NLP_CHRISTINE@LT-EDI: RoBERTa & DeBERTa Fine-tuning for Detecting Signs of Depression from Social Media Text

Christina Christodoulou
Institute of Informatics & Telecommunications,
National Centre for Scientific Research, "Demokritos"
Athens, Greece
ch.christodoulou@iit.demokritos.gr

Abstract

The paper outlines the approach used to detect signs of depression from English social media text for the 4^{th} Shared Task at LT-EDI@RANLP 2023. The solution involved data cleaning and pre-processing, leveraging additional data, addressing data imbalance, and fine-tuning RoBERTa-Large and DeBERTa-V3-Large transformer-based pre-trained language models. Four different model architectures were developed using different word embedding pooling methods, including a RoBERTa-Large bidirectional GRU model using GRU pooling and three DeBERTa models using CLS pooling, mean pooling, and max pooling, respectively. Although ensemble learning of DeBERTa's pooling methods was used to improve performance, the RoBERTa bidirectional GRU model received the 8^{th} place out of 31 submissions with a Macro-F1 score of 0.42.

1 Introduction

Depression is a severe mental disorder that can significantly impact an individual's thoughts, emotions, behavior, and daily routine. (of Mental Health, 2023). This condition is classified into three levels, namely mild, moderate, and severe, based on the number of symptoms present. These symptoms may include feelings of sadness, hopelessness, irritability, guilt, insomnia, fatigue, loss of appetite, and disinterest in activities. Mild depression typically manifests with 5-6 symptoms, while moderate depression involves 7-8 symptoms, and severe depression includes 9 or more symptoms, which may include hallucinations, delusions, suicidal thoughts, or even attempts (Cherney, 2018). Depression is a widespread issue, affecting approximately 280 million individuals worldwide (WHO, 2023). As social media continues to serve as a platform for individuals to express their emotions (Alyssa, 2021), the identification of symptoms of depression through automated means holds the potential for prompt intervention, psychological aid, and the avoidance of adverse outcomes. The 4^{th} Shared Task on *Detecting Signs of Depression from Social Media Text* at LT-EDI@RANLP 2023 (Sampath et al., 2023) challenged participants to develop text classification systems that can classify English social media posts into three classes, namely *not depression*, *moderate* and *severe* depression. This paper presents the system developed for this competition, with the code available on the provided GitHub link.[1]

The structure of this paper is as follows: Firstly, Section 2 presents a discussion of the previous related work followed by the presentation of the data analysis in Section 3, and an overview of the developed system in Section 4. In Section 5, an outline of the experimental setup is provided, while Section 6 presents the results and error analysis. Finally, the paper concludes with Section 7, which discusses future work.

2 Related Work

Previous work on *Detecting Signs of Depression from Social Media Text* was conducted at LT-EDI@RANLP 2022. The majority of participating teams used transformer-based language models, such as BERT (Anantharaman et al., 2022), DistilBERT, and RoBERTa (S et al., 2022), while several teams used traditional machine learning methods like Logistic Regression (Agirrezabal and Amann, 2022), Support Vector Machines, Random Forest, and XGBoost Classifiers (Sharen and Rajalakshmi, 2022). The top-ranking team, *OPI*, experimented with BERT, RoBERTa, and XLNet,

[1] https://github.com/christinacdl/Depression_Detection_Text_Classification/blob/main/Detecting_Signs_of_Depression_from_Social_Media_Text.ipynb

LT-EDI 2023 – Third Workshop on Language Technology for Equality, Diversity and Inclusion,
pages 109–116, Varna, Bulgaria, Sep 7, 2023.
https://doi.org/10.26615/978-954-452-084-7_016

trained RoBERTa-Large from scratch on depression corpora (*DepRoBERTa*), fine-tuned it and created an ensemble model resulting in attaining a 0.583 macro-F1 score (Poświata and Perełkiewicz, 2022). The *NYCU_TWD* team, which came in second place by achieving a 0.552 macro-F1 score, experimented with gradient boosting, pre-trained transformer language models, VADER, supervised contrastive learning, and ensemble learning (Wang et al., 2022).

3 Data

3.1 Provided Datasets

The task organizers provided both the training and development data, which included lengthy texts from social media posts along with their corresponding IDs and class labels. During the testing phase, the test data was also provided, but without labels. Thus, participants had to make predictions for the test texts and submit them without immediately knowing the results or the performance of their system. The class labels of the test data were released after the competition ended. The training data consisted of 7,201 texts, while the development data consisted of 3,245 texts. The test data comprised 499 texts. In the data cleaning process, 116 and 12 duplicate texts were removed from the training and development data, respectively. The test data did not contain any duplicates.

3.2 Additional Datasets

Two additional binary-class datasets were employed and combined with the train and development datasets. The first dataset was sourced from Hugging Face and contained 7,731 English posts from Reddit labeled as 0 (*not depression*) and as 1 (*depression*).[2] The second dataset was also found on Kaggle and contained 20,363 English posts from Reddit with the labels *depression* and *SuicideWatch*.[3] In this dataset, the class label *depression* was renamed as *moderate*, while the class label *SuicideWatch* was renamed as *severe*. During the data cleaning process, 81 and 8 duplicates were removed from the first and second datasets, respectively. Table 1 illustrates the class distribution of the provided train and development data as well as the class distribution of the additional data before and after data cleaning. The categorical labels were converted into the respective numerical labels denoted in brackets for training and evaluation purposes.

Class Label	Before Data Cleaning	After Data Cleaning
Provided Train Data		
not depression (0)	2,755	2,697
moderate (1)	3,678	3,544
severe (2)	768	728
Provided Development Data		
not depression (0)	848	841
moderate (1)	2,169	2,153
severe (2)	228	228
Additional Hugging Face Data		
not depression (0)	3,900	3,879
depression (1)	3,831	3,718
Additional Kaggle Data		
depression (1)	10,371	10,359
SuicideWatch (2)	9,992	9,988

Table 1: Class distribution of provided and additional data before and after data cleaning.

3.3 Data Used

The provided training and development datasets, along with additional datasets, were concatenated to form a new dataset. This was done to increase the amount of training data and improve the class distribution, particularly for the *severe* and *moderate* classes, which were essential for this task. However, only the *not depression* texts were utilized from the first additional dataset from Hugging Face. This was because there was no clarification concerning the depression level in its *depression* texts. Since there was no information regarding whether the texts were categorized as *moderate* or *severe* depression, they were not included in the new dataset. The new dataset consisted of 34,417 text entries with their respective labels. From the class distribution in Table 2, it can be demonstrated that the two classes, representing two levels of depression, constitute the majority of the dataset. This was anticipated to assist the system in detecting signs of depression. For data splitting into the train and development sets, ten-fold cross-validation with stratified sampling was implemented. This ensured that the train and development sets would have the same proportion of class values and, hence, would

[2] https://huggingface.co/datasets/hugginglearners/reddit-depression-cleaned/tree/main

[3] https://www.kaggle.com/datasets/xavrig/reddit-dataset-rdepression-and-rsuicidewatch

be equally represented. The train set consisted of 30,976 texts, and the development set consisted of 3,441 texts.

3.4 Tackling Data Imbalance

In addition to incorporating more depression data, the *Imbalanced Dataset Sampler* was used to create the train Dataloader. This tool balances the distribution of classes when sampling from an imbalanced dataset and automatically calculates the corresponding sampling weights.[4]

Final Dataset	
Class Label	**Number of Texts**
not depression (0)	7,417
moderate (1)	16,056
severe (2)	10,944
Train Set	
Class Label	**Number of Texts**
not depression (0)	6,675
moderate (1)	14,451
severe (2)	9,850
Development Set	
Class Label	**Number of Texts**
not depression (0)	742
moderate (1)	1,605
severe (2)	1,094

Table 2: Class distribution of final dataset, train and development sets used for training and evaluation.

4 System Overview

The presented system utilizes two robust models, RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020), for fine-tuning purposes. RoBERTa-Large, which boasts 355M parameters, includes 24 layers with a hidden size of 1024 and a vocabulary size of 50,265. DeBERTa-V3-Large, on the other hand, contains 304M parameters, 24 layers with a hidden size of 1024, and a vocabulary size of 128,100. Both models leverage the *sentencepiece* tokenizer to ensure optimal performance. The models that were chosen for the study were selected based on their exceptional architecture and outstanding performance on various Natural Language Processing (NLP) tasks and benchmark datasets. One model architecture utilizes all output hidden states for GRU pooling, while other model architectures utilize the last hidden state for CLS pooling, mean pooling, and max pooling. Through extensive experimentation, it was determined that keeping the first 7 encoder layers frozen during fine-tuning resulted in the best performance. The flow diagram of the presented system is depicted in Figure 1.

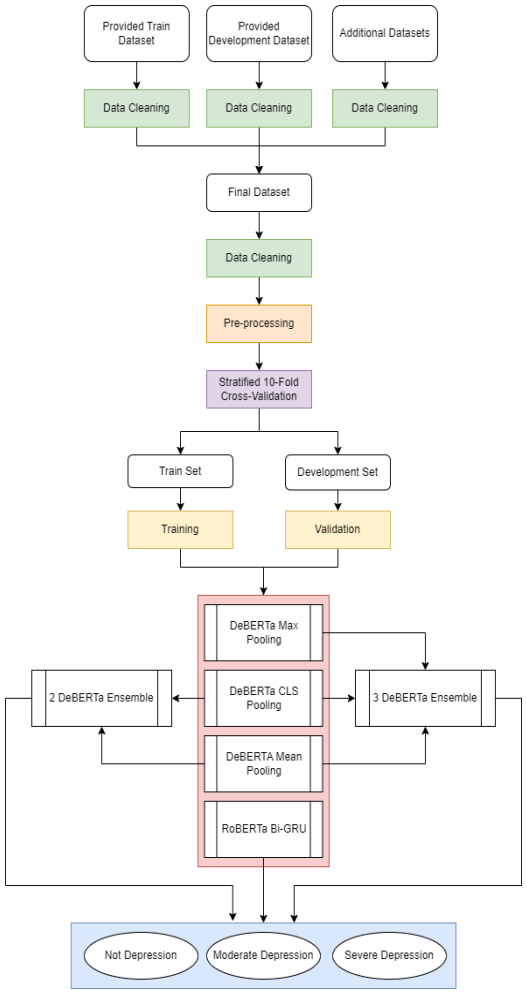

Figure 1: Flow diagram of the presented approach

4.1 GRU Pooling

The first model architecture was based on a BERT model using LSTM pooling for aspect-based sentiment analysis (Song et al., 2020). Unlike this BERT model, the model developed for this system is a RoBERTa-Large Bidirectional GRU network (***RoBERTa Bi-GRU***) utilizing GRU pooling. It is bidirectional, meaning that it can process the input and retain information from both directions. It takes all hidden states of RoBERTa ([initial embeddings + total number of layers, batch size, max sequence length, hidden size]) and passes them through a GRU network, which is used to connect all the [CLS] token representations. The output representation from the last GRU cell, which has

[4] https://github.com/ufoym/imbalanced-dataset-sampler

the size of [batch size, total number of layers, max sequence length * 2], is passed into a dropout layer. In the end, a linear layer, which has dimensions equal to the size of the maximum sequence length multiplied by two and the number of classes [max sequence length * 2, number of classes], is applied to the output from the dropout layer.

4.2 CLS Pooling

The second model architecture (***DeBERTa CLS Pooling***) implements the most common pooling method - the CLS pooling. During classification tasks, a special [CLS] token is added at the first position of each sequence to capture the entire context information of a sequence. The [CLS] token embeddings are aggregated in the pooling layer and are used as sentence embeddings. These embeddings pass through a dropout layer and finally, a linear layer that classifies the texts into three classes.

4.3 Mean Pooling

The third model architecture called (***DeBERTa Mean Pooling***), involves averaging all of the contextualized token embeddings from the last hidden state. First, the attention mask is expanded from [batch size, maximum sequence length] to [batch size, maximum sequence length, hidden size]. Then, the token embeddings are summed along the maximum sequence length axis to end up with a size of [batch size, hidden size]. The attention mask is also summed along the maximum sequence length axis so that padding tokens ([PAD]) are ignored. The mean embeddings, which are the average of the summed token embeddings and the summed attention mask, pass through a dropout layer and finally, a linear layer, which classifies the texts into the three classes.

4.4 Max Pooling

The fourth model architecture (***DeBERTa Max Pooling***) uses the maximum pooling method by taking the maximum value of the token embeddings from the last hidden state at each time step. The attention mask was expanded from [batch size, maximum sequence length] to [batch size, maximum sequence length, hidden size]. Then, the padding tokens were set to a large negative value (-1e9). The maximum token embeddings produce sentence embeddings that pass through a dropout layer and, finally, through the classifier linear layer.

4.5 Majority Vote Ensemble Learning

In this particular study, two different ensemble learning methods were utilized to predict the class labels for each given text. The ultimate label that was submitted for each text was determined by selecting the most commonly predicted label from the individual classifiers. The first ensemble method (***3 DeBERTa Ensemble***) combined predictions from all three pooling DeBERTa classifiers, while the second ensemble method (***2 DeBERTa Ensemble***) only utilized predictions from the CLS and mean pooling classifiers, since they achieved higher Macro-F1 scores compared to the max pooling classifier. This approach was taken to ensure the most accurate and reliable results possible.

5 Experimental Setup

5.1 Environment Setup

The presented approach was implemented in Python using Google Colaboratory (Colab) Pro notebook. Experiments were conducted with *Pytorch* library and NVIDIA A100-SXM4-40GB GPU.

5.2 Pre-processing Steps

Pre-processing steps were applied to the training, development, and test sets of text using a function that included regular expressions and other functions. The function removed URLs, usernames, and retweets. Emojis were converted to their textual representations (Taehoon et al., 2022).[5] The *&* and *&* were replaced with *and*. The ASCII encoding apostrophe was replaced with the UTF-8 encoding apostrophe. Consecutive non-ASCII characters were replaced with whitespace, and all extra whitespace was removed. Contracted words were unpacked, such as *isn't* being converted to *is not*. The *Ekphrasis* library was used to segment hashtags, correct spelling, elongate words, tokenize, and lowercase all words (Baziotis et al., 2017).[6] All punctuation marks were maintained as they contribute to the context of the text.

5.3 Hyperparameter Tuning

The pre-trained models required PyTorch tensors as input, including input IDs and attention masks. Sequences were padded to the fixed maximum sequence length of RoBERTa and DeBERTa (512).

[5] https://pypi.org/project/emoji/

[6] https://github.com/cbaziotis/ekphrasis

Dropout and early stopping patience were used to prevent overfitting, and gradient accumulation was employed to virtually increase batch size during training. The models utilized Cross-Entropy Loss for multi-class classification, with the *AdamW* optimizer and consistent hyperparameters across all architectures shown in Table 3. Identical hyperparameters were employed across all models to ensure easy comparison of models.

Hyperparameters	
Number of Classes	3
Number of Epochs	10
Sequence Length	512
Train Batch Size	10
Development Batch Size	16
Learning Rate	2e-6
Weight Decay	0.1
Warm-up Steps	0
AdamW Epsilon	1e-8
AdamW Betas	0.9, 0.999
Dropout	0.2
Gradient Clipping	1.0
Gradient Accumulation	2
Early Stopping Patience	5
Random Seed	42

Table 3: Hyperparameters of Models.

5.4 Metrics

The system's efficiency and final ranking were primarily evaluated based on the Macro-F1 score of the test set predictions. Additionally, submissions were evaluated by the organizers based on accuracy, Macro-Recall, Macro-Precision, Weighted-F1, Weighted-Recall, and Weighted-Precision scores. The evaluation also included the Macro-F1 score and Confusion Matrix for each class.

6 Results

6.1 Development Set

Table 4 shows that the DeBERTa Mean Pooling model achieved the highest Macro-F1 score among individual models (0.77), while the DeBERTa Max Pooling model scored the lowest (0.74). Notably, the RoBERTa Bi-GRU and the DeBERTa CLS Pooling both scored 0.76. Looking at the Macro-F1 score of each class, RoBERTa Bi-GRU is more successful in identifying the *not depression* class (0.82), while DeBERTa Mean Pooling is more successful in identifying the *moderate* class (0.74). All three DeBERTa pooling methods are better at detecting the *severe* class than the RoBERTa Bi-GRU, with a slightly higher Macro-F1 score (0.76). The ensemble including all three DeBERTa models achieves a slightly higher general Macro-F1 score as well as a little higher score in detecting the *severe* class.

Development Set	
Metric	**RoBERTa Bi-GRU**
Macro-F1	0.76
Classes	**Macro-F1**
not depression	0.82
moderate	0.72
severe	0.75
Metric	**DeBERTa CLS Pooling**
Macro-F1	0.76
Classes	**Macro-F1**
not depression	0.81
moderate	0.73
severe	0.76
Metric	**DeBERTa Mean Pooling**
Macro-F1	0.77
Classes	**Macro-F1**
not depression	0.81
moderate	0.74
severe	0.76
Metric	**DeBERTa Max Pooling**
Macro-F1	0.74
Classes	**Macro-F1**
not depression	0.80
moderate	0.68
severe	0.76
Metric	**3 DeBERTa Ensemble**
Macro-F1	0.77
Classes	**Macro-F1**
not depression	0.81
moderate	0.73
severe	0.77
Metric	**2 DeBERTa Ensemble**
Macro-F1	0.76
Classes	**Macro-F1**
not depression	0.81
moderate	0.73
severe	0.76

Table 4: Results of developed models on the development set.

6.2 Test Set

Table 5 shows that the RoBERTa Bi-GRU model outperformed the DeBERTa ensemble models in all metrics, securing 8^{th} place with a macro-F1 score of 0.42. It achieved a higher Macro-F1 score in the *not depression* class (0.11) and a slightly higher score in the *severe* class (0.46) compared to the ensemble models. Both models performed equally well in detecting *moderate* depression with a Macro-F1 score of 0.69. The ensemble models had low Macro-F1 scores across all metrics, particularly in detecting *not depression* (0.05).

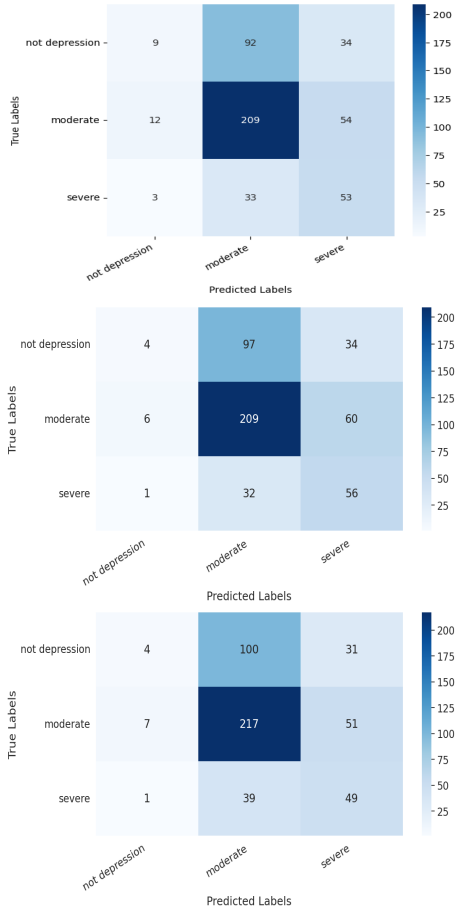

Figure 2: Test Set Confusion Matrices of RoBERTa BI-GRU, 3 DeBERTa Ensemble, 2 DeBERTa Ensemble

Test Set	
Metric	**RoBERTa Bi-GRU**
Macro-F1	0.42
Macro-Recall	0.474
Macro-Precision	0.459
Weighted-F1	0.491
Weighted-Recall	0.543
Weighted-Precision	0.513
Accuracy	0.543
Classes	**Macro-F1**
not depression	0.11
moderate	0.69
severe	0.46
Metric	**3 DeBERTa Ensemble**
Macro-F1	0.396
Macro-Recall	0.456
Macro-Precision	0.439
Weighted-F1	0.473
Weighted-Recall	0.541
Weighted-Precision	0.493
Accuracy	0.541
Classes	**Macro-F1**
not depression	0.05
moderate	0.69
severe	0.45
Metric	**2 DeBERTa Ensemble**
Macro-F1	0.396
Macro-Recall	0.456
Macro-Precision	0.439
Weighted-F1	0.473
Weighted-Recall	0.541
Weighted-Precision	0.493
Accuracy	0.541
Classes	**Macro-F1**
not depression	0.05
moderate	0.69
severe	0.45

Table 5: Results of submitted models on test set.

6.3 Error Analysis

The confusion matrices were created after the release of the test set labels so that the errors and strengths of the submitted models would be revealed. Therefore, each confusion matrix represents the performance of the RoBERTa Bi-GRU, the 3 DeBERTa Ensemble, and the 2 DeBERTa Ensemble on the test set, respectively. Considering all the confusion matrices from Figure 2, it is evident that all models tend to detect signs of depression in text with greater confidence and success, while they are not as capable of distinguishing non-depression from depression texts. They successfully detect many texts that show *moderate* signs of depression, while there seems to be confusion when it comes to identifying between the *moderate* and *severe* classes, as texts that belong to the *severe* class were assigned to the *moderate* class. A notable number of texts belonging to the *moder-*

ate class appear to be identified as *not depression*, while texts that should be labeled as *not depression* were labeled as *moderate* depression. This illustrates the difficulty of the models to distinguish non-depressive posts from depressive posts as well. The reason for the failure of the models in detecting non-depressive posts lies in the fact that the training data contained a significantly lower number of texts categorized as *not depression* (6,675) compared to those classified as *severe* (9,850) and *moderate* (14,451) classes. The training algorithm placed greater emphasis on boosting the depression classes, which further skewed the models' ability to accurately detect non-depressive posts.

7 Conclusion and Future Work

The LT-EDI@RANLP 2023 Shared Task 4 entailed the development of a system aimed at addressing data imbalance, cleaning, pre-processing, and fine-tuning pre-trained language models to accurately identify depression in English social media posts. Two pre-trained language models, RoBERTa-Large and DeBERTa-V3-Large, were employed and fine-tuned for this purpose. Among the four pooling methods tested, the RoBERTa-Large Bidirectional GRU model demonstrated the best performance. This model effectively identified posts exhibiting signs of depression, particularly at moderate levels. However, it struggled with detecting non-depressive posts and may occasionally mistake *severe* depression for *moderate* depression.

To further enhance the models' performance, future efforts should focus on incorporating more non-depressive texts into the training data and experimenting with the multi-layer structure of pre-trained Transformer models as well as various hyperparameters. Overall, this system has the potential to serve as a valuable tool for early detection of depression, enabling prompt intervention and support for individuals who may be experiencing mental health problems.

References

Manex Agirrezabal and Janek Amann. 2022. KUCST@LT-EDI-ACL2022: Detecting signs of depression from social media text. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 245–250, Dublin, Ireland. Association for Computational Linguistics.

Alyssa. 2021. Why depression is so common: Banyan mental health.

Karun Anantharaman, Angel S, Rajalakshmi Sivanaiah, Saritha Madhavan, and Sakaya Milton Rajendram. 2022. SSN_MLRG1@LT-EDI-ACL2022: Multi-class classification using BERT models for detecting depression signs from social media text. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 296–300, Dublin, Ireland. Association for Computational Linguistics.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Kristeen Cherney. 2018. Mild, moderate, or severe depression? how to tell the difference.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

National Institute of Mental Health. 2023. Depression.

Rafał Poświata and Michał Perełkiewicz. 2022. OPI@LT-EDI-ACL2022: Detecting signs of depression from social media text using RoBERTa pre-trained language models. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 276–282, Dublin, Ireland. Association for Computational Linguistics.

Sivamanikandan S, Santhosh V, Sanjaykumar N, Jerin Mahibha C, and Thenmozhi Durairaj. 2022. scubeMSEC@LT-EDI-ACL2022: Detection of depression using transformer models. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 212–217, Dublin, Ireland. Association for Computational Linguistics.

Kayalvizhi Sampath, Thenmozhi Durairaj, Bharathi Raja Chakravarthi, Jerin Mahibha C, Kogilavani Shanmugavadivel, and Pratik Anil Rahood. 2023. Overview of the second shared task on detecting signs of depression from social media text. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, Varna, Bulgaria. Recent Advances in Natural Language Processing.

Herbert Sharen and Ratnavel Rajalakshmi. 2022. DLRG@LT-EDI-ACL2022:detecting signs of depression from social media using XGBoost method. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 346–

349, Dublin, Ireland. Association for Computational Linguistics.

Youwei Song, Jiahai Wang, Zhiwei Liang, Zhiyue Liu, and Tao Jiang. 2020. Utilizing BERT intermediate layers for aspect based sentiment analysis and natural language inference. *CoRR*, abs/2002.04815.

Kim Taehoon, Tahir Kevin, Wurster, and Jalilov. 2022. Emoji.

Wei-Yao Wang, Yu-Chien Tang, Wei-Wei Du, and Wen-Chih Peng. 2022. NYCU_TWD@LT-EDI-ACL2022: Ensemble models with VADER and contrastive learning for detecting signs of depression from social media. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 136–139, Dublin, Ireland. Association for Computational Linguistics.

WHO. 2023. Depressive disorder (depression).