

# ELEC 5305 Project Report



**Name** Jiaqi Zhao  
**SID** 510526775

## Introduction

Speech Emotion Recognition (SER) has become popular in recent years because of its potential uses in a wide range of fields. Encompassing customer service, healthcare, and human-computer interaction.

The Odyssey dataset is used in this study. It includes a variety of speech samples that have been annotated for emotional characteristics like dominance, valence, and arousal. A broad range of emotional expressions is ensured by the dataset's inclusion of recordings from several speakers. We use the current models and assessing the accuracy of the model with the Odyssey dataset.

Presenting the techniques used to refine and assess the SER models on the Odyssey dataset is the main goal of this research. The ramifications of our findings and how they can affect

further study in this field will also be covered. Also contrasting the results of conventional emotion recognition with predictions of arousal, valence, and dominance.

## Code

### 1. Arousal Model Evaluation Code

```
1 import os
2 import subprocess
3
4 def run_evaluation(seed, ssl_type, pool_type):
5     model_path = f"C:/Users/user/Desktop/MSP/model/dim_aro_ser/wavLM_adamW/{seed}"
6     store_path = f"C:/Users/user/Desktop/MSP/result/dim_aro_ser/wavLM_adamW/{seed}.txt"
7
8     os.makedirs(os.path.dirname(store_path), exist_ok=True)
9
10    try:
11        subprocess.run([
12            "python", "eval_aro_dim_ser.py",
13            "--ssl_type", ssl_type,
14            "--pooling_type", pool_type,
15            "--model_path", model_path,
16            "--store_path", store_path
17        ], check=True)
18
19        print(f"Evaluation for seed {seed} completed successfully. Results saved to {store_path}.")
20    except subprocess.CalledProcessError as e:
21        print(f"Error occurred: {e}")
22
23    if __name__ == "__main__":
24        ssl_type = "wavlm-large"
25        pool_type = "AttentiveStatisticsPooling"
26        seed = 7
27
28        run_evaluation(seed, ssl_type, pool_type)
29
```

The arousal model evaluation is carried out in this code using the `run_evaluation` function. Both the model path and the result storage path are set by the code. Confirming the existence of the storage directory as well. `Subprocess.run` calls the evaluation script `eval_aro_dim_ser.py` and passes it the necessary parameters, such as the model path, pooling type, and SSL type. The output results are saved to the designated path once the evaluation is finished.

### 2. Categorical Emotion Model Evaluation Code

```
1 import os
2 import subprocess
3
4 def run_training_and_evaluation(seed, ssl_type, pool_type):
5     model_path = f"C:/Users/user/Desktop/MSP/model/weight_cat_ser/weight_cat_ser/wavLM_adamW/{seed}"
6     store_path = f"C:/Users/user/Desktop/MSP/result/weight_cat_ser/wavLM_adamW/{seed}.txt"
7
8     os.makedirs(os.path.dirname(store_path), exist_ok=True)
9
10    try:
11        subprocess.run([
12            "python", "eval_cat_ser_weighted.py",
13            "--ssl_type", ssl_type,
14            "--pooling_type", pool_type,
15            "--model_path", model_path,
16            "--store_path", store_path
17        ], check=True)
18
19        print(f"Evaluation for seed {seed} completed successfully. Results saved to {store_path}.")
20    except subprocess.CalledProcessError as e:
21        print(f"Error occurred: {e}")
22
23    if __name__ == "__main__":
24        ssl_type = "wavlm-large"
25        pool_type = "AttentiveStatisticsPooling"
26        seed = 7
27
28        run_training_and_evaluation(seed, ssl_type, pool_type)
29
```

This code is similar to the evaluation code of the arousal model. It is used to evaluate the categorical emotion model. It sets the model and result paths. Then run the evaluation script `eval_cat_ser_weighted.py`. Again, all necessary parameters are passed to ensure that the evaluation script can access the model and storage path.

### 3. Dominance Model Evaluation Code

```
1  # -*- coding: UTF-8 -*-
2  import os
3  import subprocess
4  import argparse
5
6  def run_evaluation(seed, ssl_type, pool_type):
7      model_path = f"C:/Users/user/Desktop/MSP/model/dim_dom_ser/wavLM_adamW/{seed}"
8      store_path = f"C:/Users/user/Desktop/MSP/result/dim_dom_ser/wavLM_adamW/{seed}.txt"
9
10     os.makedirs(os.path.dirname(store_path), exist_ok=True)
11
12     try:
13         subprocess.run([
14             "python", "eval_dom_dim_ser.py",
15             "--ssl_type", ssl_type,
16             "--pooling_type", pool_type,
17             "--model_path", model_path,
18             "--store_path", store_path
19         ], check=True)
20
21         print(f"Evaluation for seed {seed} completed successfully. Results saved to {store_path}.")
22     except subprocess.CalledProcessError as e:
23         print(f"Error occurred: {e}")
24
25 if __name__ == "__main__":
26     ssl_type = "wavlm-large"
27     pool_type = "AttentiveStatisticsPooling"
28     seed = 7
29     run_evaluation(seed, ssl_type, pool_type)
```

The prevailing emotion model is assessed using this code. Setting the model path and result path and executing the script are similar to the code of the first two models. `Subprocess.run` is used to run the evaluation and store the findings.

### 4. Multi-task Emotion Attributes Model Evaluation Code

```
1  import os
2  import subprocess
3
4  def run_evaluation(seed, ssl_type, pool_type):
5      seed = 7
6
7      model_path = f"C:/Users/user/Desktop/MSP/model/dim_ser/wavLM_adamW/{seed}"
8      store_path = f"C:/Users/user/Desktop/MSP/result/dim_ser/wavLM_adamW/{seed}.txt"
9
10     print(f"Model path: {model_path}")
11     print(f"Store path: {store_path}")
12
13     os.makedirs(os.path.dirname(store_path), exist_ok=True)
14
15     try:
16         subprocess.run([
17             "python", "eval_dim_ser.py",
18             "--ssl_type", ssl_type,
19             "--pooling_type", pool_type,
20             "--model_path", model_path,
21             "--store_path", store_path
22         ], check=True)
23
24         print(f"Evaluation for seed {seed} completed successfully.")
25     except subprocess.CalledProcessError as e:
26         print(f"Error occurred: {e}")
27
28 if __name__ == "__main__":
29     ssl_type = "wavlm-large"
30     pool_type = "AttentiveStatisticsPooling"
31     seed = 7
32
33     run_evaluation(seed, ssl_type, pool_type)
```

In this code, the multi-task emotion attribute model is evaluated. This code is consistent with the previous code and is designed to conveniently handle the evaluation of multi-task emotion models.

## 5. Valence Model Evaluation Code

```
1  # -*- coding: UTF-8 -*-
2  import os
3  import subprocess
4
5  def run_evaluation(seed, ssl_type, pool_type):
6      model_path = f"C:/Users/user/Desktop/MSP/model/dim_val_ser/wavLM_adamw/{seed}"
7      store_path = f"C:/Users/user/Desktop/MSP/result/dim_val_ser/wavLM_adamw/{seed}.txt"
8
9      os.makedirs(os.path.dirname(store_path), exist_ok=True)
10
11     try:
12         subprocess.run([
13             "python", "eval_val_dim_ser.py",
14             "--ssl_type", ssl_type,
15             "--pooling_type", pool_type,
16             "--model_path", model_path,
17             "--store_path", store_path
18         ], check=True)
19
20         print(f"Evaluation for seed {seed} completed successfully. Results saved to {store_path}.")
21     except subprocess.CalledProcessError as e:
22         print(f"Error occurred: {e}")
23
24     if __name__ == "__main__":
25         ssl_type = "wavlm-large"
26         pool_type = "AttentiveStatisticsPooling"
27         seed = 7
28         run_evaluation(seed, ssl_type, pool_type)
```

This code is written for evaluating the emotion value model. It also follows the same structure and logic. This code also ensures that the storage path is created before running the evaluation so that the results can be saved.

The above code shows the evaluation process for different models. But they specify different evaluation scripts and parameters to achieve model verification and result storage. Each code is logically similar, especially in how to run the evaluation and manage the path.

# Results

## 1. Arousal Model



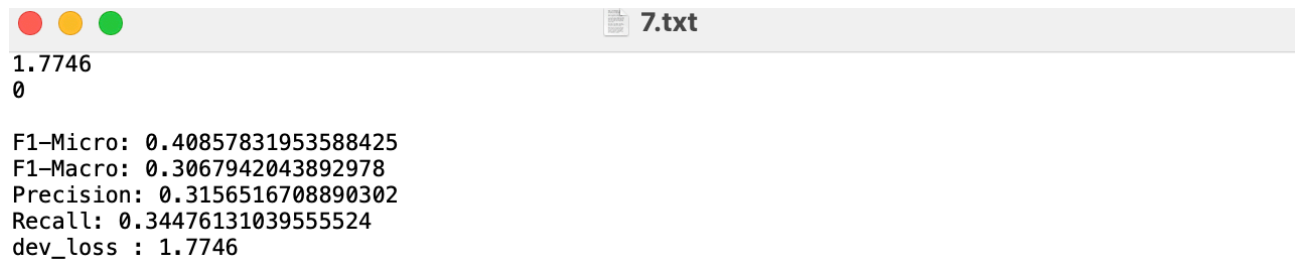
```
0.6509
dev_aro : 0.6509
```

The arousal model got the score of 0.6509 for the evaluation dev\_aro.

This score indicates the ability of the model to predict the arousal levels of the audio samples in the dataset. Arousal in the context of emotion is the intensity of the emotion being expressed. It is ranging from calm to excited. The score of 0.6509 shows the model has a reasonable understanding of the emotional intensity in the speech data. But there is

still room for improvement. Future iterations could benefit from more diverse training data or tuning of hyperparameters to enhance performance.

## 2. Categorical Emotion Recognition Model



```
1.7746
0

F1-Micro: 0.40857831953588425
F1-Macro: 0.3067942043892978
Precision: 0.3156516708890302
Recall: 0.3447613103955524
dev_loss : 1.7746
```

F1-Micro: 0.4086  
F1-Macro: 0.3068  
Precision: 0.3157  
Recall: 0.3448  
Loss: 1.7746

The performance of the categorical emotion recognition model is characterized by a relatively low F1-Macro score (0.3068). It means the model is struggle to balance precision and recall across different emotion classes. The F1-Micro score (0.4086) is slightly better while some classes may be predicted well. But others are underperforming. The precision (0.3157) and recall (0.3448) scores further illustrate this imbalance.

## 3. Dominance Model

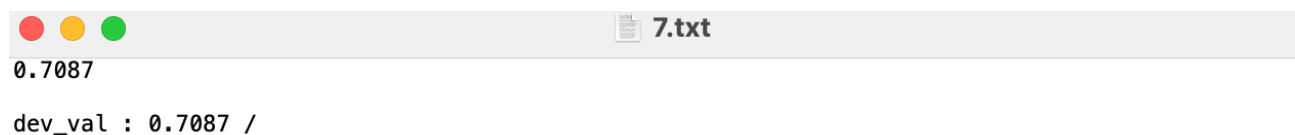


```
0.584

dev_dom : 0.584
```

The evaluation metric dev\_dom gave the dominance model a score of 0.584. The model's reasonable ability to predict the levels of dominance in the audio samples is indicated by its score of 0.584. The level of power or control someone feels in a certain emotional situation is reflected in their dominance. This score indicates that the model is effective, albeit it might not be able to fully capture subtle variations in the speakers' levels of dominance.

## 4. Valence Model



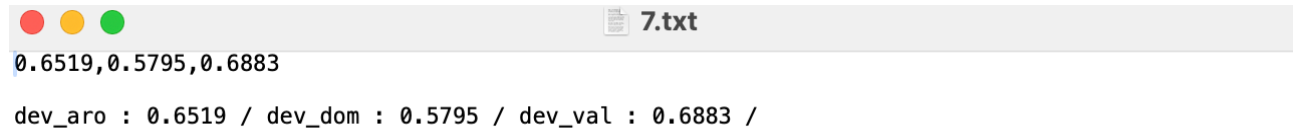
```
0.7087

dev_val : 0.7087 /
```

The evaluation metric dev\_val gave the valence model a score of 0.7087. The valence score of 0.7087 indicates that the audio samples' emotional worth was accurately predicted. It

stands for the stated feelings' positivity or negative. This indicates that the model does a reasonable job of differentiating between positive and negative emotions. The accuracy of this model could be significantly increased with more training and optimization.

## 5. Multi-task Emotion Attributes Model

A screenshot of a terminal window with a title bar containing three colored circles (red, yellow, green) and a file icon labeled '7.txt'. The terminal displays the following text:

```
0.6519,0.5795,0.6883  
dev_aro : 0.6519 / dev_dom : 0.5795 / dev_val : 0.6883 /
```

Arousal, dominance, and valence scores on the multi-task emotional characteristics model were 0.6519, 0.5795, and 0.6883, respectively.

The outcomes of the multi-task model demonstrate balanced performance for each of the three characteristics. It makes good use of information that is sharing between tasks. The dominance score (0.5795) is much lower than the arousal score (0.6519) and valence score (0.6883). Although the model does a good job of forecasting emotional intensity and positivity/negativity, it is bad in determining the power dynamic (dominance).

Overall, the evaluation of these models highlights strengths in recognizing emotional attributes, particularly valence and arousal.

## Conclusion

In this project, we systematically evaluated a variety of emotion recognition model. Including arousal, dominance, sentiment value, and categorical emotion models. We use the Odyssey 2024 dataset and obtain the performance of the models on various emotion recognition tasks.

The results shows the arousal and sentiment value models performed relatively well, with scores of 0.6509 and 0.6883 respectively. It indicates that they have strong capabilities in capturing and understanding emotional expressions. However, the dominance model scored 0.584. At the same time, the F1 score of the categorical emotion model shows that there are certain difficulties in recognizing some emotion categories, especially the recognition accuracy of low-frequency emotion categories needs to be improved.

Overall, this project provides important empirical evidence for speech emotion recognition. This project shows the complexity of different emotion types and their challenges to model design. Future research directions should include further optimizing the model structure, enhancing feature learning capabilities, and expanding the training dataset to improve the performance and robustness of the model in practical applications.

---

## Appendix

The link to the github:

<https://github.com/zjqjqjq/ELEC5305-Speech-Emotion-Recognition-Project.git>