# Final Report
# Dreaddit: A Reddit Dataset for Stress Analysis in Social Media

Ciona Sha-ked - 336368220, Ilan Golik - 321238297

August, 2022

# 1 Anchor Paper Review

## 1.1 Introduction

Much of the literature on stress centers around the physiological aspect. Because stress is widely expressed across social media platforms, this poses an opportunity to study the linguistic nature of stress. The research presented in this article focuses on stress classification in social media and demonstrates the relevance of studying stress from a lexical standpoint. The research presented in this article results in a dataset that is a concatenation of NLP based features and features contributed by psychologists. It attempts to create a model that is comparable in performance to Bert, but simultaneously also interpretable.

## 1.2 Dreaddit Dataset

The dataset used in the study is comprised of text extracted from Reddit posts (dubbed “subreddits”) that span various stress-related domains. The advantage of the Reddit dataset (”Dreaddit dataset”) over other subject material offered on social media is the length of the posts which help provide deeper insight.

Thousands of segments from Reddit posts (with equal representation across domains) were annotated by Mechanical Turk Workers. They labeled each segment as either “stress” or “not stress”. The majority vote became the label and the percentage of annotators that agreed with the label was recorded. There was consensus among the annotators for only 39% of the data. This indicates variety in how stress is expressed.

Depending on the domain, one can observe a difference in the variety of stress expression from the text. The authors conducted an analysis of the percentage of words that each domain contains from a specialized LIWC word list. LIWC (Linguistic Inquiry and Word Count) is a lexicon tool that assesses text based on topic prevalence. They also analyzed the percentage of words from the LIWC list found in the domain. Their conclusion was that when it comes to the negative emotion word list, domains that are interpersonal have a larger variety of expression. More words from the negative emotion word list are covered compared to the non-interpersonal domains.

Another method of measuring lexical diversity by domain is through calculating Yules I. The lower left graph shows that there is lower lexical diversity from the domains related to mental illness. When the Yule I was used to compare the domains for words only found in the LIWC negative emotion word list, the anxiety domain showed the lowest lexical diversity (in the lower right graph).

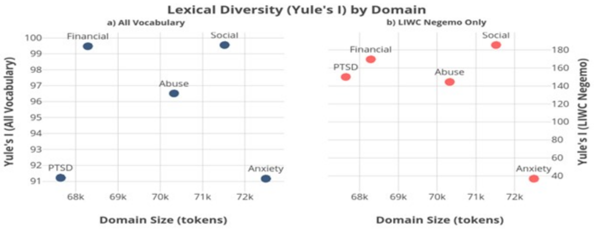


Figure 1: Lexical Diversity by Domain

An additional way of exploring lexical diversity is measuring it against the percentage of annotators that agreed on a label. They found that there was an inverse correlation between the two. This intuitively makes sense, the more lexical diversity, the harder it is to agree on connotative meanings.

They furthermore assessed the data by breaking it down by label. They found that stressed labeled data contained more first-person pronouns compared to non-stressed. Also, it was observed that stress-labeled data is more syntactically complex (containing more clauses, conjunctions, etc...), yet simultaneously easier to read according to the F-K readability index.

### 1.3 Methods

In the Dreaddit study authors used supervised models for classification. They had 2,838 training samples and 715 test samples. They included four types of features in the model:

1. text from the posts
2. lexical features (which includes 93 LIWC features)
3. syntactic features – they used part-of speech uni-grams and bi-grams as well as the F-K readability index previously mentioned
4. social media features – this includes the post's UTC timestamp as well as statistics regarding the posts; upvotes and downvotes

The authors experimented with different subsets of these features as well as with subsets based on a feature's Pearson correlation with the label. The researchers stratified the data according to the level of annotation agreement.

They also experimented with different input representations that we attempted to replicate. These representations included Google News pre-trained Word2Vec embeddings, Word2Vec and BERT embeddings on their unlabeled data, and bag-of-n-grams. Some versions of the experiment then used random initialization for training embeddings, and for others, they initialized with Word2Vec embeddings that were domain specific. The non-neural experiments used SVM, logistic regression, Naïve Bayes, decision trees, and Perceptron. In addition, the authors used neural network-based models like CNN and a bidirectional two-layered GRNN. Unfortunately, the neural network-based model resulted poorly because of the lack of data (”Dreaddit dataset” is relatively small).

The best result for the supervised experiments was obtained when using logistic regression with domain-specific Word2Vec embeddings. This was the case when it performed on the best subset of features. They found that the best set of features for the dataset were the features with high correlation with the training labels ($\rho \geq 0.4$ Pearson correlation) and "high agreement data" which indicates at least 80% of the annotators agreed with the label. Features with high correlation to the training labels tended to be LIWC features (more so than social and syntactic features for example). One of these main LIWC features were first-person pronoun features.

The best supervised model came close in performance to the Bert model results with F1 scores of 0.798 and 0.806 respectively. In the error analysis of the two models, they found that both models tend to over-classify stress (with a difference in their specific predictions). This was oftentimes the case when there was low-annotation agreement.

## 2 Anchor Paper Implementation

### 2.1 External code repository

1. The code presented in the next ”kaggle” thread was used in order to re-implement paper code: https://www.kaggle.com/code/sohommajumder21/bert-tokenizer-with-9-models-nlp-stress-analysis.
2. sklearn libraries
3. BERT transformers

## 2.2 Coding effort

### 2.2.1 Technical challenges

1. We couldn't find correlation that are above $\rho \geq 0.4$, all our selected features are around $0.2 \leq \rho < 0.4$

2. Our feature-based model has slightly worse F1 than that which was presented in the paper. The reason for this is that we have less domain data (3.5K labeled vs. 190K that researches have in total).

3. Poor computational resource

### 2.2.2 Models implementation

We tried five different approaches in attempt to replicate the results in the article. In all of them we got good comparable results. For each approach, we first cleaned up the dataset. For example, we cleaned up the column feature that contains the text from the Reddit and performed stop word removal, stemming, lemmatization etc. We also clean it by removing features that aren't highly correlated to the training label. After that we performed embeddings based on the type of the model we were dealt with.

Word2Vec and 'word2vec-google-news-300' corpus:

1. Word2Vec - we generated embeddings based on the fitted w2v model which was pretrained on 'word2vec-google-news-300' corpus.

2. Results:

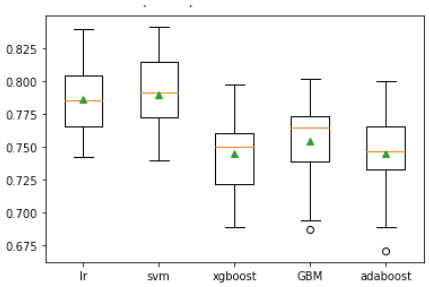
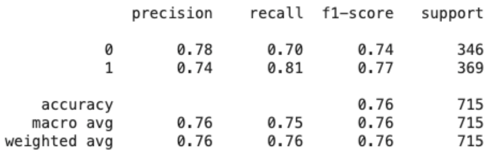


Figure 2: model based on Word2Vector generated text features

3. Here the LoR based model is a winner model as well



```
              precision    recall  f1-score   support

           0       0.78      0.70      0.74       346
           1       0.74      0.81      0.77       369

    accuracy                           0.76       715
   macro avg       0.76      0.75      0.76       715
weighted avg       0.76      0.76      0.76       715
```

Figure 3: Word2Vector classification report

Word2Vec and domain data:

1. Word2Vec - we generated embeddings based on the fitted w2v model which was pretrained on domain data.
2. Results:

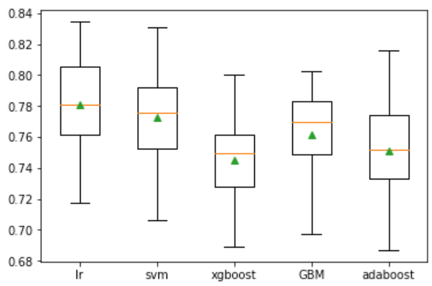
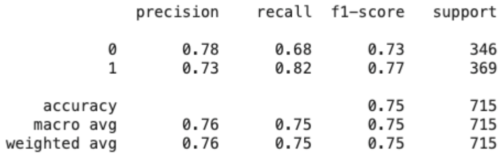


Figure 4: model based on Word2Vector generated text features

3. It is easy to see that the LoR based model has very good performance comparing to other, thus we can take it to calculate the report.



```
              precision    recall  f1-score   support

           0       0.78      0.68      0.73       346
           1       0.73      0.82      0.77       369

    accuracy                           0.75       715
   macro avg       0.76      0.75      0.75       715
weighted avg       0.76      0.75      0.75       715
```

Figure 5: Word2Vector classification report

BERT text embeddings:

1. BERT - we generated embeddings based on the fitted BERT tokenized text data.
2. Results:

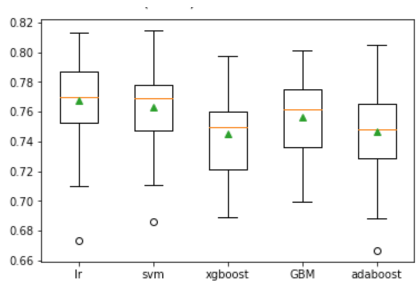
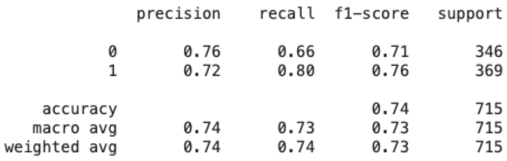


Figure 6: model based on BERT generated text features

3. It is easy to see that the LoR based model has very good performance comparing to other, thus we can take it to calculate the report.



```
              precision    recall  f1-score   support

           0       0.76      0.66      0.71       346
           1       0.72      0.80      0.76       369

    accuracy                           0.74       715
   macro avg       0.74      0.73      0.73       715
weighted avg       0.74      0.74      0.73       715
```

Figure 7: BERT classification report

TF-IDF:

1. TF-IDF - we generated embeddings based on the fitted TfidfVectorizer model
2. Results:

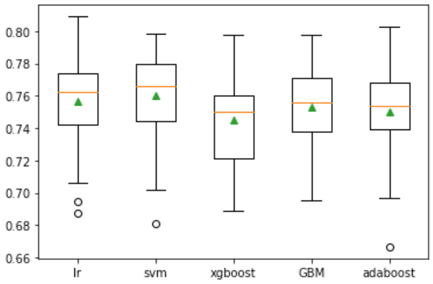
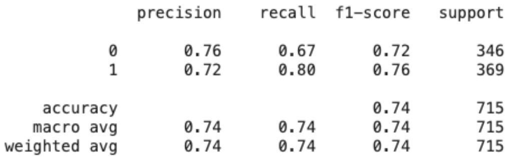


Figure 8: model based on TF-IDF generated features

3. We could see here that the LoR based model has very good performance comparing to other, and therefore we can take it again in order to calculate the report.

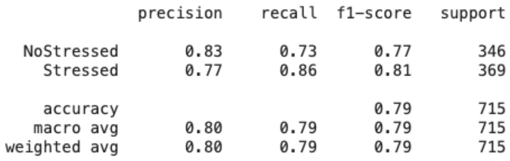


```
              precision    recall  f1-score   support

           0       0.76      0.67      0.72       346
           1       0.72      0.80      0.76       369

    accuracy                           0.74       715
   macro avg       0.74      0.74      0.74       715
weighted avg       0.74      0.74      0.74       715
```

Figure 9: TF-IDF classification report

BERT text classifier:

1. Here we used the BERT model to make a tokenizing and to perform the classification as well. It is easy to see that BERT resulted in best F1 and precision.



```
              precision    recall  f1-score   support

  NoStressed       0.83      0.73      0.77       346
    Stressed       0.77      0.86      0.81       369

    accuracy                           0.79       715
   macro avg       0.80      0.79      0.79       715
weighted avg       0.80      0.79      0.79       715
```

Figure 10: BERT classification report

Models F1 plot:

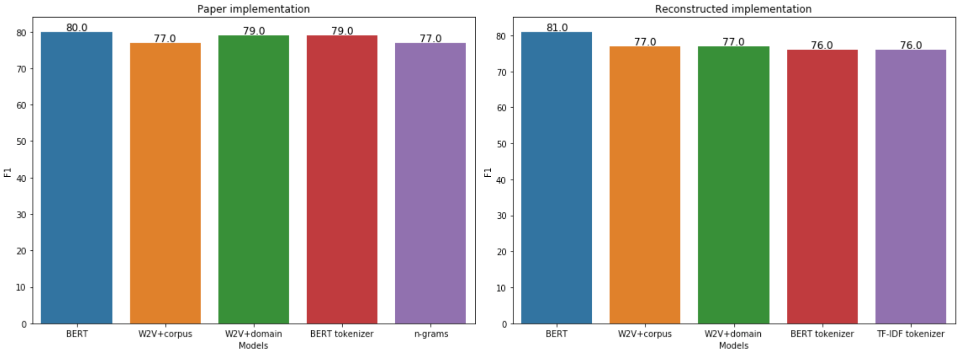


Figure 11: Models F1 report

Links to models:

- BERT model - link to google collab
- W2V domain pretrained model - link to google collab
- W2V news corpus pretrained model - link to google collab
- BERT embeddings pretrained model - link to google collab
- TF-IDF model - link to google collab

### 2.3 Anchor papers final words

It appears to us that the mission of the authors was threefold. First, they wanted to find the dataset with the best subset of features. Secondly, they wanted to find the right model to analyze the data properly and confirm that the dataset chosen was a good one. And lastly, they wanted the model to have interpretable results. More important than obtaining good accuracy, the researchers were looking for models that would yield results that are more interpretable and to be able to point to a specific subset of features that most help the classification accuracy. They succeeded in finding such models and these models gave nearly-identical F1 scores to Bert.

As seen in figure 10, our results are quite similar to the ones obtained in the study. We took the models with the highest F1 scores and managed to recreate very similar performances. The results weren't identical and this makes sense given that we weren't given access to their full dataset (we were restricted to labeled data). In both the paper implementation as well as in our reconstructed implementation, Bert yields the best results in terms of F1 score. However the remaining models also yielded reasonable accuracies.

## 3 Innovative part

### 3.1 Overview

Our original proposal was to explore the impact of the topic generation to better understand the underlying semantics. We separated stressed and non-stressed due to the assumption that there are different hidden state behaviors in the semantics. We used a popular LDA generator because it's known to be a powerful tool for large-sized texts. LDA helped in the k-fold training stage by performing very well on the validation but didn't perform well on the test predictions. This was observed when we used various models (random forest, logistic regression, SVM, MLP) as well as different hyperparameters. This led us to the idea that we needed a more data-centric approach to improve the performance of our model.

In the results and discussion section of the paper, it is noted that "the examples misclassified by both models are often, though not always, ones with low annotator agreement (with the average percent agreement for misclassified examples being 0.55 for BERT and 0.61 for logistic regression)." We deducted from this that to improve upon the results of the model, it's worth focusing on good quality data.

We created an improved training dataset where we only took data points that reached a high level of annotator agreement. The threshold that we chose was 80%, meaning at least $\frac{4}{5}$ of the annotators agreed on a label. However, by filtering for high level agreement data, we lose out on quantity of the data points. To address this issue, we performed data augmentation using SynonymAug and created new texts by using similar words to the original. We wanted to keep a close balance between the two types of labeled texts, so we augmented the stressed-text and non-stressed-text in the correct proportions. The augmented data points are unlabeled.

With a dataset of only partially labeled data, we used a semi-supervised learning approach after seeing examples of this in other studies. In general, it can be useful when labeled data is lacking in number to include unlabeled data. This is because unlabeled data is usually more abundant and requires no man-effort in the labeling process.

In one such study "Handling partially labeled network data: a semi-supervised approach using stacked sparse autoencoder", the authors used semi-supervised learning in the effort of classifying network traffic. They wanted to explore the relationship between their model's performance and the ratio of labeled-to-unlabeled data.

## 3.2 Relevant Studies

It might be assumed that if the number of unlabeled data is greater in number than the labeled data, it would adversely affect the performance of the model. However, they found the opposite to be the case. They calculated the ratio by dividing the number of unlabeled data by the number of labeled data. In the graph below you can see the accuracy as a function of the ratio (Ru).

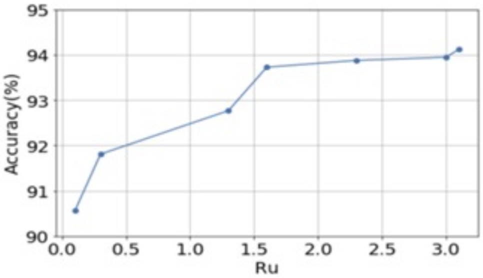


Figure 12: Performance of models with different unlabeled ratios

The graph illustrates that the accuracy improves as the ratio increases. It is explained in the article that this is the case perhaps because it offers more "informative characteristics" that helps with the deep learning in the pre-training stage which in turn improves the performance of classification on the test set. We used a ratio of 1-1 because our unlabeled data is augmented and isn't "natural" data and therefore didn't want it to outnumber the amount of data in the training set. However, we felt assured in making it match in size to the training set.

A similar technique was used in the article "Semi Supervised Audio Classification with Partially Labeled Data" where they started with partially labeled data and afterwards improved the quality of labeling based on a semi-supervised approach. The main idea in this paper was to classify the type of instruments in audio recordings. It was difficult to obtain labeling because of the length of the clips and because of the number of instruments playing simultaneously. Therefore they sought a way to obtain labeling without requiring human-effort in the endeavor. It can be learned from their study that using semi-supervised learning was a successful approach.

Based on these studies, we felt confident in using a similar technique and improve the quality of labeling of our data.

## 3.3 Implementation

To improve upon the results from the anchor paper, we decided to focus on three things:

- Topic modeling
- Data Augmentation
- Semi-supervised learning

The three together yielded good results. To obtain the topic modeling we used LDA because it is known to be helpful with longer texts. The number of topics was selected based on the coherence measure.

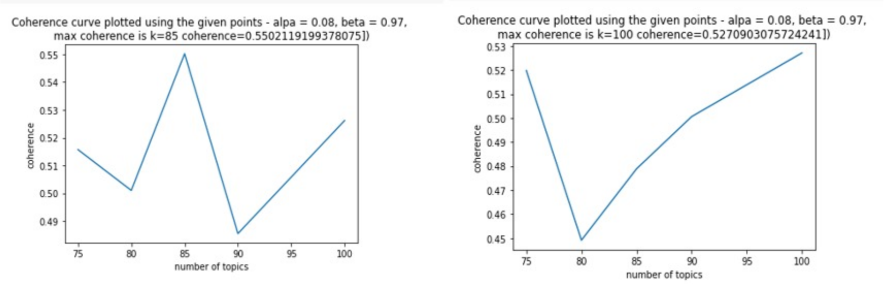


Figure 13: The number of topics as the function of coherence

In k-fold cross validation we observe better performance compared to that in the reconstructed anchor paper.

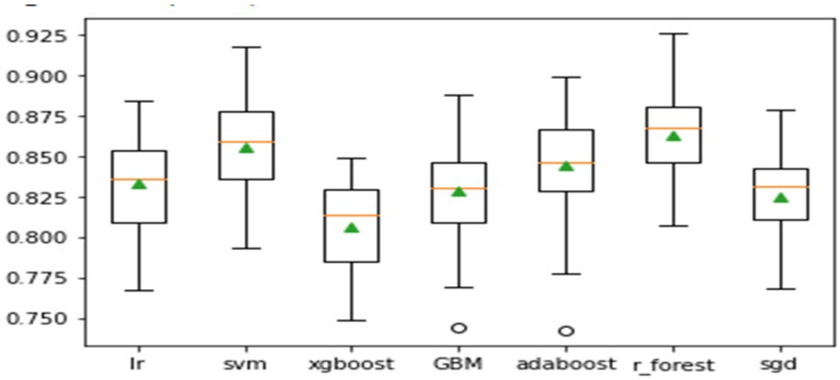
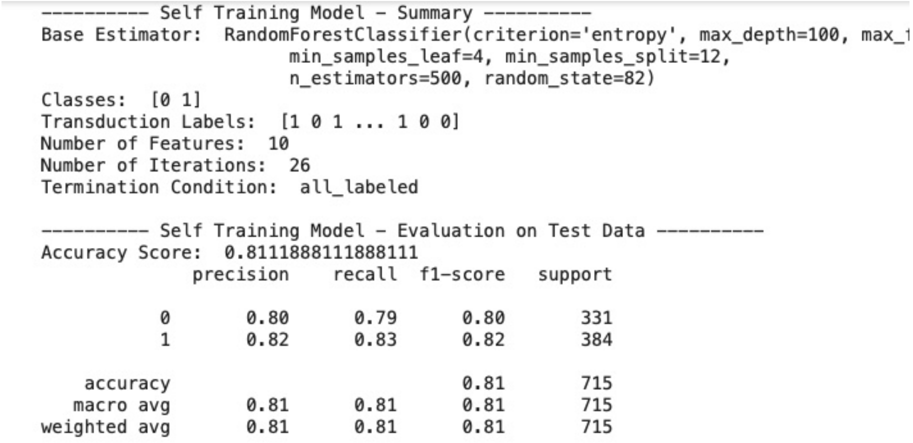


Figure 14: K-Fold training results

It is easy to observe that we are improving validation results over those presented in the anchor paper. However, the topic generation wasn't helpful in the test stage. We decided to augment our data and preserve the ratio between stressed and non-stressed instances (data augmentation is known to be a powerful technique when improving the training stage). Topic generation and data augmentation didn't yield improved test results. This led us to the understanding that we needed an improvement in the quality of the labeling in our dataset (on both training and in test). Therefore, we decided to use semi-supervised learning in order to improve the quality of the labeling.

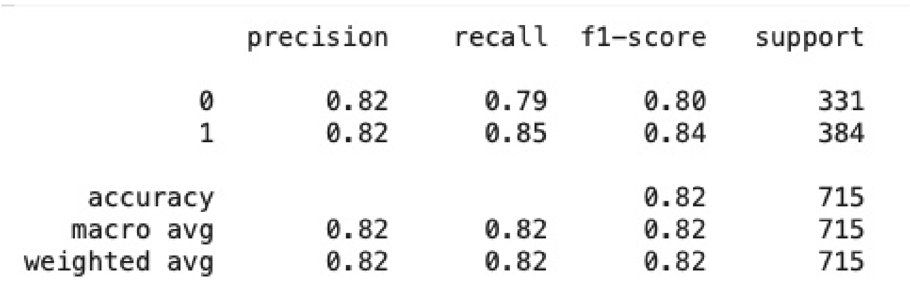
We generated labels for the augmented data through the inner semi—supervised model and then we validated that the performance was good by running a new test process (random forest) and seeing if it behaves as expected.



```
---------- Self Training Model - Summary ----------
Base Estimator:  RandomForestClassifier(criterion='entropy', max_depth=100, max_f
                       min_samples_leaf=4, min_samples_split=12,
                       n_estimators=500, random_state=82)
Classes:  [0 1]
Transduction Labels:  [1 0 1 ... 1 0 0]
Number of Features:  10
Number of Iterations:  26
Termination Condition:  all_labeled

---------- Self Training Model - Evaluation on Test Data ----------
Accuracy Score:  0.8111888111888111
              precision    recall  f1-score   support

           0       0.80      0.79      0.80       331
           1       0.82      0.83      0.82       384

    accuracy                           0.81       715
   macro avg       0.81      0.81      0.81       715
weighted avg       0.81      0.81      0.81       715
```

Figure 15: Semi-supervised training results

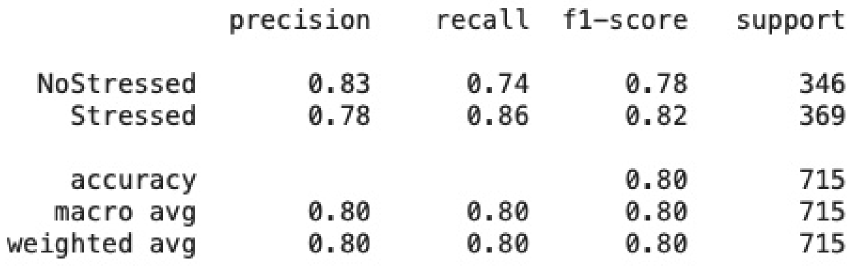
So, we can see that this approach indeed yields the expected improvements. In the end we ran the training process with the LoR mode, and observed even better results.



```
              precision    recall  f1-score   support

           0       0.82      0.79      0.80       331
           1       0.82      0.85      0.84       384

    accuracy                           0.82       715
   macro avg       0.82      0.82      0.82       715
weighted avg       0.82      0.82      0.82       715
```

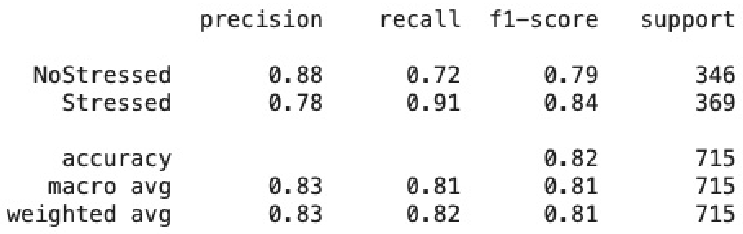
Figure 16: Training on new labels

In addition we have a BERT based models that was trained using the augmented data and here we can observe the improvement as well.



```
              precision    recall  f1-score   support

  NoStressed       0.83      0.74      0.78       346
    Stressed       0.78      0.86      0.82       369

    accuracy                           0.80       715
   macro avg       0.80      0.80      0.80       715
weighted avg       0.80      0.80      0.80       715
```

Figure 17: The BERT bert-base-uncased model



```
              precision    recall  f1-score   support

  NoStressed       0.88      0.72      0.79       346
    Stressed       0.78      0.91      0.84       369

    accuracy                           0.82       715
   macro avg       0.83      0.81      0.81       715
weighted avg       0.83      0.82      0.81       715
```

Figure 18: The BERT mental-bert-base-uncased model

### 3.4 Link to git

link git

### 3.5 Summary

We have two models for the improvement attempt. Our main goal was to preserve the interpretability of the model. In the anchor section we were able to show similar results to the ones shown in the article. In the innovative section we succeeded to improve the quality of the labeling by augmenting the data and performing supervised learning.

Further work would be to improve upon the semi supervised model and try to incorporate the T-Bert approach