UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Kaire Koljal

Predicting Depression Symptoms Based on Reddit Posts

Master's Thesis (30 ECTS)

Supervisor(s): Kairit Sirts, PhD

Tartu 2022

Predicting Depression Symptoms Based on Reddit Posts

Abstract:
Using social media posts to predict mental health problems has become a popular topic in Natural Language Processing (NLP). Machine learning has been used for detecting a diagnosis or single symptoms associated with depression. As the clinical picture of depression can differ for people, it is better to detect symptoms instead of diagnosis from the social media posts. In this work, depression symptoms are predicted based on posts from Reddit page r/depression using NLP methods and multi-label classification. This work focuses on evaluating the quality of the annotations and analysing if such data can be used to train a predictive model. Each post is annotated by three annotators and the labels are aggregated in three ways to create three datasets that are used to train Transformers models. The results of this work reveal that on a small dataset with a lower annotation agreement, a majority vote over annotations gives the most reliable dataset and results. RoBERTa model shows the best learning and generalization ability in this work.

Keywords:
Multi-label classification, Transformers, symptom prediction, depression, social media

CERCS: P176 Artificial intelligence

Depressioonisümptomite tuvastamine Redditi postitustest

Lühikokkuvõte:

Sotsiaalmeedia postituste kasutamine vaimse tervisega seotud probleemide tuvastamiseks on muutunud populaarseks teemaks loomuliku keele töötluse valdkonnas. Masinõpet on kasutatud ka diagnoosi või üksikute depressiooni sümptomite tuvastamisel. Kuna depressiooniga inimestel võib olla erinev kliiniline pilt, siis on mõistlikum diagnoosi asemel sotsiaalmeedia postitustest tuvastada sümptomeid. Selles töös ennustatakse depressiooni sümptomeid Redditi r/depression lehelt kogutud postituste baasil kasutades NLP meetodeid ja mitme märgendi klassifiaktsiooni. Töö raames uuritakse märgenduste kvaliteeti ja hinnatakse, kas sellise kvaliteediga andmestikku saab kasutada ennustava mudeli treenimiseks. Igat postitust märgendas kolm inimest ning märgendusi ühestatakse kolmel moel, et luua andmestikud, mida kasutatakse Transformeri mudelite treenimiseks. Tulemused näitavad, et madalama märgenduste üksmeelega andmete puhul annab enamushäälega märgenduste ühestamine kõige usaldusväärsemad tulemused ja andmestiku. RoBERTa mudel näitab kõige paremat õppimis ja üldistamisvõimet.

Võtmesõnad:
Mitme märgendi klassifikatsioon, Transformerid, sümptomite tuvastamine, depressioon, sotsiaalmeedia

CERCS: P176 Tehisintellekt

Contents

1 Introduction

Predicting mental health problems based on social media posts has been a trending topic in Natural Language Processing (NLP). NLP researchers have found that machine learning can be used to detect mental health problems or symptoms related to mental health problems [1]. This topic is gaining popularity since mental health problems are prevalent and can have a negative effect on an individual's physical health and well-being [2].

Depression is one of the most common mental health problems and affects a large part of the population and has been related to a range of physical conditions [3]. NLP has been used to conduct diagnostic research in which individual's writings are analysed to detect depression [4, 5].

However, the symptoms for depression diagnoses can differ for people by showing a different clinical picture. Therefore, a diagnostic classification might not be informative enough. The ICD-10 [6] is used to for determining the diagnosis. The 10 depression symptoms are divided into two categories: main symptoms and additional symptoms. The main symptoms include low mood, low energy and loss of interests while additional symptoms include: disturbed sleep, poor concentration, low self-confidence, poor or increased appetite, suicidal thoughts or acts, agitation or slowed movements, guilt. The symptoms can be different for people, but must contain at least one of the main symptoms and none, one or multiple additional symptoms. For example, one individual might show symptoms related to low mood and poor concentrations while another individual might have loss of interests and disturbed sleep. As a result, the diagnosis and treatment of depression may be treated differently and can be based on presenting symptoms and not the generic diagnosis. Therefore, for detecting depression, it is more effective to detect the symptoms instead of the diagnosis. NLP can used for detecting symptoms of depressions by formulating a multi-label classification task where a model is trained to detect the symptoms as labels.

Social media platforms have become a more widely used source of data for mental health related NLP tasks. Guntuku et al. [7] presents the two main ways of assessing depression from social media. One is using answers of online psychological tests and use the answers in supervised machine learning task. Such tests do not always result in enough data [8]. The other option is extracting public social media data which is shared by people that have claimed to be suffering from depression. This kind of data can be found on sub-pages of social media platforms that are aimed to connect people with similar problems. There is also a third option which is manually annotating symptoms based on the social media posts.

In this work, a dataset consisting of posts collected from Reddit is analysed. Reddit is used by broad population and has a sub-page r/depression[1] where site users can write

[1] `https://www.reddit.com/r/depression/`

posts explaining their situation. The data used in this work is collected from the sub-page and consist of 2002 posts. The posts were manually annotated according to depression symptoms listed in ICD-10.

This work focuses on two research questions:

1. What is the quality of the annotated data
2. Can data with such quality and amount be used to train predictive models

The first question will be answered by examining the annotation agreement of the data. The second question will be answered by training Transformer models with the given data. This will include aggregating the annotations in three ways: union, majority and intersection of the annotations. The differently aggregated datasets will be used to train BERT [9], ALBERT [10] and RoBERTa [11] models.

The results showed that the dataset was heavily imbalanced and the annotation agreement of the data was low overall and also for some specific symptoms. The training results showed that the models are able to learn well on Union and Majority datasets while Intersection dataset had too few samples for the models to learn properly. Noticeably, RoBERTa model achieved the best results and Majority dataset is considered to be the most reliable dataset.

To the best of the author's knowledge, this kind of work that uses social media posts to predict depression symptoms has not been done before. The findings of this work will show if a small dataset with lower annotation agreement can be used to train predictive models and under which conditions. This will contribute to the overall research of predicting depression symptoms in individuals through their social media posts.

Section 2 will describe related works in the topic of detecting mental health issues from social media posts. Section 3 will give an overview of multi-label classification, Transformers architecture and models and evaluation metrics used in this work. In section 4 the source and annotation of data will be explained. This will include a short summary of the the annotation process. The processing and further analysis of data will be covered in section 5. Section 6 will describe the setup of the experiments and evaluation methods. Section 7 will describe the results for each dataset.

The code used in this work is available at GitHub repository[2].

[2] https://gitlab.com/KaireKoljal/predicting-depression-symptoms-based-on-reddit-posts-thesis/

2 Related Work

The most common method of detecting mental health problems form text is using binary classification. Al Hanai et al. [4] conducted a study where individuals went through depression screening. Their interest was to determine the state of the subject (depressed or not). The features used for the experiments were extracted from both text and video. The experiments used logistic regression models and additionally an LSTM model. Their results showed that depression can be detected better using LSTM compared to regression models. The work by Yang et al [5] also constructed a depressed/not-depressed classifier model. They used Paragraph Vector (PV) and Support Vector Machine (SVM) depression classification framework trained on interview transcripts for their text-based depression/non-depression classification. For both works [4, 5] the classification results were considered to be satisfying. Matero et al. [12] used data from Facebook to predict the degree of depression. They used BERT based models for their study in the role of contextual embeddings for depression prediction. Noticeably, it was pointed out that RoBERTa was the model that achieved state-of-the-art performance.

Another point of interest besides detecting depression diagnosis or a mental health issue is the detection of a specific symptom. Since 2017 the CLEF eRisk Lab[3] organizes tasks for early detection of mental health issues like depression, anorexia and thoughts of self-harm. One of the many teams that participated in the 2020 competition was team prhlt-upv that achieved one of the highest results [13]. The team prhlt-upv also described their results in their work by Uban et al. [14]. One of the tasks in the 2020 competition was self-harm detection which used Reddit posts and comments selected from relevant sub-reddits. Uban et al. [14] used BiLSTM with attention, Hierarchical Attention Network (HAN) and pre-trained BERT in their work and found that BERT shows the best results for detecting whether an individual is in risk of developing self-harm tendencies.

Focusing on solely depression symptoms detection is a more recent progress. Yadav et al. [15] based their data collection form Twitter on the Patient Health Questionnaire-9 (PHQ-9) that is used in clinical practice. They used ICD-10 based lexicon to collect relevant profiles and thoroughly pre-processed the tweets before annotating. They also had to take into account figurative speech and character limits of tweets by using auxiliary task of figurative usage detection when training their BERT model. Karmen et al. [16] used the ICD-10 as the basis for detecting symptoms from the data they collected from Psycho-Babble forum. They used the psychiatric jargon as basis and broadened the vocabulary by including synonyms and synonyms of synonyms to cover as many symptom related words as possible.

Different kind of social media platforms have been used for obtaining data. One of the popular platforms has been Twitter [15]. Facebook has also been used as a source of data by Matero et al. [12] and Schwartz et al. [17]. In works that use Facebook posts,

[3] https://erisk.irlab.org/

the authors of the posts also completed a personality questionnaire. Reddit has also been used by collecting posts from one or multiple relevant sub-reddits [14, 18].

The methods used for processing the social media content varies depending on the source of the data and the platform's limitations. Transformer models were used in multiple researches [12, 14, 15]. Most works concluded that using pre-trained models is beneficial and BERT based models show the best results. It was also pointed out in many cases that RoBERTa can achieve better results than BERT.

3 Technical Background

This chapter gives an overview of the background of methods and models used in this work. This includes the introduction the principles of multi-label classification and evaluation metrics. Additionally, the architecture and work of fastText model and Transformer models are explained.

3.1 Multi-label Classification

Classifying content is a supervised machine learning technique that is employed in contexts like image classification, text analysis, audio and many more [19]. The classification task can be divided into three domains:

- Binary classification
- Multi-class classification
- Multi-label classification

Binary classification task involves two distinct classes and the instances that need to be to classified belong to one of those classes, e.g. to classifying an instance as positive or negative.

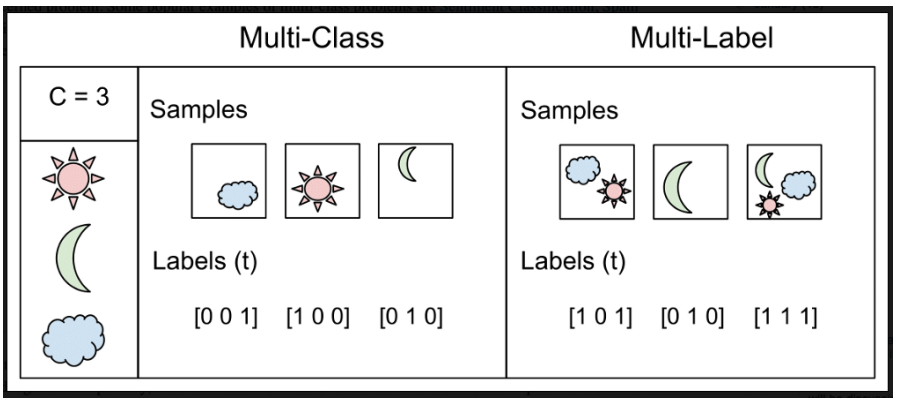

Figure 1. Multi-class vs multi-label classification [20]

Multi-class classification is used when there are three or more classes and the data instance belongs to one of these classes.

Multi-label classification is a supervised learning problem where an instance can be associated with multiple labels as opposed to the task of single-label classification. This means that the data instance can have simultaneously none, one or more than one of the labels [19]. An example comparison of multi-class and multi-label classification is shown in Figure 1.

Multi-label classification methods can be divided into two categories: a) problem transformation methods, and b) algorithm adaption methods [21]. Problem transformation methods transform multi-label problem into one or more single-label classification or regression problems. For example transforming the multi-label classification problem into binary classification or multi-class classification problems. Binary classification amounts to training independent binary classifier for each model which are collectively used to make predictions. Multi-class classification creates one binary classifier for every label combination in the training set. Algorithm adaption methods extend specific learning algorithm to handle multi-label data directly. For example, using k-nearest neighbors algorithm, decision trees or neural networks. In this work, the models chosen for the experiments implement the algorithm adaption method.

3.2 fastText

Linear classifiers are often used as baselines for text classification tasks. In their work, Bojanowski et al. [22] used their approach fastText to scale the baseline for larger corpus. The fastText model uses continuous bag of words (BoW) and linear classifier. The model learns the vectors from words using an n-gram of characters. The n-grams are used to help the embeddings understand suffixes and prefixes. This can result in a large number of unique n-grams that are hashed to reduce the size of the dictionary. The words are represented as the sum of the n-gram vectors. After obtaining the n-grams, the embeddings are learned by a skip-gram model which is a BoW model. The text representations are formed by averaging the embedded word representations to form the hidden variable. The text representations are then given as an input to a linear classifier [22].

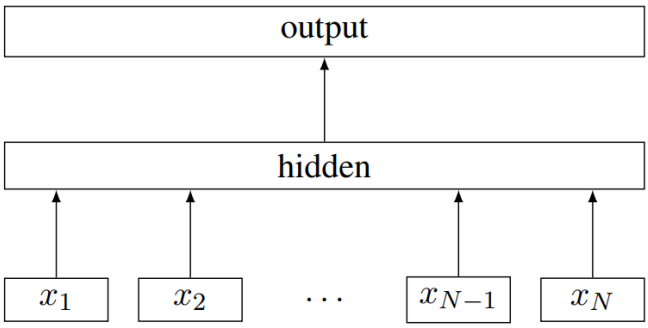

Figure 2. fastText architecture [22]

Figure 2 shows the fastText model architecture when given a sentence. The ngram features are marked as $x_1, ..., x_N$.

3.3 Transformers

This section gives an overview of the Transformer architecture and models used in this work that have Transformer architecture.

3.3.1 Transformer as Architecture

Transformer is a model architecture that relies on attention mechanism to retain relevant information of longer sequences. Using attention makes the Transformer's work parallelizable and more efficient than a RNN (Recurrent Neural Network).

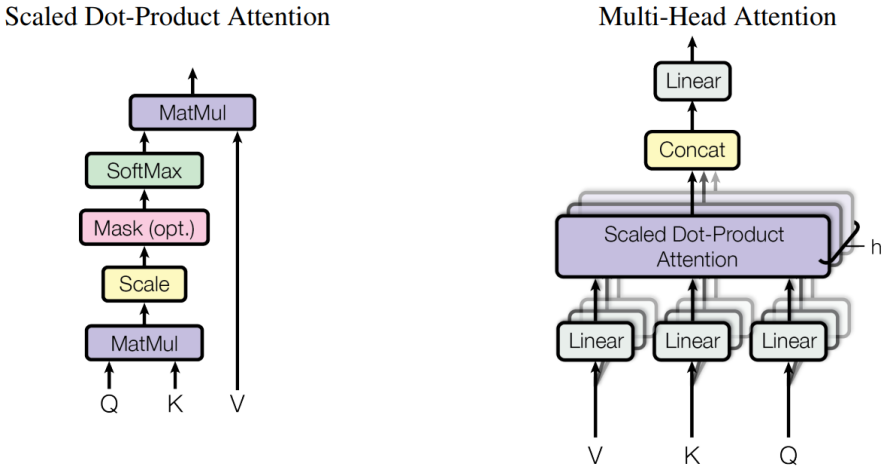

Figure 3. Scaled Dot-Product Attention (left). Multi-Head Attention (right) [23]

The Transformer model extracts the features of words by using self-attention. This is a sequence-to-sequence operation where a sequence of vectors are used as input and output. Using self-attention allows the model to understand how important the words are in relation to each other in the input sequence. Extracting this information does not require the use of recurrent units and instead can be done by using weighted sums over the input vectors and activations [24]. Simply put, the query vector and a set of key-value pairs vector are mapped to to an output vector by the self-attention function. If the input consists of queries and keys with dimension d_k and values with dimension d_v then a dot product is computed for queries with all keys and divided by $\sqrt{d_k}$. Applying Softmax function to these dot products results in the weight on the values. This method is referred to as Scaled Dot-Product Attention in the work by Vaswani et al. [23].

The Scaled Dot-Product attention is shown on the left side in Figure 3. In practice, the attention function is used simultaneously on a series of queries that are packed together into a matrix Q. The keys and values are packed into matrices K and V respectively.

Instead of using single attention function, Vaswani et al. [23] found that it is beneficial to use queries, keys and values linearly h times, as shown on the right side in Figure

3. Each projection then performs the attention function in parallel. The results are concatenated and projected again to get the final output values. This logic is called Multi-Head Attention is also shown on the right side of Figure 3.

The self-attention calculation is done in a transformer block that also includes additional feed-forward layers, residual connections and normalizing layers [24]. The input and output dimensions are the same for transformer blocks which enables them to be stacked like in the case of stacked RNNs. An examples of a transformer block in shown in Figure 4. The example consists of a single attention layer followed by feed-forward layer and layer normalization.

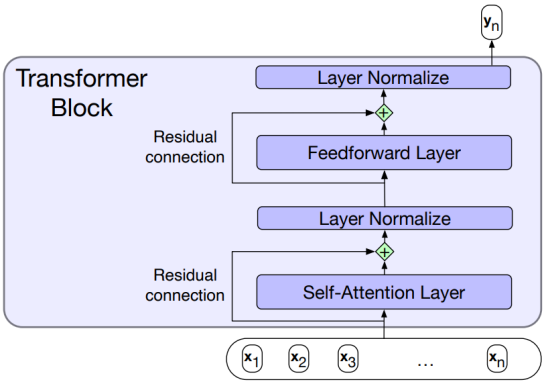

Figure 4. A transformer block [24]

The order of the words in input sentence is an issue because the self-attention is permutation invariant and the model does not contain recurrence or convolution [23]. To maintain the order it is necessary to have a representation of the word position and add it to the word embedding. One option is to use positional embedding that codes every position with a vector [23, 24]. These positional encodings are added to the input embeddings. These encodings have the same dimensions as the model so that they can be summed [23].

3.3.2 BERT

BERT is based on the architecture of a multi-layer bidirectional Transformer encoder which was proposed by Vaswani et al. [23]. BERT was proposed by Devlin et al. [9] as a transformer based language model that uses masked language modeling and next sentence prediction.

To handle a variety of downstream tasks, BERT's input representation is able to represent both a single sentence and a sentence pair. In this context, a sentence refers to a span of continuous text while pair of sentences can be for example a set of question and answer. An input token sequence will be referred to as 'sequence', which can be a single sentence or a sentence pair.

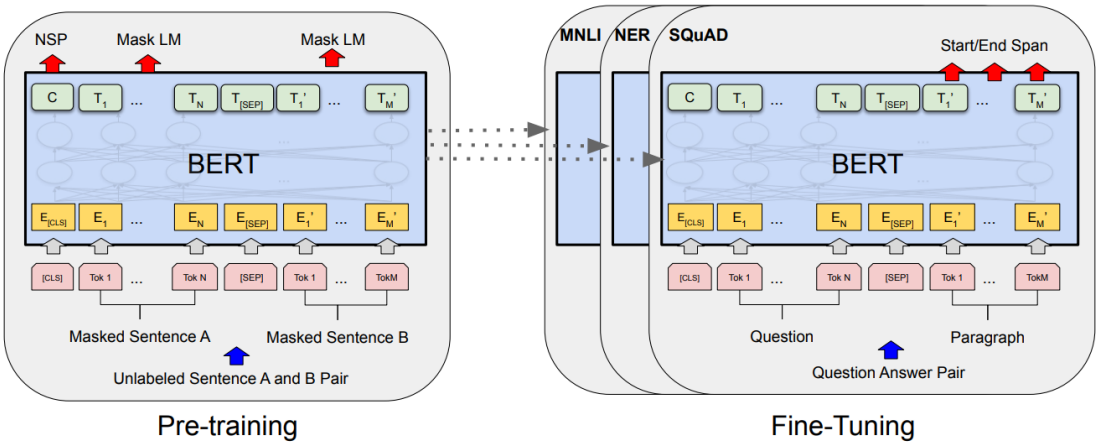

Figure 5. Pre-training and fine-tuning procedures for BERT [9]

Devlin et al. [9] explain in their work that the first token of every sequence is set to be a special classification token $[CLS]$. The final hidden state that matches that token is used as an aggregate sequence representation. This is used for classification tasks. Whether the input is a single sentence or a pair, it will be packed into a single sequence. The sentences are separated by a $[SEP]$ token. In addition, a learned embedding is added to every token to indicate which sentence it belongs to. As shown in Figure 5, the input embeddings are denoted as E, the final hidden vector of $[CLS]$ token as C and the final hidden vector for the i^{th} input token as T_i [9].

The context-independent representations are learnt by WordPiece embeddings while context-dependent representations are learnt by hidden-layer embeddings.

Devlin et al. [9] used two unsupervised tasks to pre-train BERT. The first task used masked language modeling (MLM) which randomly masks some percentage of input tokens and then predicts those masked tokens. This is done in order to train a deep bidirectional representation [9]. The final hidden vectors corresponding to masked tokens are then fed to an output softmax as in standard language modeling (LM). However, even though it creates a bidirectional pre-trained model, there is a mismatch between pre-training and fine-tuning since the $[MASK]$ token isn't present during fine-tuning. To mitigate this, the chosen token is replaced with $[MASK]$ token in 80% of cases, with a random token 10% and left unchanged in 10% of cases.

The authors [9] presented the second task as next sentence prediction (NSP). Many downstream tasks depend on understanding the relationship between two sentences. Since language modeling doesn't capture that, a binarized next sentence prediction task can be generated from any corpus, given that it is monolingual. BERT has been trained on sentence pairs, where in 50% of cases, sentence B is an actual next sentence to A and in 50% of cases B is a random sentence from the corpus. Sentences B are labeled as $IsNext$ and $NotNext$ accordingly. As can be seen from Figure 5, vector C is used

for NSP. This method has proven to be beneficial for downstream tasks line Question Answering and Natural Language Inference.

3.3.3 RoBERTa

It was suggested by Liu et al. [11] that BERT is significantly undertrained. The main faults were explained to be related to masking, NSP, text encoding and training batch size. For BERT, the masking is done only once during data processing which results in a single static mask that is given to model on every epoch. BERT uses NSP loss, which was hypothesized to be important as removing NSP would hurt performance. Bert also uses character-level Byte-Pair Encoding (BPE) with vocabulary size 30K. Additionally, the 1M steps with 256 sequences batch size for training leaves room for improvement.

Liu et al. [11] presented RoBERTa (Robustly optimized BERT approach) as a new Transformer model that is larger and has four modifications compared to BERT: 1) absence of NSP objective, 2) the masking pattern applied to training data is dynamically changed, 3) training is done on longer sequences, and 4) the model is trained longer with bigger batches, over more data.

Multiple experiments were conducted to address BERT's shortcomings regarding masking, NSP and trainig process. Compared to static masking, dynamic masking showed to be comparable or slightly better. Training without NSP loss showed that the performance of downstream tasks are slightly improved. This was compared to no NSP segment-pair and sentence-pair implementation. RoBERTa was also trained with larger byte-level BPE with 50K sized vocabulary. This was done without additional pre-processing or tokenization of the input. Although the performance was slightly worse on some tasks, Liu et al. [11] concluded that universal encoding scheme outweigh a minor degredation in performance. The batch size was increased during training process and it was observed that the perplexity for the masked language modeling objective improved in addition to end-task accuracy.

Through their experiments, it was found that the best combination for training would be using dynamic masking, no NSP loss, large mini-batches and larger byte-level BPE.

3.3.4 ALBERT

BERT's power comes from the use of context that provides the signal for learning. ALBERT uses parameter reduction techniques to enable better scaling of pre-trained models. Parameter reduction is significant as it enables the opportunity to have different embedding size for different words [25]. This allows to reduce the training time and about 70% of parameters can be reduced [10].

The backbone of ALBERT architecture is similar to BERT by using transformer encoder [23] and additionally GELU nonlinearities. According to Lan et al. [25], there are three main design improvements that ALBERT has over BERT.

One parameter reduction method is factorizing the word embedding matrix into two smaller matrices. This allows to grow the size of the hidden layers without having to significantly increase the parameter size of vocabulary embeddings [25].

The second method is using cross-layer sharing for model's parameters. For ALBERT, all the parameters are shared across layers. This makes the transitions from layer to layer smoother than for BERT. Also, weight sharing has shown to be effective for stabilizing network parameters [25].

In addition to masked language modeling (MLM) loss, BERT uses NSP which was designed to improve the performance on downstream tasks but was later found to be ineffective [25]. Instead for ALBERT, an inter-sentence coherence loss was proposed. For that, the sentence-order prediction (SOP) loss is used as it avoids topic prediction and focuses on modeling inter-sentence coherence [25]. When the model is forced to learn fine-grained distinction about coherence properties, the downstream task performance for multi-sentence encoding task is consistently improved.

3.3.5 Fine-tuning

Most pre-trained Transformer models can be fine-tuned for a specific task. Fine-tuning is simple due to self-attention mechanism that allows Transformers to model downstream tasks by swapping the appropriate inputs and outputs. If the sequence is text pairs, the model uses self-attention mechanism to unify the stages of independently encoding text pairs and applying bidirectional cross attention. At the output, the token representations are passed to output layer for token level tasks. The $[CLS]$ representation is passed to the output layer for classification [9]. Therefore, Transformers can be fine-tuned on different tasks by adding at least one additional output layer to suit the task. This has made Transformer models a popular choice for text classification, including multi-label classification.

3.4 Evaluation Metrics

During single label classification problems, partial correctness is not observed since the classification can be either correct or wrong. In multi-label classification, partial correctness of classification is common occurrence when at least one class is classified correctly and one or more class is classified in a wrong manner. Therefore, compared to other classification tasks, multi-label classification requires a different set of performance metrics [26].

The main evaluation metrics used to evaluate the training results in this work are F1-micro score, precision, recall and AUC-PR. The choices for choosing evaluation metrics were based on papers [26, 27] discussing evaluation metrics suitable for multi-label classification.

3.4.1 Precision, Recall and F1-score

For multi-class classification tasks, usually precision, recall and F1-score are calculated.

Precision denotes the fraction of predicted positive cases that are actually true positives and is calculated using the formula shown in (1). Recall measures the coverage of true positive cases and is calculated using the formula shown in (2). Both recall and precision focus on the positive class and assess the performance of a classifier for a given class.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

F1-measure combines precision and recall and is the harmonic mean of the two. The formula for calculating F1 score is shown in (3).

3.4.2 Micro- and Macro-Averaging

Micro-averaging focuses on studying the individual classes, weighing each sample or prediction equally. This averaging method is also able to capture possible class imbalance.

Macro aggregates the results over all classes, weighing each class equally. Macro-averaging computes the metric for each class independently and takes the average. This ensures that all classes are treated equally.

The micro- and macro-averaging can be applied to precision, recall and F1-score. Micro averaged F1-score is calculated using all predictions for all labels. This calculates the proportion of correctly classified instances. Macro averaged F1-score calculates F1-score for each label and then averages them. Therefore, F1-micro score is affected less by rare labels, while F1-macro weighs each label equally.

3.4.3 AUC Measures

Area Under Curve (AUC) is used to measure the classifier's ability to distinguish between classes. It aggregates over all possible thresholds and doesn't depend on one specific threshold.

In this work, AUC is used as a measure to find the most suitable threshold. Usually, AUC is used as a summary of ROC curve (AUC-ROC: Area Under Curve - Receiver Operating Characteristics). ROC calculates the true positive rate and false positive rate

for every threshold. AUC-ROC is calculated to evaluate how good the ROC curve is, and possibly use the score as a value for finding the optimal threshold.

Youden's index (J) is calculated for every point on the ROC curve and the maximum index value can be used to determine the optimal cut-off point (threshold). The value of Youden's index ranges from 0 to 1. Value 1 indicates perfect results, i.e. that the predictions included no false positives or false negatives.

The formulas for calculating the Youden's index is shown in (4), (5) and (6).

$$Sensitivity = Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity = \frac{TN}{FP + TN} \tag{5}$$

$$J = Sensitivity + Specificity - 1 \tag{6}$$

AUC can also be used together with precision-recall curve (AUC-PR: Area Under Precision-Recall curve). Precision-recall curve displays recall and corresponding precision for every threshold. The higher the curve is on y-axis the better the model performance.

For perfect classifier, the value of AUC-PR would be 1. A value of 0.5 shows that the classifier has not learned to differentiate between classes and the predictions are equivalent to a coin-toss. If the AUC-PR value gets closer to 0, then it shows that the classifier distinguishes between classes, but mixes them up.

In this work, ROC-PR is used as a measure for finding the optimal threshold. ROC-PR curve is used to determine the best cut-off point by finding the maximum Youden's index.

3.4.4 Annotation Agreement Measures

In classification tasks with multiple annotators, it is most likely that the annotators will have given at least a slightly different evaluation to the classified items. To determine the reliability and agreement among those annotator, several measures are used, including Fleiss' Kappa and Krippendorff's Alpha. Both measures range from -1 to 1, with 1 indicating perfect agreement. Zero indicates randomness and a score below zero indicates systematic disagreement among annotators.

Fleiss' Kappa and Krippendorff's Alpha are statistical measures for assessing reliability of agreement among a fixed number of raters who are given a task of rating or classifying items. Fleiss' Kappa measures agreement among annotators for each annotated class. This works for any number of raters as long as there are a fixed number of items. The kappa can be defined as shown in 7.

$$\kappa = \frac{P - P_e}{1 - P_e} \tag{7}$$

$1 - P_e$ is a degree of agreement that is attainable above chance. $P - P_e$ gives the degree of agreement that is actually achieved above chance.

Krippendorff's Alpha measures the overall agreement among annotators. The alpha can be defined as shown in 8.

$$\alpha = \frac{D_o}{D_e} \tag{8}$$

D_o denotes the disagreement observed and D_e shows the disagreement expected by chance.

4 Annotated Data

This section introduces the source of data and describes annotation rules. The labels used for annotations are also explained and examples of corresponding symptoms are provided.

4.1 Source of Data

This section describes shortly the collection of data and the annotation process. However, this was not organized by the author of this work and the annotated data was provided by the supervisor.

The data used in this work consists of 2002 randomly chosen Reddit posts from the r/depression sub-reddit that were written between 01.01.2014 and 31.12.2017. These posts consist only of the original post and do not contain any followups or comments. Any information about the author or date of post were removed.

The posts were manually annotated using Label Studio[4]. Annotation was done by computer science students who took the Natural Language Processing course in spring 2021. This means that the annotators were non-professionals who were given annotation guidelines and a list of symptoms with examples.

The posts were divided into groups of hundred. Each student was given 100 post to annotate, i.e one group of posts. Each group of posts was annotated by 3 students. None of the sets of 100 posts overlapped with other sets, resulting in three annotators for each post. Due to personal reasons, two students asked to be pardoned from the annotation task and therefore 200 posts had two annotators instead of three.

During the annotation process, the annotators had to follow guidelines:

- A label applies to a post only if the post contains a reference to a specific depression symptom.
- Only the symptoms that occur at the time of the writing can be labeled.
- Only the depressive symptoms that the post author describes about themselves can be annotated.

The annotators were given a list of ten labels to choose from, which are described and explained in chapter 4.2.

4.2 Label Set

The label set for annotation guideline consisted of ten most frequent depression symptoms along with explanations and examples.

[4] `https://labelstud.io/`

Symptom	Examples	Label
Persistent sadness or low mood	Expression of sadness and low mood, Also feeling of emptiness, Sadness from feeling lonely, Feelings of anxiety, irritability, hopelessness or helplessness	MOOD
Loss of interests and pleasure	Lack of interest in previously liked activities, The person does not find anything pleasurable.	INTERESTS
Fatigue or low energy	Expressions of not having energy, Feeling of fatigue even after good night sleep, Being tired all the time	ENERGY
Disturbed sleep	Expressions of sleeping too much, Expressions of not being able to sleep	SLEEP
Poor concentration or indecisiveness	Trouble concentrating and focusing on a task, Inability to remember specific details, Inability to make decisions, Difficulties thinking clearly	CONCENTRATION
Low self-confidence	Expressions of low self-esteem, Withdrawing from social relationships due to low self-confidence	CONFIDENCE
Poor or increased appetite	Eating too little, Eating too much, Recent rapid weight loss or weight gain	APPETITE
Suicidal thoughts or acts	Passive suicidal thoughts, Active suicidal thoughts, Thoughts about harming oneself	HARM
Agitation or slowing of movements	Restlessness, Feeling of having “slowed down”	MOVEMENTS
Guilt or self-blame	Feeling of worthlessness, Feelings of guilt, Expressions of blaming oneself	GUILT

Table 1. Symptoms, examples and corresponding label

According to ICD-10 (the International Classification of Diseases) [6], depression has a main diagnostic criteria or symptoms. These symptoms along with examples and corresponding labels are shown in Table 1. The symptoms are divided into main symptoms and additional symptoms. The main symptoms are the first three symptoms given in the table:

- Persistent sadness or low mood
- Loss of interests and pleasure
- Fatigue or low energy

Other symptoms given in Table 1 are additional symptoms. All the symptoms with examples and corresponding labels shown in Table 1 were used in the guidelines for the annotation task. For each symptom, multiple examples of behaviors were presented to help the annotators detect the presence of a symptom.

5 Data Processing

For this work, the annotations had to be analysed and new datasets had to be created for comparisons. This chapter describes the format of annotated data during processing and the analysis of annotations. The formation of new datasets and their analysis is also provided.

5.1 Annotation Analysis

Each annotator's set of annotations was stored in a separate JSON file. To make the processing of data easier, all the annotations were collected into one JSON file in format:

```
{"annotations":
{"id":0,
 "annotation1":[...],
 "annotation2":[...],
 "annotation3":[...],
 "text": "..." },
{"id":1,
 ...},
 ...
}
```

The values for "annotation1", "annotation2" and "annotation3" were lists of labels that could be empty, have one value or contain multiple labels.

Before data was processed any further, it was divided into train, development and test set. The data was partitioned based on 70/10/20 ratio. Therefore, train set has 1401 samples, development set has 200 samples and test set has 401 samples. This was done before any other partitioning or processing to ensure that all future datasets have the same train-development-test samples.

Since each post had 3 annotators, annotation agreement was computed to determine how easy it was to delineate the labels and how trustworthy the annotations are. Overall agreement was observed on the annotated data using Krippendorff's Alpha and Fleiss' Kappa with MASI distance. The results can be seen in Tables 2 and 3.

Looking at the results in Table 2 and the interpretation shown in Table 4, it can be said that overall agreement is fair. The agreement score for train, development and test sets are similar. To understand the agreement score better, agreement for each labels should be examined. From Table 3 it can be seen that the annotators' agreement varies over symptoms from slight to moderate. It needs to be taken into account that Fleiss' Kappa is bigger when there are less categories. In this case, there are ten categories and the results are expected. The highest agreement for labels, in this case moderate agreement, was achieved for labels 'HARM', 'APPETITE' and 'SLEEP'. The lowest agreement (slight) were for 'CONCENTRATION' and 'MOVEMENTS'.

Set	Alpha
All data	0.228
Train set	0.225
Development set	0.254
Test set	0.221

Table 2. Krippendorff's Alpha on data

Label	Kappa
MOOD	0.229
INTERESTS	0.293
ENERGY	0.375
SLEEP	0.459
CONCENTRATION	0.176
CONFIDENCE	0.259
APPETITE	0.546
HARM	0.565
MOVEMENTS	0.126
GUILT	0.242

Table 3. Fleiss' Kappa on data

Kappa value	Strength of Agreement
< 0.00	None
0.00 - 0.20	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.00	Almost Perfect

Table 4. Krippendorff's Alpha and Fleiss' Kappa interpretation [28]

In conclusion, the overall agreement and label agreements are not very trustworthy and therefore it raises a question about how to aggregate the annotations in this case.

5.2 Creating Datasets

To achieve a consensus of labels and examine models' ability to generalize, it was proposed to create 3 datasets that each use a different method for aggregating labels.

1. Union dataset
2. Majority dataset
3. Intersection dataset

The reason for creating three datasets was to have a variation and take into account human error during annotation. Since the annotation was done by non-professionals, different datasets allow to observe the differences in annotations and how the models' performance depends on this.

The Union dataset was formed by taking the union of all annotated labels. The list of annotations consisted of the labels that were present in any of the tree annotation lists and duplicate values were removed. This dataset would have the largest number of annotated labels and the best label representation.

The majority dataset was formed by taking into account the majority vote. Only the labels that were annotated by at least two annotators out of three were considered for the final annotation list.

The intersection dataset was formed by taking intersection of the annotations. Any annotated label would be counted only if all three annotators chose the label. This dataset would have the smallest set of annotations for each instance and the least versatile label representation.

An example of a post from the data is provided below to show the processing of instances for each dataset. Rephrased text of the post (rephrased for post author's protection) :
'stupid disease. its hard to deal with this, and its only been 2 months. i feel like my life is already over and im gonna be detached and miserable for the rest of my life. The worst thing is that i cant remember what it was like to feel. im losing memories and im becoming a husk. my memories are chronologically messed up, my memory from yesterday is as vague as memories from 2 years ago. im confused all the time and my reaction time has slowed and it is also affecting my social life. im getting more awkward. it feels like its fine until i think about it, then i lose it and turn into a different person who cant remember anything about whats going on. please help me. im sorry if its hard to follow. i just needed to get that off my chest.'

annotation1:['MOOD', 'INTERESTS', 'CONCENTRATION', 'MOVEMENTS'],
annotation2:['CONCENTRATION'],
annotation3:['MOOD', 'ENERGY', 'CONCENTRATION'],

Annotations for Union dataset:
['CONCENTRATION', 'ENERGY', 'MOOD', 'INTERESTS', 'MOVEMENTS']
Annotations for Majority dataset: ['CONCENTRATION', 'MOOD']
Annotations for Intersection dataset: ['CONCENTRATION']

The underlined parts of the text show the possible interpretations of symptoms. This example also illustrates how difficult it is to detect symptoms from similar posts.

5.3 Label Analysis of Datasets

The distribution of labels in train, development and test sets can be seen in Table 5. The table presents the number of posts that were assigned a certain label.

	Union			Majority			Intersection		
Label	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
MOOD	1108	164	321	720	109	210	303	48	82
INTERESTS	300	39	87	95	12	23	32	5	7
ENERGY	195	32	58	67	13	22	28	5	5
SLEEP	138	24	31	66	11	15	23	3	5
CONCENTRATION	153	24	48	34	1	8	8	0	0
CONFIDENCE	509	66	120	190	26	42	55	1	10
APPETITE	67	7	15	37	2	8	17	1	2
HARM	406	69	120	262	42	73	131	17	40
MOVEMENTS	83	8	19	8	3	1	3	0	0
GUILT	467	66	136	162	27	53	40	3	11
Posts with labels	1289	186	365	986	144	271	521	71	137
Posts with no labels	112	14	36	415	56	130	880	129	264

Table 5. Statistics of labels in train, development and test sets in each dataset

It can be seen from Table 5 that the datasets are heavily imbalanced, especially Union dataset where the most frequent label is 'MOOD'. It is one of the main symptoms and therefore it is expected that it would occur the most in the dataset. During data processing it was also noted that 'MOOD' was assigned most often just by one annotator instead of three. This again refers to inconsistency and disagreement among annotators.

Comparing the datasets, we can see that some labels like 'CONCENTRATION', 'APPETITE' and 'MOVEMENTS' have very few samples in Majority and Intersection sets. This could affect the performance of models since there might not be enough samples to train on.

Figure 6 shows the distribution of texts over labels in all datasets. It can be observed that in case of every set, the label 'MOOD' is annotated most often, having the number of occurrences at least double compared to other labels. 'CONFIDENCE', 'HARM' and 'GUILT' are also annotated many times. Other labels, like 'APPETITE' and 'MOVEMENTS' are annotated the least across all datasets.

In each of 3 datasets the number of labels per text differs. Figure 7 shows that in Union dataset there are more labels per texts compared to other datasets. This is an expected results since according to the dataset forming rule, all proposed labels were counted. Therefore the number of posts without labels is considerably lower than for other datasets. For Union dataset, it can be seen that the posts have most likely two or

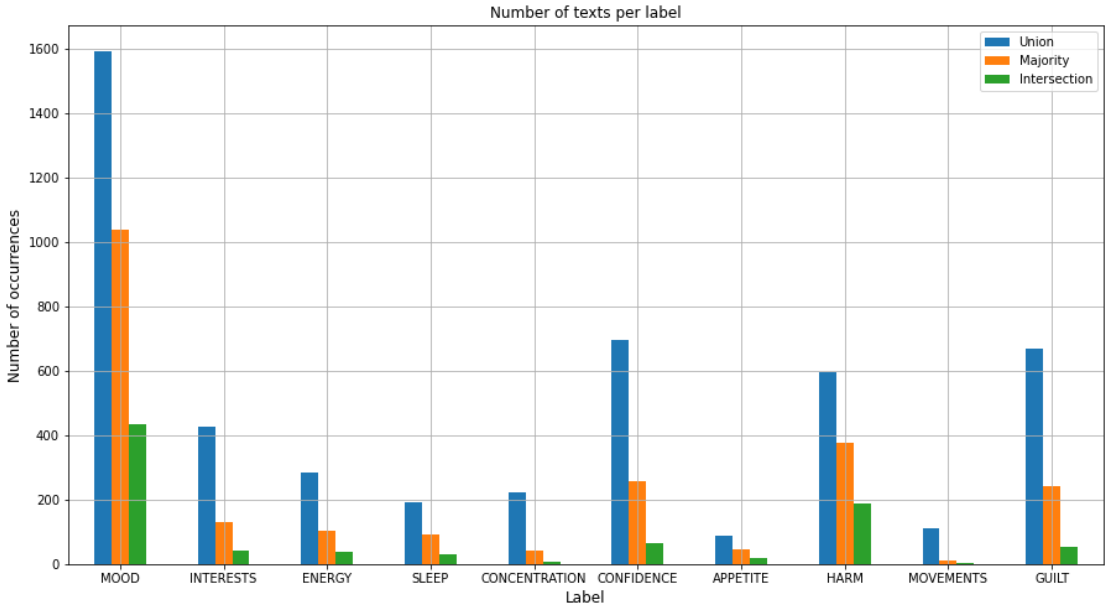

Figure 6. Number of texts per label

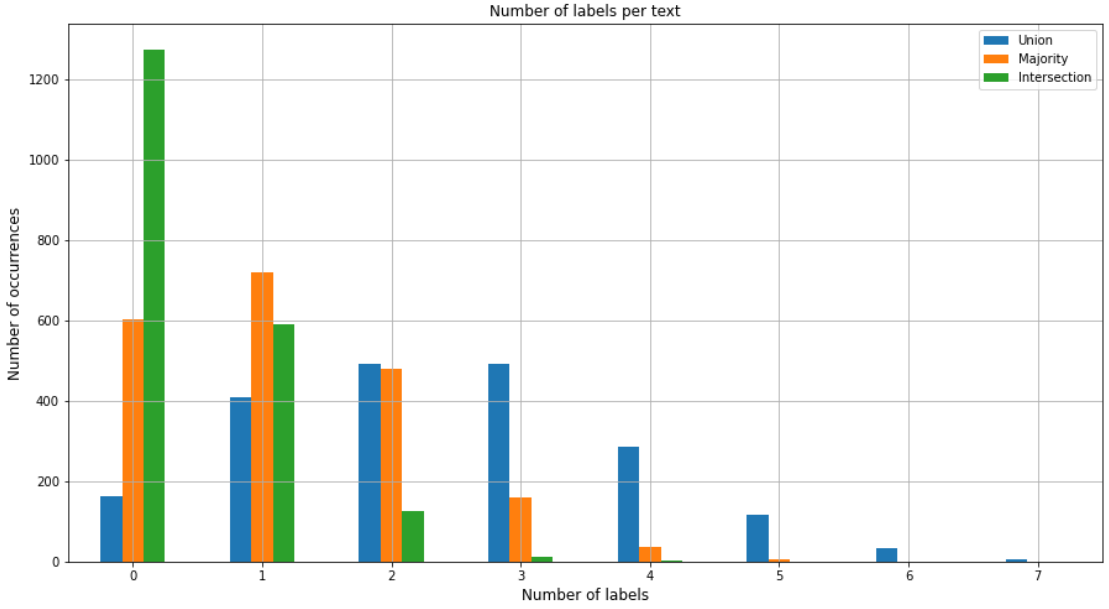

Figure 7. Number of labels for texts

three labels. For Intersection dataset, the posts will most likely have no labels or have one.

6 Methodology

This section gives an overview of the classification experiment's setup and explains the choices made in the model selection. In addition, the modification of models and training specifications will be described.

6.1 Baseline Model

To better evaluate the performance of the Transformers models, a baseline model was trained. The baseline model used in this work is fastText[5].

The fastText model was trained using data files that were modified to suit the input format of the model. The models were trained for 15 epochs with learning rate 0.9[6]. The loss function was set to 'ova' (one-vs-all) to enable multi-label classification. The wordNgrams (maximum length of word ngram) parameter was set to 2. The parameters for minimum and maximum character ngrams were set to 3 and 5 respectively. Additionally, the pre-trained english vectors used for training were available in fastText documentation[7].

During model evaluation, the whole test file was given to the model. The model was set to return 2 labels (k=2) and threshold was set to 0.25. Model was set to return two labels since the more common number of labels for posts were two rather than one. Threshold 0.25 was set after preliminary testing that showed that to be the optimal threshold. The performance was evaluated with precision, recall and F1-score.

6.2 Transformer Models

The selection of models was based on previous works [15, 12] that used Transformers models for multi-label classification tasks. Due to the small size of the datasets, it was decided to use pre-trained Transformers models instead of creating and training a new model. For this work, HuggingFace[8] library was used to acquire the pre-trained BERT models.

In total, three BERT models were used:

1. bert-base-uncased[9]
2. roberta-base[10]

[5] https://fasttext.cc/
[6] Based on preliminary testing
[7] https://fasttext.cc/docs/en/crawl-vectors.html
[8] https://huggingface.co/transformers/
[9] https://huggingface.co/bert-base-uncased
[10] https://huggingface.co/roberta-base

3. albert-base-v2[11]

The bert-base-uncased was chosen as the basic BERT model. The uncased model was chosen since case information in text was not considered to be important in this work. Roberta-base was chosen as a slightly larger model for comparison. Bert-base-uncased has 110M parameters while roberta-base has 125M parameters. The albert-base-v2 was chosen as a smaller model with 11M parameters. It was favored over albert-base-v1 for having additional training data and longer training[12].

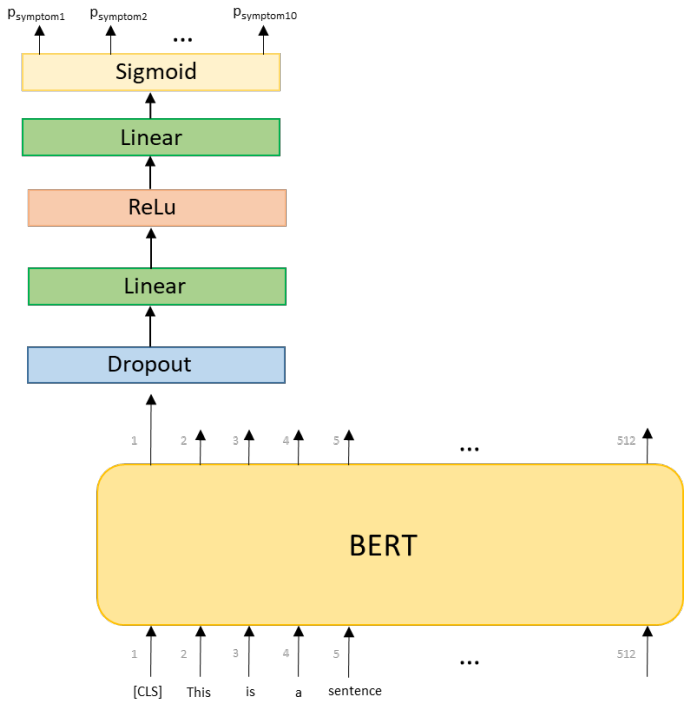

Figure 8. Classifier implemented on the basis of BERT

Hugging Face[13] provides examples of custom BERT classes for different downstream tasks, including multi-label classification[14] which was used as the basis for the classifier in this work.

A new class BERTClass was created which redefined the init() and forward() functions. The new class uses the pre-trained models as basis and adds 4 additional layers to create a classifier, illustrated in Figure 8. The additional layers were added to have an activation function and have the correct number of outputs. The Dropout regularization was

[11] https://huggingface.co/albert-base-v2

[12] https://huggingface.co/transformers/v3.3.1/pretrained_models.html

[13] https://huggingface.co/docs/transformers/index

[14] https://github.com/abhimishra91/transformers-tutorials/blob/master/transformers_multi_label_classification.ipynb

set to 0.3. The Linear layers used the models' and output layer dimensions. The output dimension was 768 for BERT, RoBERTa and ALBERT. The additional dimension was set also to 768[15] for both Linear layers. The layers were implemented using PyTorch[16].

6.3 Training Details

The training process was carried out at the University of Tartu High-Performance Computing Center [29]. To train the models, Transformer's Trainer class was used. A custom class MultilabelTrainer was created which redefines the compute_loss() function. The loss function was replaced with BCEWithLogitsLoss function based on the example given on the Trainer class page. Using BCEWithLogitsLoss allows the model to assign independent probabilities to labels. The Trainer used default adam_hf optimizer with given learning rate and training parameters.

During model training, logging and evaluation strategy was set to "epoch". After training was completed, the best model was loaded at the end. The batch size was set to 32 and sequence length was set to 512, i.e. the maximum possible. The number of epoch was set to 50 with early stopping on a condition that the evaluation metric hadn't changed in 10 epochs.

Preliminary testing on the development set with learning rates included 5e-4, 1e-4, 5e-5, 3e-5, 1e-5 and 1e-6. Based on the results, all further work was done with learning rates 5e-5 and 3e-5 as these gave the best results. The tests showed that there weren't significant differences in the results for the same model when trained with different learning rates. However, the learning rates affected the results across different models and thus both learning rates were included in the final training process.

Each model, dataset and learning rate combination was trained with five different seeds to calculate AUC-PR and to find the most optimal threshold along with standard deviation. The five seeds used in the work were: 12321, 23432, 34543, 56765, 78987.

Since the models' output consist of ten probabilities, it was necessary to find the threshold to determine the decision point for mapping the probabilities to ones and zeros. AUC-PR was used to find the threshold value for each dataset. Initially, AUC-ROC was used. However, this did not take into account precision, which was important in the context of this work. It was decided to use AUC-PR instead.

During training, the best threshold was recorded for each model. It was considered to round the thresholds to two decimal points and find one optimal threshold for each dataset. However, for different models within the same dataset, the range of threshold values varied greatly. The thresholds for Union dataset ranged from 0.22 to 0.37, for Majority from 0.21 to 0.35 and for Intersection from 0.28 to 0.34. Therefore an average

[15]Preliminary testing showed that changing the dimension has no significant influence on the models' performance

[16]`https://pytorch.org/docs/versions.html`

of these values was used as the optimal for each dataset. The average threshold for Union dataset was 0.29, for Majority dataset 0.30 and for Intersection dataset 0.31. It was decided to round the thresholds to one decimal place. This resulted in the same optimal threshold for all the datasets which was 0.3. Optimal thresholds for different models were also observed. The average threshold for ALBERT was 0.26, 0.31 for BERT and 0.32 for RoBERTa. Averaging these to one decimal point would also result in threshold 0.3. Therefore, for the evaluation, the threshold value was set to constant 0.3. This was applied for final evaluation of models on train, development and test sets.

6.4 Evaluation

The metrics used for evaluation were F1-micro score, precision and recall, that were explained in Section 3.4. The most frequently used metric for estimating the performance of learning systems in classification problems is accuracy. Due to the large imbalance of samples for each label in the data, accuracy is not used as a metric for evaluation in this work. However, accuracy is measured during training and evaluation for informative purposes.

7 Results

This section gives an overview of the training and evaluation results. An analysis of models' general performance, symptom based evaluation and qualitative analysis is described. The results in Tables 7, 8, 9 and 12 are averages over five seeds for each learning rate and presented with standard deviation.

7.1 fastText

The baseline model was also evaluated on all the datasets. The training showed that the models either tend to overfit to some labels or are not able to learn well. Therefore, the results of the best combination are presented. The results are shown in Table 6.

	Development			Test		
Dataset	F1-micro	Precision	Recall	F1-micro	Precision	Recall
Union	0.537	0.682	0.442	0.546	0.685	0.454
Majority	0.456	0.519	0.407	0.484	0.533	0.443
Intersection	0.556	0.569	0.542	0.587	0.61	0.566

Table 6. fastText model results

The fastText models' evaluation on development set shows that the models do better on Majority and Intersection datasets. The F1-score for Union development set was 0.537 and 0.546 on test set. Evaluating individual labels on test set showed that the model tends to over-predict 'MOOD' while other labels are not predicted as often as they actually appear in the test set. Additionally, the number of posts predicted to have no symptoms is considerably lower than the actual number of posts with no symptom labels. 'MOOD' is also the label with the most samples and considering how the model over-predicts this symptom, the model most likely overfits.

The model for Intersection dataset achieved F1-score of 0.556 on development set and 0.587 on test set. However, the Intersection models seems to be overfitted, since for individual label predictions on test set, the model predicted only labels 'MOOD', 'CONFIDENCE' and 'HARM'. Those cases made up 94% of the labeled posts. Therefore the baseline model learned to predict the most common labels.

For majority dataset, the results for development and test set were 0.456 and 0.484 respectively. Label evaluation on test set showed that the model predicts well 'MOOD' and can also label the posts with no symptoms accordingly. Although the model still seems to predict mostly the most common labels it's symptom prediction regarding other labels is better than compared to models for Union and Intersection datasets.

7.2 Training Results

For each dataset, models were trained with different seeds and learning rates. For Transformers models, three different models with two different learning rates trained with 5 different seeds amounted to 30 models that were trained for each dataset.

As the datasets are small, the main concern was the models' ability to learn. Therefore the trained models were evaluated also on train sets, to determine how well the models were able to learn.

Model + lr	Accuracy	F1-micro	Precision (micro)	Recall (micro)	F1-macro
ALBERT 3e-05	0.697±0.325	0.904±0.109	0.875±0.136	0.937±0.076	0.835±0.181
ALBERT 5e-05	0.602±0.297	0.866±0.114	0.827±0.143	0.913±0.079	0.762±0.191
BERT 3e-05	0.741±0.173	0.919±0.054	0.876±0.075	0.968±0.031	0.862±0.087
BERT 5e-05	0.900±0.090	0.971±0.026	0.951±0.044	0.992±0.007	0.948±0.046
RoBERTa 3e-05	0.552±0.104	0.876±0.031	0.815±0.043	0.947±0.026	0.810±0.044
RoBERTa 5e-05	0.405±0.156	0.824±0.058	0.762±0.063	0.899±0.066	0.736±0.105

Table 7. Training results for models on the Union dataset

Table 7 shows the training metrics for all models trained on Union dataset. It can be seen that the accuracies don't reach 100% during training which indicates that the model is learning but is not able to overlearn since the dataset is small.

The F1-micro scores are within normal range and high enough to indicate proper learning and ability to differentiate between labels. Precision and recall (both micro) are also high and indicate a good learning ability. This is also backed up by F1-macro scores that are not significantly smaller than F1-micro scores.

Similar results are shown for Majority dataset in Table 8 where it can be seen that the accuracy is relatively high and on average higher than on Union dataset. The only exception is the ALBERT model trained with learning rate 5e-05. The scores for F1-micro, precision and recall are also high and indicate proper learning ability.

Model + lr	Accuracy	F1-micro	Precision (micro)	Recall (micro)	F1-macro
ALBERT 3e-05	0.917±0.084	0.949±0.053	0.801±0.107	0.937±0.063	0.961±0.042
ALBERT 5e-05	0.499±0.196	0.664±0.152	0.361±0.249	0.650±0.114	0.684±0.195
BERT 3e-05	0.854±0.058	0.899±0.037	0.696±0.077	0.862±0.048	0.938±0.026
BERT 5e-05	0.908±0.087	0.942±0.055	0.821±0.121	0.916±0.078	0.970±0.030
RoBERTa 3e-05	0.758±0.171	0.856±0.096	0.646±0.143	0.821±0.128	0.899±0.059
RoBERTa 5e-05	0.885±0.101	0.929±0.064	0.773±0.105	0.903±0.091	0.959±0.031

Table 8. Training results for models on the Majority dataset

For the Intersection dataset, the accuracies on the train set do not generally reach as high as for Majority and Union datasets. The F1-micro scores remain low, mostly zeros

or staying between 0.26 and 0.6, with some exceptions when the score was from 0.7 to 0.93. For ALBERT, training with learning rate 3e-05 showed higher results on average, reaching 0.615±0.132. The other models had a relatively low F1 score around 0.273 to 0.488 with a standard deviation that is mostly as big or bigger than the average value of the scores.

Evaluation on train set shows that Majority dataset is more reliable than Union dataset regarding annotations. Also, models are unable to sufficiently learn from Intersection dataset. Based on these results, it was decided to treat Majority dataset as a main dataset for model evaluation and results for Union and Intersection datasets are for given for comparison purpose.

7.3 Development and Test Set Results

The development and test set results for the Majority dataset are shown in Table 9. As shown in the table, the F1 scores for development and test set reach 0.612 and 0.624 respectively, indicating that the models have learned and are also able to generalize to the test set.

	Development			Test		
Model + lr	F1-micro	Precision	Recall	F1-micro	Precision	Recall
ALBERT 3e-05	0.576±0.019	0.581±0.014	0.573±0.036	0.594±0.014	0.581±0.008	0.608±0.027
ALBERT 5e-05	0.542±0.038	0.559±0.035	0.530±0.066	0.549±0.047	0.540±0.028	0.563±0.083
BERT 3e-05	0.594±0.008	0.591±0.026	0.600±0.028	0.604±0.015	0.575±0.026	0.636±0.025
BERT 5e-05	0.612±0.014	0.584±0.019	0.645±0.029	0.612±0.016	0.564±0.015	0.670±0.032
RoBERTa 3e-05	0.599±0.010	0.598±0.035	0.604±0.035	0.614±0.010	0.579±0.033	0.656±0.035
RoBERTa 5e-05	0.605±0.010	0.586±0.011	0.624±0.018	0.624±0.015	0.576±0.024	0.680±0.013

Table 9. The Majority dataset metrics for development and test sets

The F1-micro scores are slightly higher on test set compared to development set. The only exception to this a BERT model trained with learning rate 5e-05 that had the same average F1-score on both development and test set. On both development and test sets, BERT and RoBERTa models trained with learning rate 5e-05 show slightly higher F1-scores than models trained with learning rate 3e-05. ALBERT achieved higher F1-score with learning rate 3e-05. The learning rates show some influence as the differences between F1 scores for the same model are from 0.006 to 0.034 on development set and 0.008 to 0.045 on test set.

For models trained on Union dataset, the averaged scores for F1-micro and recall were slightly higher compared to Majority dataset. The differences of scores on development and test set are not significant and don't indicate over-learning and show a good generalization ability. The models trained on Intersection dataset showed very low scores with the worst performing model being RoBERTa with F1-micro score 0.099±0.161 on

development set and 0.087±0.134 on test set. Compared to the scores on training set, the results indicate overfitting. However, comparing development and test set results, the model was able to generalise.

Based on the results on Majority dataset, RoBERTa models achieved the highest F1-scores. In comparison, RoBERTa was also the best performing model on Union dataset and the worst on Intersection dataset. For Intersection dataset, the best performing model was ALBERT. This raised the question whether some best performing models could also be the worst regarding development and test sets.

7.4 Best Performing Models

From the 30 models trained on Majority dataset, one model with highest F1-score for ALBERT, BERT and RoBERTa was chosen for comparison on development and test sets. To observe any differences between different model types, one model for each type was chosen regardless of seed and learning rate. This comparison was done to determine the best performing model type for further analysis and to evaluate whether one best performing model could be the worst performing model on another set. The comparison is done on Majority development and test sets.

The best performing models' results are shown in Tables 10 and 11. The selection for models in Table 10 was made based on the F1-micro scores on development set, and for models shown in Table 11, F1-micro score on test set was considered. Each model has it's own ranking from 1 to 10, with smaller rank indicating higher F1 score. The ranks are set from 1 to 10 since 10 models for each model type was trained (5 seeds and with 2 learning rates). Column 'Test Rank' in Table 10 shows the model's rank on the test set. Column 'Dev Rank' shows the model's rank on development set.

Model	lr	Seed	Train F1-micro	Dev F1-micro	Dev precision	Dev recall	Test F1-micro	Test Rank
ALBERT	3e-05	23432	0.97	0.595	0.593	0.598	0.609	1
BERT	5e-05	23432	0.976	0.626	0.585	0.675	0.622	2
RoBERTa	5e-05	12321	0.971	0.618	0.602	0.634	0.628	3

Table 10. Majority dataset: Best models based on F1-micro score on development set

Table 10 shows the models for ALBERT, BERT and RoBERTa that achieved the highest F1-scores on development set. For comparison, F1-scores for train and test set are shown. Based on the development set results, the best performing model out of three is BERT. But it did not achieve the highest F1-score on test set compared to the other two models. None of the best models on development set were the worst performing models on test set.

Model	lr	Seed	Train F1-micro	Dev F1-micro	Test F1-micro	Test precision	Test recall	Dev Rank
ALBERT	3e-05	23432	0.97	0.595	0.609	0.593	0.626	1
BERT	5e-05	56765	0.99	0.621	0.634	0.586	0.69	2
RoBERTa	5e-05	78987	0.947	0.592	0.642	0.599	0.692	9

Table 11. Majority dataset: Best models based on F1-micro score on test set

Table 11 shows the best performing models on test set. The highest F1-score on test set was achieved by RoBERTa. The same model also placed ninth in the RoBERTa models ranking on development set. BERT and ALBERT did not achieve as high F1-scores on test set but were placed in the higher end of the ranking on development set. The best ALBERT model was the same model for development and test set.

BERT achieved the highest F1-score on development set and second highest on test set. The model also had a high rank on the other set. While RoBERTa achieved the highest F1-score on test set, it was almost the worst performing model on development set. For Union dataset, the best performing model on test set was BERT while RoBERTa had a higher F1-score on development set.

The comparison of the best performing models do not all have the same learning rate. However, best BERT and RoBERTa models were trained with learning rate 5e-05 and also achieved higher F1-scores. There are also no clear sign that using one specific seed helps to achieve a better score.

The best model comparisons show that BERT could be preferred since it is more stable regarding the ranks and also doesn't achieve a significantly lower F1-score on test set than RoBERTa. However, RoBERTa showed better results on test set (Table 9) and the best performing model on test set was also RoBERTa. Based on these results, it was decided to use RoBERTa model as the model to evaluate individual label predictions.

7.5 Symptom Based Evaluation Results

To understand the results of overall evaluation, it is also important to look at individual label predictions to see how well the models perform. The main analysis for each label is done on Majority dataset using RoBERTa models. This analysis is done to see if the F1-scores for individual labels are even and how well RoBERTa could learn to detect corresponding symptoms. The RoBERTa models are evaluated on Majority test set for models trained with both learning rates. The results were averaged over five seeds.

The label evaluation was done using confusion matrices for each label. Precision, recall and F1 score for each label was also calculated. The results are shown in Table 12. Since there were many cases in which the models did not predict a specific label for any test sample, many of result values were 'nan' (not a number). The number presented in

the brackets behind the scores shows the number of models that gave a numeric prediction and over which the average and standard deviation is taken. No number in the brackets indicates that the average is taken over five values.

	RoBERTa 3e-05			RoBERTa 5e-05		
Label	Precision	Recall	F1-score	Precision	Recall	F1-score
MOOD	0.632±0.026	0.857±0.065	0.726±0.008	0.630±0.020	0.857±0.034	0.726±0.015
INTERESTS	0.481±0.156	0.252±0.152	0.316±0.167	0.420±0.028	0.470±0.104	0.438±0.055
ENERGY	0.611±0.242	0.200±0.149	0.267±0.135	0.583±0.190	0.327±0.122	0.414±0.137
SLEEP	0.410±0.092	0.280±0.173	0.308±0.148	0.461±0.091	0.373±0.060	0.406±0.048
CONCENTRATION	0.500 (1)	0.050±0.112	0.333 (1)	0.233±0.252 (3)	0.050±0.068	0.177±0.033 (2)
CONFIDENCE	0.418±0.077	0.524±0.079	0.457±0.041	0.471±0.072	0.448±0.026	0.458±0.045
APPETITE	0.167±0.236 (2)	0.025±0.056	0.182 (1)	0.320±0.344	0.175±0.190	0.376±0.182 (3)
HARM	0.581±0.034	0.773±0.048	0.661±0.014	0.567±0.044	0.836±0.032	0.674±0.033
MOVEMENTS	nan	0	nan	nan	0	nan
GUILT	0.543±0.056	0.475±0.121	0.495±0.045	0.539±0.076	0.457±0.062	0.487±0.019

Table 12. Symptom based evaluation on the Majority test set

The results show that the models could not detect label 'MOVEMENTS' regardless of the learning rate. Additionally, for labels 'CONCENTRATION' and 'APPETITE', some average scores are calculated based on less models since some models gave 'nan' values. For the model trained with learning rate 3e-05, the number of 'nan' values was greater. The aforementioned labels also had less samples in the dataset, as shown in Table 5 and the results are expected. However, it can be also seen that learning rate 5e-05 is more suitable for RoBERTa model in this case.

Label analysis was also done with BERT model. Learning rate 3e-05 gave more 'nan' values for 'CONCENTRATION'. Additionally, BERT models also had a hard time detecting 'ENERGY' as for both learning rates the scores included at least one 'nan'. Overall RoBERTa achieved better scores than BERT.

These results show that even when the general F1-scores are high and different learning rates don't show significant difference in general results, examining individual label scores shows that learning rate affects the models' ability to generalize. Based on these results, learning rate 5e-05 enabled better generalization ability.

7.6 Qualitative Analysis

To further investigate the model's ability to generalize, specific samples were given to RoBERTa models trained on Majority dataset to evaluate their ability to detect symptoms. The results were counted for 10 models that included all RoBERTa models trained with learning rate 3e-05 and 5e-05. For evaluation, one sample was selected from train set

and two others were selected from test set. Both rephrased and original texts were given to the models to detect symptoms.

The rephrased sample post for train set was also presented in section 5.2. A sample was chosen from train set to provide a sample that the models have seen during training. It would also serve as a comparison to the results of the rephrased text.

The labels of the post were 'MOOD' and 'CONCENTRATION'. The models could all correctly detect the 'MOOD' label. For 'CONCENTRATION', only two models out of ten could predict the label. In comparison, if given the original post to the models, only five models predicted the 'CONCENTRATION' label. It was expected that 'MOOD' was detected as it was one of the main two symptoms and also had the most occurrences in the data. Since label 'CONCENTRATION' had very few samples in Majority train set, it was expected that the models could not learn this label properly and therefore would be bad detecting it.

Two more samples were picked from the Majority test set and both original and the rephrased texts were evaluated. The first test sample had labels 'MOOD', 'SLEEP' and 'APPETITE'. For these labels, given the original post, nine models could detect correctly 'MOOD'. Eight models could detect 'SLEEP' and none of the models detected 'APPETITE'. Given the rephrased version of the same post, eight models detected 'MOOD', all ten models detected 'SLEEP' and none of the models detected 'APPETITE'. For the test set, 'SLEEP' and 'APPETITE' also had relatively few samples. Therefore bad results for these labels is also expected for the same reason as for label 'CONCENTRATION'.

The second test sample had labels 'MOOD', 'CONFIDENCE' and 'GUILT'. Out of ten models, all detected 'MOOD, eight detected 'CONFIDENCE' and five detected 'GUILT' from the rephrased text. With the original post, ten models detected 'MOOD' and 'CONFIDENCE', and six detected 'GUILT'. For this test sample, a post with three out of four most common labels was selected. It was unexpected that the models could not detect 'GUILT' that well.

The post below is the rephrased version of the second test sample.
I made this throw-away account because I feel so done for. I hate what kind of person I am. I dont think Im very attractive. Im also very weak. I mess up end up saying the wrong things all the time. I feel like Im a bad friend because Im not there for my friends enough. I find it hard to show empathy and talk to people. An to top it all, Im lazy. I can see and understand my problems but Im not doing anything to fix them. I don't know how to fix them and Im running out of ideas what to do. Im behind with work, I havent even started on a paper that is due in few hours. Im a college freshman and the finals week is so stressful for me. I cant continue living like this if I want to give up all the time. This might be the wrong sub to talk about this. Most likely I dont have depression but Im exhausted and have noone else to turn to.

The same post with union annotations was given to RoBERTa models trained on

Union dataset. All the models could detect all necessary symptoms for both original and rephrased post. This indicates that even though the Majority dataset is cleaner than Union, it is still lacking samples to learn well enough.

7.7 Discussion

The overall annotation agreement of the data was fair. This is most likely due to the inexperience of the annotators and the difficulty level of the task. Most of the individual label agreements were also on the lower end and affected the overall score. It was also seen that the labels with higher agreement had also more samples in the dataset.

Creating different datasets helped to analyse the training results and also show which annotation aggregation would be preferred for such data. The Union dataset consisted of all proposed labels and is therefore more unreliable. The intersection method is often used in similar works but in this case this method is not suited due to the very low annotators' agreement. The Intersection method could result in artificially clean dataset in case of smaller dataset like the one used in this work. The Majority dataset used majority vote for annotations and is therefore the most reliable out of the three datasets. For similar data, intersection method could work, provided that the annotators have a more professional background or the annotation is done by a model of similar ability. Union dataset could be considered on similar ground. Based on the results of this work, majority method could be preferred for similar data.

The training evaluation of fastText showed that the models tend to overfit. The main problem concerned label 'MOOD', that was prevalent in all datasets. The overall scores were acceptable, but were most likely the result of predicting the most common labels. The Transformers models also had scores in the same range. However, the individual label predictions were considerably better with Transformer models.

Evaluation on training set showed that despite the small size of the data, the Transformer models could learn on Union and Majority datasets but not on Intersection dataset. It is possible that the intersection method could be used on a larger dataset that has also more balanced label distribution or dataset where the annotation agreement is higher.

Out of ALBERT, BERT and RoBERTa, none of the models achieved significantly better results within a dataset. The differences were mostly due to learning rate. For RoBERTa model, the results indicated that 5e-05 could slightly better in terms of label-wise evaluation. This could be due to RoBERTa's advantages over BERT, which have been also brought out in the work by Liu et al. [11]. For future work, it might be considered to use learning rates with smaller steps to see if the effect is caused by the learning rate.

The single samples given to RoBERTa models for symptom detection indicated that even though the models can learn on Majority dataset then on single samples quite a lot of models are not able to detect all the required symptoms. This is most likely due to the general lack of data, which is also impacted by the aggregation method.

8 Summary

Since using NLP has become popular in detecting depression symptoms from social media posts, it was decided to use NLP for symptom detection from a small dataset collected from Reddit.

The dataset consisted of 2002 Reddit posts that had been posted to r/depression page. The posts were annotated by non-professionals so that each post would have three annotators. The annotators could choose between ten labels that represented ten most common depression symptoms. The annotation analysis showed strong label imbalance and low annotation agreement. Four symptom's labels had significantly better label representation.

This work focused on analysing the quality of the annotated data and if this level of quality is enough to train a predictive model. The overall annotation agreement was fair and the agreement for individual labels ranged from slight to moderate.

Three aggregation methods were used to create datasets with different annotations. The methods were taking union of the annotations (Union dataset), using majority vote (Majority dataset) and taking intersection of the proposed annotations (Intersection dataset). These datasets were used to train three Transformers models: ALBERT, BERT and RoBERTa. Different seeds and learning rates were tested. The results did not show significant differences from using various seeds. Use of different learning rates indicated slight changes, mostly regarding individual label predictions while the averaged results for the models were in close vicinity and clear effect of learning rates was not observed. All three models showed good learning and generalization ability on Union and Majority dataset. Intersection dataset had too bad label representation for the models to learn and generalize.

Due to unreliability of the Union and Intersection datasets, Majority dataset was chosen as the 'main' dataset and based on the comparison of best models, RoBERTa was chosen as the model for further testing. RoBERTa models trained on Majority dataset were used for examining individual symptom detection. The results showed that the models trained with learning rate 5e-05 were able to detect symptoms better than the models trained with learning rate 3e-05. Additionally, individual samples from train and test sets were used to evaluate the symptom detection. It was seen that even with the best represented labels in the dataset, the models struggled to detect corresponding symptoms from the posts.

In conclusion, for a smaller dataset with fair annotation agreement, using majority vote for aggregating gives the most reliable results and the data can be used to train predictive models. The most promising results were given by RoBERTa model.

References

[1] JT Wolohan, Misato Hiraga, Atreyee Mukherjee, Zeeshan Ali Sayyed, and Matthew Millard. Detecting linguistic traces of depression in topic-restricted text: Attending to self-stigmatized depression with nlp. In *Proceedings of the first international workshop on language cognition and computational models*, pages 11–21, 2018.

[2] Damian F Santomauro, Ana M Mantilla Herrera, Jamileh Shadid, Peng Zheng, Charlie Ashbaugh, David M Pigott, Cristiana Abbafati, Christopher Adolph, Joanne O Amlag, Aleksandr Y Aravkin, et al. Global prevalence and burden of depressive and anxiety disorders in 204 countries and territories in 2020 due to the covid-19 pandemic. *The Lancet*, 398(10312):1700–1712, 2021.

[3] Ronald C Kessler and Evelyn J Bromet. The epidemiology of depression across cultures. *Annual review of public health*, 34:119–138, 2013.

[4] Tuka Al Hanai, Mohammad M Ghassemi, and James R Glass. Detecting depression with audio/text sequence modeling of interviews. In *Interspeech*, pages 1716–1720, 2018.

[5] Le Yang, Hichem Sahli, Xiaohan Xia, Ercheng Pei, Meshia Cédric Oveneke, and Dongmei Jiang. Hybrid depression classification and estimation from audio video and text information. In *Proceedings of the 7th annual workshop on audio/visual emotion challenge*, pages 45–51, 2017.

[6] World Health Organization et al. *The ICD-10 classification of mental and behavioural disorders: clinical descriptions and diagnostic guidelines*. World Health Organization, 1992.

[7] Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49, 2017.

[8] Glen Coppersmith, Craig Harman, and Mark Dredze. Measuring post traumatic stress disorder in twitter. In *Eighth international AAAI conference on weblogs and social media*, 2014.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020*

25th International Conference on Pattern Recognition (ICPR), pages 5482–5487. IEEE, 2021.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[12] Matthew Matero, Albert Hung, and H Andrew Schwartz. Understanding roberta's mood: The role of contextual-embeddings as user-representations for depression prediction. *arXiv preprint arXiv:2112.13795*, 2021.

[13] David E Losada, Fabio Crestani, and Javier Parapar. Overview of erisk at clef 2020: Early risk prediction on the internet (extended overview). *CLEF (Working Notes)*, 2020.

[14] Ana-Sabina Uban and Paolo Rosso. Deep learning architectures and strategies for early detection of self-harm and depression level prediction. In *CEUR Workshop Proceedings*, volume 2696, pages 1–12. Sun SITE Central Europe, 2020.

[15] Shweta Yadav, Jainish Chauhan, Joy Prakash Sain, Krishnaprasad Thirunarayan, Amit Sheth, and Jeremiah Schumm. Identifying depressive symptoms from tweets: Figurative language enabled multitask learning framework. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 696–709, 2020.

[16] Christian Karmen, Robert C Hsiung, and Thomas Wetter. Screening internet forum participants for depression symptoms by assembling and enhancing multiple nlp methods. *Computer methods and programs in biomedicine*, 120(1):27–36, 2015.

[17] H Andrew Schwartz, Johannes Eichstaedt, Margaret Kern, Gregory Park, Maarten Sap, David Stillwell, Michal Kosinski, and Lyle Ungar. Towards assessing changes in degree of depression through facebook. In *Proceedings of the workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality*, pages 118–125, 2014.

[18] Daniel M Low, Laurie Rumker, Tanya Talkar, John Torous, Guillermo Cecchi, and Satrajit S Ghosh. Natural language processing reveals vulnerable mental health support groups and heightened health anxiety on reddit during covid-19: Observational study. *Journal of medical Internet research*, 22(10):e22635, 2020.

[19] Jesse Read and Fernando Perez-Cruz. Deep learning for multi-label classification. *arXiv preprint arXiv:1502.05988*, 2014.

[20] Multi-class vs multi-label classification. https://medium.com/analytics-vidhya/multi-label-classification-using-fastai-a-shallow-dive-into-fastai-data-block-api-54ea57b2c78b.

[21] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[22] Armand Joulin, Edouard Grave, and Piotr Bojanowski Tomas Mikolov. Bag of tricks for efficient text classification. *EACL 2017*, pages 427–431, 2017.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[24] Dan Jurafsky and James H. Martin. *Speech and Language Processing (3rd ed. draft)*. https://web.stanford.edu/ jurafsky/slp3/, 2021.

[25] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[26] Rajasekar Venkatesan and Meng Joo Er. Multi-label classification method based on extreme learning machines. In *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 619–624. IEEE, 2014.

[27] Xi-Zhu Wu and Zhi-Hua Zhou. A unified view of multi-label performance measures. In *International Conference on Machine Learning*, pages 3780–3788. PMLR, 2017.

[28] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[29] University of Tartu. Ut rocket, 2018. `https://doi.org/10.23673/PH6N-0144`.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Kaire Koljal**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Predicting Depression Symptoms Based on Reddit Posts,
   *(*title of thesis)

   supervised by Kairit Sirts.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Kaire Koljal
17/05/2022