mel-Spectrogram image. MFCC vectors are created by applying MFCC algorithm to the wav files corresponding to each sentence. Mel-Spectrogram images are created by having the intermediate signals in MFCC processed as images. More details on data processing procedure are described in section 4.1.

The model consists of three featurizers which are audio, text, and image featurizer, followed by dense projection layers, and they are connected to one final classifier. Audio, text, image features are extracted from the input data, as the input passes through each featurizer. Subsequently, features in different embedding spaces are transmitted through each respective projection layer to be mapped to the same embedding space. The model then generates a new feature by concatenating three features in that embedding space. This final feature is passed to the softmax layer and then fed as an input to the classifier to predict emotion. The prediction score is calculated using cross entropy loss. For sparse prediction, halved squared mean of the output vector is added to the loss.

3.1 Text Featurizer

We used pre-trained BERT large model with fine-tuning of 4 epochs. for the text featurizer to get text feature. The sentence is the input of BERT and we use the embedding layer of BERT as a text feature.

3.2 Audio Featurizer

Our audio featurizer has two BiLSTM layers, receives MFCC data as an input and produces MFCC embedding vector as an output. Since MFCC is variable-length data, the featurizer uses RNN-based networks such as LSTM. The featurizer applies self-attention to the output of first BiLSTM layer. Then attention-value is fed into the second BiLSTM layer and the last hidden state of the layer becomes the audio feature.

3.3 Image Featurizer

In the image featurizer, input images go through the 3x3 convolution layer, ReLU layer, and 2x2 max-pooling layer sequentially. We define this process as an unit routine. There are six routines

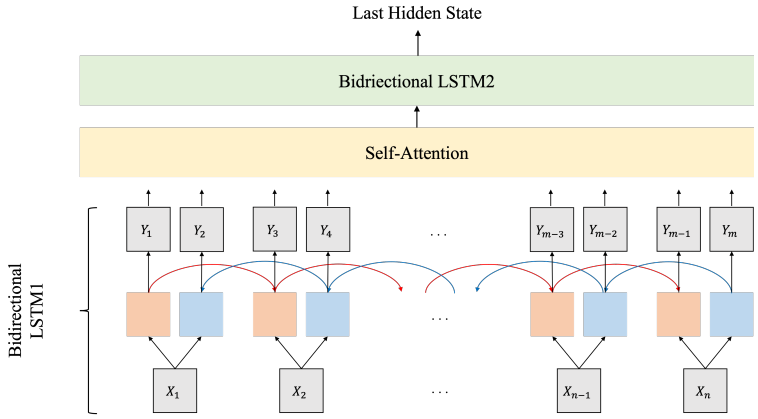

Figure 2: An overview of the audio featurizer in our proposed model, which consists of two Bidirectional Long Short-term Memory convolution layers (LSTM). It compute self attention value between the two layers.

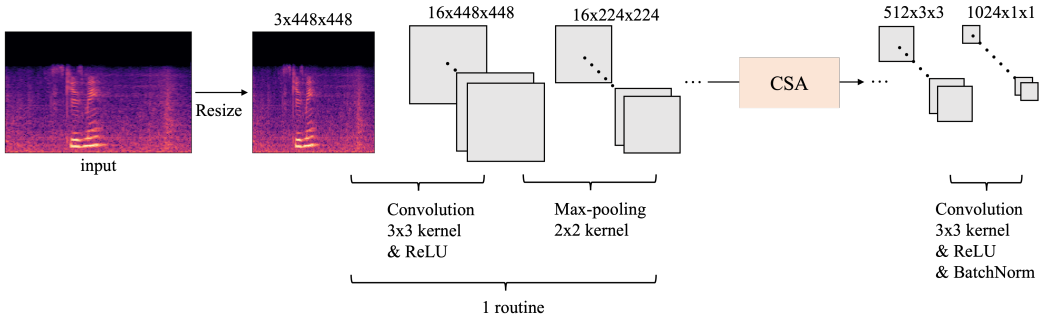

Figure 3: An overview of the image featurizer in our proposed model, which consists of seven convolution layers, three batch normalization layers and six max-pooling layers. This figure shows a simplified version of our image featurizer.

in the model, and after the sixth routine, convolution layer, ReLU layer, and batch normalization layer comes afterward in order. Between the fifth routine and the sixth routine, there is a process of obtaining self-attention. There are three batch normalization layers after the second routine, the fifth routine, and the last ReLU layer.

4 Experiments

4.1 Data

The IEMOCAP[13] dataset is used for training and evaluation. The corpus contains approximately twelve hours long audio-visual data of conversations with ten speakers. The entire dataset is divided into 5 sessions, each session containing dialog from a pair of male and female speakers, along with transcriptions. Every dialog is separated into unit sentences and they are named as segments. Each segment is labeled with nine emotions and additional two labels: *happy, angry, excited, fear, sad, surprised, frustrated, disappointed, neutral*, (additional) *xxx and other*.

Following previous SER researches based on IEMOCAP [5][14], we use IEM4, which is IEMOCAP evaluated with only four classes. IEM4 combines *happy* with *excitement*, and does not use *angry, fear, surprised, frustrated, disappointed, xxx, other*

Label	#
hap	1636
sad	1084
ang	1103
neu	1708
total	5531

Table 1: *Number of samples of each emotion label in IEM4*

MFCC and Mel-spectrogram features are extracted from raw audio vector using Python Librosa [15] library with sample rate of 44100. We set n_mfcc as 20 which are used in MFCC, and the number of Mel-filters as 256 which are used in Mel-spectrogram.

The sequence length of tokenized text and audio feature vector varies. In each mini-batch, both audio and text data are padded to equally be the maximum length.

4.2 Evaluation method

In IEMOCAP, the numbers of the samples between classes are imbalanced. Therefore, the evaluation scores use both unweighted accuracy (UA) and weighted accuracy (WA).

$$UA = \frac{\#\,of\,correct\,instances}{\#\,of\,total\,instances},\ \ WA = \frac{1}{C}\sum_{i=1}^{C}\frac{\#\,of\,correct\,class{-}i\,instances}{\#\,of\,class{-}i\,instances},$$

where C is the number of classes.

4.3 Experimental details

The model is constructed and trained by using PyTorch[16] framework. We used Adaptive Moment Estimation (Adam) optimizer [17] to train the model, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 0, learning rate of 0.001 and mini-batch size of 32. The number of epoch is 4. For each model configuration in table 2, we trained the model 5 times, and selected the model which had the best score.

4.4 Results

Table 2: This table shows the accuracy of our model. All of the scores in the table are the values rounded off to the third decimal place. In this table, *audio*, *text*, and *images* refer to three features created by the three featurizers respectively.

Model	UA(%)	WA(%)
(1) : audio	51.47	52.75
(2) : audio + text	**68.29**	69.2
(3) : audio + image	51.01	53.12
(4) : audio + text + image	68.2	**71.02**

Table 2 shows us the performance of four versions of our model. Model (1) shows the worst performance, while Model (4) shows the best performance. Model (2) is a model which have text features corresponding to the wav files added from Model (1), and Model (3) is a model which have Mel-spectrogram image features corresponding to the wav files added from Model (1). Model (2) achieved about 17% performance improvements compared to Model (1), while Model (3) has almost same performance. And all of our models achieve higher WA score than UA.

5 Analysis

According to Table 2, model (2) and (4) have recorded higher both UA and WA compared to the models without having texts as its features. This shows that text had an apparent role in detecting the emotion. Moreover, in cases of comparing model (2) and (3), adding image features improved the performance as well, but adding text had a higher increment in performance compared to adding the image. This implies that if there already were some audio feature, adding textual feature is more efficient than adding supplementary audio feature. Consequently, exploiting all those features marked the highest score in terms of WA. The experiment results reveal that text and audio feature both are influential in better prediction, especially with text performing better.

6 Conclusion

We attempted to improve the SER model performance via vectorizing text data through BERT, as well as processing audio data in various ways, such as applying self-attention or converting it into an image. This approach led to significant performance improvement. In particular, when text features were added to the model, the model performance improved the most: UA(WA) is increased by average 17%p. This indicates that the content of the text features contain significant information that their importance should not be neglected in SER tasks. Therefore, we conclude that using text data is necessary as well as audio data in the SER field. However, in cases where the dataset does not have transcription, ASR conversion from speech data to text works well in these days, so it is reasonable to replace transcription data with extracted text data produced by ASR. Although our model's performance is not as good as the state-of-art (SOTA) model's, we suggest that it is worth to add text-based methods with ASR to the audio-only models, including the SOTA model, and we expect those to increase their performance.

References

[1] Zheng Lian, Jianhua Tao, Bin Liu, Jian Huang, Zhanlei Yang, and Rongjun Li. Context-Dependent Domain Adversarial Neural Network for Multimodal Emotion Recognition. In *Proc. Interspeech 2020*, pages 394–398, 2020.

[2] Bagus Tris Atmaja, Kiyoaki Shirai, and Masato Akagi. Speech emotion recognition using speech feature and word embedding. pages 519–523, 11 2019.

[3] Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504*, 2020.

[4] T. Adam and Md Sah Salam. Spoken english alphabet recognition with mel frequency cepstral coefficients and back propagation neural networks. *International Journal of Computer Applications*, 42:21–27, 03 2012.

[5] Md. Asif Jalal, Rosanna Milner, and Thomas Hain. Empirical interpretation of speech emotion perception with attention based model for speech emotion recognition. In *INTERSPEECH*, 2020.

[6] K.V. Krishna Kishore and P. Krishna Satish. Emotion recognition in speech using mfcc and wavelet features. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 842–847, 2013.

[7] Kannan Venkataramanan and Haresh Rengaraj Rajamohan. Emotion recognition from speech, 2019.

[8] Aharon Satt, Shai Rozenberg, and Ron Hoory. Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms. In *Proc. Interspeech 2017*, pages 1089–1093, 2017.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[10] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.

[11] Md Asif Jalal, Roger K Moore, and Thomas Hain. Spatio-temporal context modelling for speech emotion classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 853–859, 2019.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[13] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359, 2008.

[14] Rosanna Milner, Md Asif Jalal, Raymond WM Ng, and Thomas Hain. A cross-corpus study on speech emotion recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 304–311. IEEE, 2019.

[15] librosa-0.8.1.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Appendix: Team contributions

Common: Building networks, model testing and configuration, writing the paper.

Nokyung: *Main contribution, writing codes of training and model part, image preprocessing.

Seongcho: Making dataloader, text preprocessing, making CSA network.

Jooeun: Audio preprocessing, making dataloader, paper examination and correction.