Emotional Feature Fusion Network Using MELD

Dohoon Kim
Department of Data Science
Hanyang University
kdhyu@hanyang.ac.kr

Nayeon Park
Department of Data Science
Hanyang University
nayeon03@hanyang.ac.kr

Seyong Park
Department of Data Science
Hanyang University
sayong123@hanyang.ac.kr

Abstract

Depression is one of the most severe mental diseases in post pandemic society. Early detection of depression is good. However, one of the main differences between physical diseases and mental diseases is that there's no clear symptoms. Instead, dealing with Emotion Recognition in Conversation (ERC) task can be lead to the foundation for digital healthcare. In this paper, we suggest the simplified feature fusion network for ERC. While test is around 50%, we observed that train, valid and test performaces are very similar, which means model can generalize emotion recognition if trained properly. We hope that if model captures emotions well, it can contribute to the mental healthcare field. More details are available at https://github.com/kdh-yu/AI_ERC.

1 Introduction

Emotion Recognition in Conversation(ERC) is the task to identify the emotion of each utterance from several pre-defined emotions, using given transcript of a conversation along with speaker information of each constituent utterance. Since it uses multimodal data, it is very challenging to compress information while excluding unwanted noises.

1.1 Problem definition

According to OECD, the incidence rate of major depressive disorder, which includes depression and anxiety disorder, has been doubled more than twice. Among them, Korea ranked first in the prevalence of depression (36.8%). It is shocking that 4 out of 10 Koreans feel depressed or has depression. However, the rate of treatment is very low. One of the reasons might be the lack of clear symptoms. Main difference between mental disorder and physical disease is that it only exists in the form of spectrum. All of us have symptoms, but it does not necessarily lead to the diagnosis itself. Rather, psychiatrists diagnose their patients based on their subjective decision. This is hard for people who don't have knowledge about mental disease to be treated at the right time.
Also, the degree of feeling depressed varies from person to person, so it is challenging for individuals to recognize themselves as depressed. Moreover, even if they do recognize it, they may prefer not to expose themselves as having a mental illness due to the stigma and judgmental attitudes associated with it. These characteristics often create reluctance in individuals to seek diagnosis and treatment by visiting hospitals voluntarily. Therefore, the ultimate goal is to develop a model for detecting depression using multi-modal approaches, allowing individuals to receive diagnosis individually without psychological pressure.

1.1.1 MELD

MELD [1], Multimodal EmotionLines Dataset, not only contains the same dialogue instances available in EmotionLines, but also encompasses audio and visual modality along with text. It is created from Friends TV series, with more tan 1400 dialogues and 13000 utterances. Unlike other

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

datasets like IEMOCAP, MELD contains real-world data; not only noises, but also not face-focused video, too. That's why it is very challenging task. According to Paperswithcode, the state-of-the-art benchmarks is recorded with 68.280% accuracy and 69.27% weighted-F1 score.

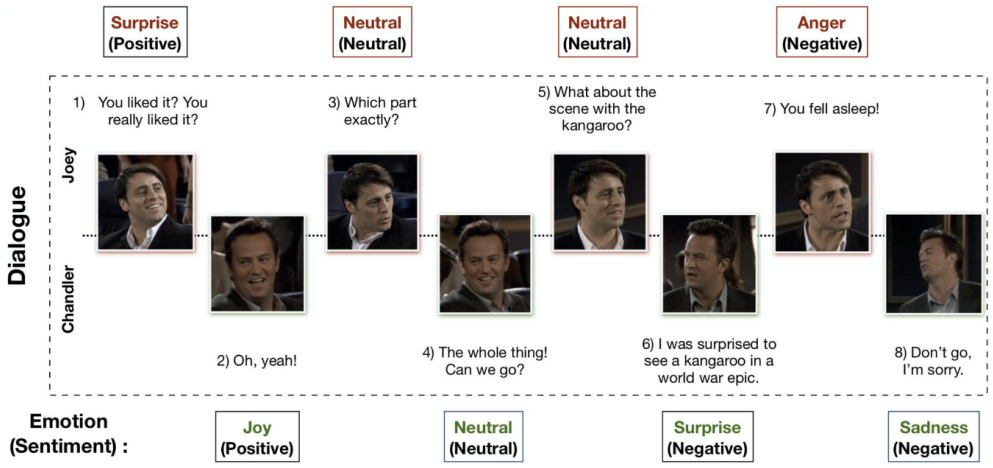


Figure 1: Overall view of MELD.

1.2 Contributions

For that, effective feature fusion method is required. We tried to solve this problem. So in short, our contributions are these; 1. We suggest the appropriate backbone models to perform a good feature extraction. 2. We suggest lightweighted feature fusion method, based on the research DF-ERC [2].

2 Related Works

Depression detection model. Yoo and Oh [3], successfully proposed a model that fuses two modalities with speaker's text and voice signals and detects depression. Features of voice signal were extracted using CNN, and text data using Transformer. However, their research has the limitation that they could not connect voice conversation systems. If built, they could suggest the possibilities for untact diagnosis system.

Depression detection from textual data. Amna Amanat el al [4], proposed a model by implementing the LSTM model, consisting of two hidden layers with RNN with two dense layers. They trained RNN on textual data to identify depression from text, semantics, and written content and achieved 99.0%accuracy, which is higher than using CNN, SVM, and Decision Trees. To evaluate models, they used various evaluation criterions such as precision, recall, f1-measure, support, and accuracy. They used many different approaches like SVM, naive bayes, and one-hot-encoding, and compared each accuracy to improve the model.

Multi-Modal fusion emotion recognition model. Liu el al [5], creatively proposed a model that consists of three parts; Feature extraction, Feature selection, Emotion classification. To extract each feature, the CNN-LSTM is used for the voice and Inception-Res Net-v2 is used for the facial expression in the video. To select features, LSTM is used to capture the correlation between different modalities and within the modalities. The classifier LIBSVM is used for the final five emotions recognition. However, their research has the limitation that the model's recognition accuracy has a quite large interval depending on each emotion. In addition, it struggled with handling the complexity in the recognition process due to redundancy of facial data.

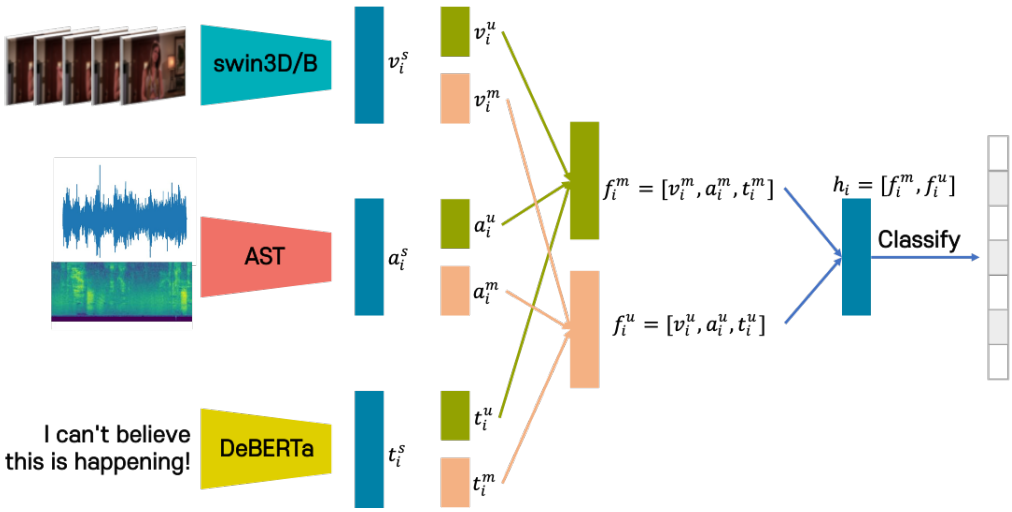


Figure 2: Overall pipeline of our model. After preprocessing each data, feature fusion network properly modifies feature vectors into two level. Separated features are concatenated and passed to the linear classifier to get the final decision.

3 Methods

3.1 Backbones

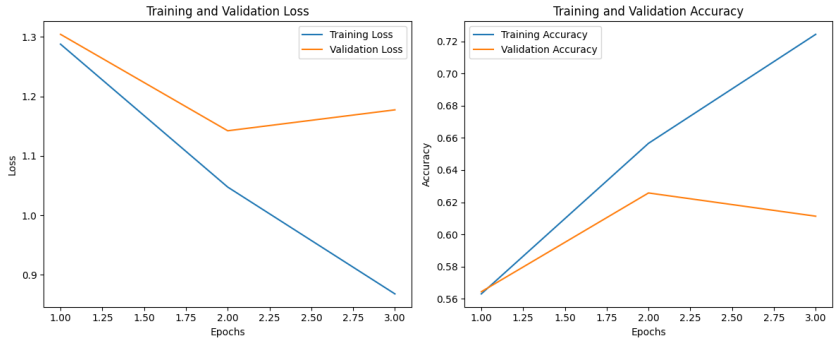
We used the pretrained feature extractors for each modality. To get the high-quality features, we used swin3D/B, AST, and DeBERTaV3.

Video. Swin3D/B [6] is a transformer based model that globally connects patches of videos across the spatial and temporal dimensions. In ERC task, it is very important to capture overall situations, so neither late fusion nor early fusion methods are used, to remain the temporal information. The data shape is $[B \times T \times C \times H \times W]$, where B is batch size, T is the time axis (the number of frames), and C,H,W for the image of that frame.

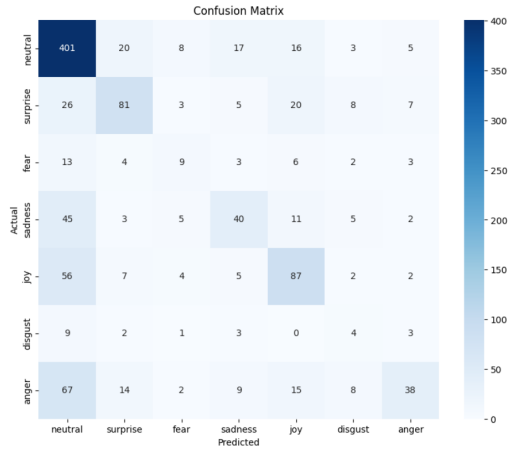
Audio. AST [7] is the first model that purely uses self-attention mechanism to deal with auditory data. It achieved state-of-the-art performance on various benchmarks such as AudioSet, ESC-50, and Speech Commands V2. After changing waveforms into mel spectrogram, AST uses attention mechanism to capture global context. More specifically, audio waveform with shape $[B \times C \times Sr]$, where C stands for channel and Sr for sampling rate, is changed into mel spectrogram with shape $[B \times F \times T]$. Converted mel spectrogram is used as input to the AST, and the AST model generate an embedding vector of size $[B \times D]$.

Text. DeBertaV3 model [8] is an improved model for the defects of the existing BERT [9] and DeBERTa [10] models. The existing DeBERTa model uses MLM to share the embeddings. Unlike the MLM used in BERT, the replaced token detection (RTD) borrowed from the ELECTRA model is additionally used. The generator uses MLM like the existing model, but the discriminator uses a binary classifier of token-level.

However, even in this ELECTRA's method [11], the MLM used in generator training tries to pull semantically similar tokens close to each other, whereas RTD of the discriminator runs to separate them to optimize binary classification accuracy, creating a negative correlation. As a result, a "tug-of-war" phenomenon occurs, which reduces both training efficiency and model quality. Gradient-disentangled embedded sharing (GDES) is used to improve the flaws in these existing models in DeBERTaV3. The generator still shares the discrete embedding, but it blocks the discrete from sharing the gradient to the generator, preventing "tug-of-war". This method maintains a convergence speed similar to that without embedding and the efficiency obtained by embedding.
The data shape is $[B \times C \times L]$, Where B is batch size, C is the channel, L is length of input text, and the DeBERTaV3 generate an embedding vector as $[B \times D]$.



Getting Predictions: 100%| 139/139 [00:04<00:00, 32.10it/s]



	precision	recall	f1-score	support
neutral	0.85	0.85	0.74	470
surprise	0.62	0.54	0.58	150
fear	0.28	0.23	0.25	40
sadness	0.49	0.36	0.41	111
joy	0.56	0.53	0.55	163
disgust	0.12	0.18	0.15	22
anger	0.63	0.25	0.36	153
accuracy			0.60	1109
macro avg	0.48	0.42	0.43	1109
weighted avg	0.59	0.60	0.57	1109

3.2 Feature Evaluation

The performance of the DeBERTaV3 model was evaluated on the text data with the highest proportion. The validation accuracy was calculated and visualized.

Good feature means the separability of them. If they form clusters or separable linearly, it can be concluded that they are good features. To check this, we performed two task; t-SNE analysis and training linear classifier.

t-SNE is the most intuitive way to check whether features are separable or not.

We observed that while text features form clusters, video and audio features do not tend to form

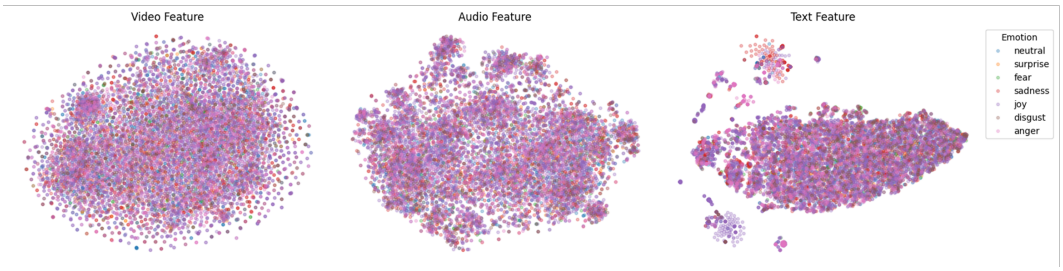


Figure 3: t-SNE results of features without any modification, which means they are totally produced by backbone models.

clusters. We suspected that relatively high noises and unrelated information are included very high, while text only includes utterance-level information only.
Linear Classifier. This problem is also revealed when tested by linear classifier.

Table 1: Linear Classifier Results

Modality	Train Accuracy (%)	Valid Accuracy (%)
Video	48.28	36.70
Audio	47.59	37.51
Text	95.50	60.32

3.3 Feature Fusion

Based on problems above, we built two types of model. First one is linear classifying model, which maps concatenated features to 7 emotions. Actually, it is the simplest way to use features altogether. Second one is the model that uses the idea of DF-ERC [2]. At first we project features and concatenate them. Let i^{th} extracted features as v_i, a_i, and t_i. Features are projected as below.

$$v_i^m / a_i^m / t_i^m = \text{MLP}^m_{v/a/t}(v_i / a_i / t_i) \tag{1}$$

$$v_i^u / a_i^u / t_i^u = \text{MLP}^u_{v/a/t}(v_i / a_i / t_i) \tag{2}$$

where m stands for modality and u for utterance. After projection, features are concatenated in alternating order.

$$f_m = [v_1^m, a_1^m, t_1^m, v_2^m, a_2^m, t_2^m, ...] \tag{3}$$

$$f_u = [v_1^u, a_1^u, t_1^u, v_2^u, a_2^u, t_2^u, ...] \tag{4}$$

Using concatenated vectors, contrastive loss is calculated. -1 is multiplied to mode-matching features. This loss makes features with same modality/utterance closer and the others farther. By this multiplication, model can maximize cosine similarity with same mode and minimize others without changing loss minimizing strategy.
And finally, we get the prediction of this model.

$$\hat{y} = \textsc{Linear}([f_m, f_u])$$

4 Results

4.1 Implementation Details

We used AdamW optimizer with learning rate 1e-4. We trained 10 epochs on L4 GPU of colab. As we explained above, CrossEntropyLoss and ContrastiveLoss are used.

Table 2: Model Evaluation on testset

Model	Accuracy(%)	Weighted-F1 Score(%)
Vanilla Model	33.36	29.52
FFN (Ours)	46.93	36.37

4.2 Evaluation

Vanilla model tended to be overfitted to train data. While train accuracy got 61.72%, valid accuracy was 40.67% and test accuracy was 33.36%. Although we just performed this experiment as a comparison, we were able to say that feature fusion is essential, not a choice.
Also we could figure out that our feature fusion method is very stable. At epoch 10 model got the train accuracy 53.73% and valid accuracy 44.27%. When we tested this model, it got the test accuracy 46.93%. What we focused on is that there's relatively low difference between train and valid/test. It might be hasty, but here we could conclude that our suggesting method can achieve generalizability at least in MELD.

5 Conclusion

5.1 Discussion

The limitation of our project is that the model is still too heavy, which leads to longer processing times and higher computational resource requirements.
Also, our model's current approach to handling missing data is insufficient. In the MELD dataset, the number of data labeled as neutral was higher compared to the data labeled as positive or negative. However, the process of handling this imbalanced dataset was inadequate in our project. The model needs a more robust mechanism to effectively manage to improve accuracy in its predictions.

5.2 Future Works

In future research, it would be better to handle the issue of imbalanced data distribution. The problem of imbalanced data significantly affects the performance of the model, so it is crucial to address adequately. Potential solutions include data resampling, adjusting class weights, or employing data augmentation techniques.
Furthermore, while we did not fully implement our final disentanglement model, we anticipate that achieving an efficient implementation would considerably enhance performance. Disentanglement models help in separating various factors within data, leading to improve the model's accuracy and generalization capabilities.
Successfully performing the ERC task would greatly contribute to the development of the depression model which we initially planned. Emotion recognition models can capture subtle emotional changes expressed in conversations, and by analyzing these changes, it is possible to more accurately detect mental health issues such as depression.

Acknowledgement

This work is done in CSE4007 Artificial Intelligence class, and supported by professor Sungyong Baik.

References

[1] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. Meld: A multimodal multi-party dataset for emotion recognition in conversations, 2019.

[2] Bobo Li, Hao Fei, Lizi Liao, Yu Zhao, Chong Teng, Tat-Seng Chua, Donghong Ji, and Fei Li. Revisiting disentanglement and fusion on modality and context in conversational multimodal emotion recognition, 2023.

[3] Hyewon Yoo and Hayoung Oh. Depression detection model using multimodal deep learning. *Preprints*, May 2023.

[4] Amna Amanat, Muhammad Rizwan, Abdul Rehman Javed, Maha Abdelhaq, Raed Alsaqour, Sharnil Pandya, and Mueen Uddin. Deep learning for depression detection from textual data. *Electronics*, 02 2022.

[5] Dong Liu, Zhiyong Wang, Lifeng Wang, and Longxi Chen. Multi-modal fusion emotion recognition method of speech expression based on deep learning. *Frontiers in Neurorobotics*, 15, 2021.

[6] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer, 2021.

[7] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer, 2021.

[8] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2023.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[10] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

[11] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

Appendix

A. Calculate Loss from related papers

[8] [10]

$$Q_c = HW_{q,c}, K_c = HW_{k,c}, V_c = HW_{v,c}, Q_r = PW_{q,r}, K_r = PW_{k,r} \tag{5}$$

$$\tilde{A}i,j = \underbrace{Q_i^c {K^c}^\top j}_{\text{(a) content-to-content}} + \underbrace{Q_i^c {K^r}^\top \delta(i,j)}_{\text{(b) content-to-position}} + \underbrace{K_j^c {Q^r}^\top \delta(j,i)}_{\text{(c) position-to-content}} \tag{6}$$

$$H_o = \text{softmax}\left(\frac{\tilde{A}}{\sqrt{3d}}\right) V_c \tag{7}$$

$$L_{MLM} = \mathbb{E}\left(-\sum_{i \in C} \log p_{\theta_G}\left(\tilde{x}_{i,G} = x_i \mid \tilde{X}_G\right)\right) \tag{8}$$

$$L_{RTD} = \mathbb{E}\left(-\sum_{i} \log p_{\theta_D}\left(1(\tilde{x}_{i,D} = x_i) \mid \tilde{X}_D, i\right)\right) \tag{9}$$

$$g_E = \frac{\partial L_{MLM}}{\partial E} + \lambda \frac{\partial L_{RTD}}{\partial E} \tag{10}$$

B. Improvement Ideas to Address Imbalanced Classification

In this study, the neutral label accounted for a significant portion compared to the other six labels. To address this data imbalance, feature extraction was performed using two methods: for text data, assigning weights to the classes and increasing the insufficient data sets through oversampling. The results of these methods are presented below.

Using the extracted features from these two methods, it is expected that the model will achieve higher performance when running the classifier.

C. All figures related to this work

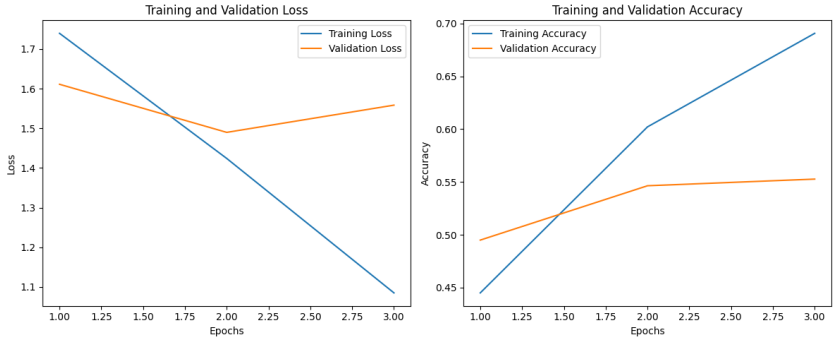


Figure 4: Training and Validation Accuracy and Loss (Weighted Class Labels)

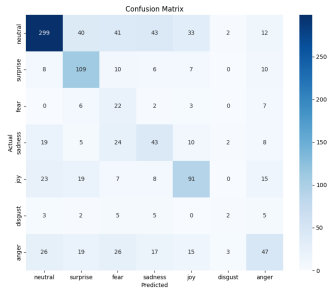


Figure 5: Confusion Matrix (Weighted Class Labels)

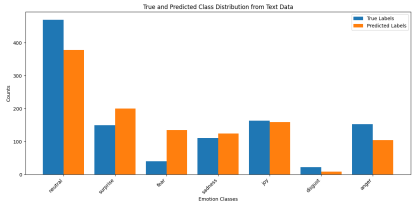


Figure 6: Class Distribution (Weighted Class Labels)

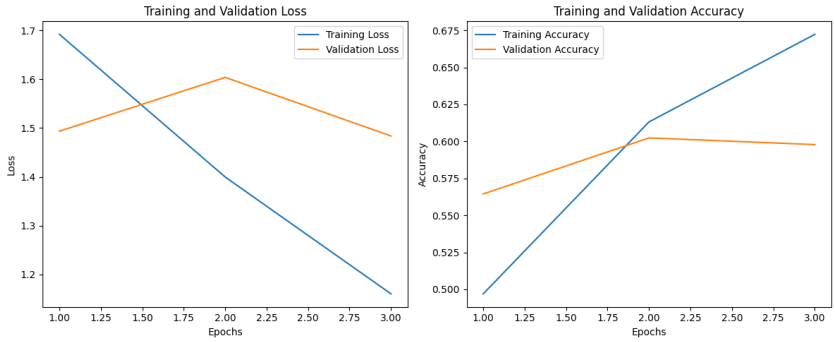


Figure 7: Training and Validation Accuracy and Loss (Oversampling)

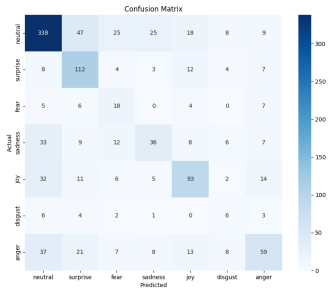


Figure 8: Confusion Matrix (Oversampling)

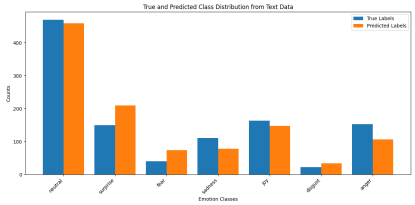


Figure 9: Class Distribution (Oversampling)

D. Code Access

All of our codes are accessible at here;

- https://colab.research.google.com/drive/1U8ZLysbFu-1tVWyK1v8GO6P980Lc8PBK?usp=sharing
- https://colab.research.google.com/drive/1BxSVHh28eVNZce4DilrZ0RVqkfV54KlM?usp=sharing
- https://github.com/kdh-yu/AI_ERC