Identifying Trends Among Depressed Users of Reddit

Adrian Thinnyun
Georgia Institute of Technology
athinnyun3@gatech.edu

Jongseok Han
Georgia Institute of Technology
jhan405@gatech.edu

Abstract

In this project we attempt to identify trends across posts made by depressed and non-depressed users on the social media platform Reddit and seek to train a classifier that can discriminate both types of posts. We trained seven models on a total of 1293 posts from depressed users and 548 posts from non-depressed users, achieving a total of 93.23% accuracy and an F1 score of 0.95 with a BERT-based classifier. We ran into difficulties during our experiments on Topic Modeling, which returned poor and relatively uninterpretable results; however we found greater success with creating WordClouds of the most common unigrams and bigrams found in depressed and non-depressed posts and especially from analyzing the most predictive n-grams of both types of posts by extracting the coefficients of a linear classifier trained with n-gram features, ultimately forming a coherent picture in the differences in the language and attitudes of depressed and non-depressed people.

1 Introduction

Depression is a common and serious mental health disorder. The World Health Organization (WHO) estimated that globally 332 million people suffer from depression, leading to a serious plunge in self-perception and heightened risk of suicide. Thus, in order to tackle the problem using Natural Language Processing (NLP) techniques, we propose two tasks: detecting whether or not a user is depressed based on the contents of their online posts, and identifying common keywords/topics discussed in posts made by depressed users.

2 Related Work

Depression detection have been highlighted with advent of social media and language model. Compared to the other natural language processing area, depression detection studies focused on the feature analyses to interpret the observations in medical perspective instead of prediction performance. De Choudhury et al. (2013) examined the diverse signals from language, emoticon, style, and user engagement using Twitter feeds and found that depressed individual showed lower social activity and high self-attentional focus. Based on the linguistic cues and user posting patterns, Kumar et al. (2019) defined features set which consists of lexicon word, tweet timing and frequency, sentiments and proposed ensemble vote classifier that combined the result of Naïve Bayes, Gradient Boosting, and Random Forest. Similarly, Tadesse et al. (2019) did a comprehensive study to examine the relationship between depression and a user's language usage. Then, using the LDA features, they captured the depression-related topics and suggested the optimal feature combinations. However, despite their numerous feature extraction methods, their approaches were quite limited in word level linguistic model. Since depression attitudes can be exposed via nuance or implied expressions in the context, there have been a strong demand for inclusive depression detectors and the deep neural network has emerged as a promising approach. In order to capture the semantic representation of a sentence, Tang et al. (2015) proposed target-specific Long Short-term Memory (LSTM) model that utilizes the connection between target word and context word in composing the sentence representation. Also, Amanat et al. (2022) built LSTM-RNN and significantly reduced the false-positive prediction error. Additionally, given the enlarged online forum, Rao et al. (2020) developed the novel hierarchical depression detector using Convolution Neural Network (CNN) which achieved scalability without losing accuracy. Although diverse methods have been introduced, there is still room for improvement in both features extractions and prediction model. Thus, through this project, we will explore the attributes unrevealed in previous works and build deep learning-based architecture which

incorporates the attributes for early depression diagnostic system.

3 Methodology

3.1 Data Preprocessing

For our experiments, we used the Reddit dataset created by Pirina and Çöltekin (2018) that consists of depression and non-depression posts with multiple sentences. The depression posts are mainly collected from the depression support forums and subreddits. Also, for data augmentation, the authors included the posts written by users who posted depression related topics during the same time period. At the same time, they randomly sampled the non-depression posts from subreddits related to a family and friends. For data preprocessing, we followed the steps taken in Tadesse et al. (2019). First, we extended the contracted words in given sentences and removed stop words, URLs, and punctuations. Then, we reduced the words to root form to easily group the words in the feature extraction stage. Although we adopted their data preprocessing steps to reproduce the baseline which focused on word-level features, we consider the additional sentence-level data preprocessing to keep the semantic information hard to captured in word-level features as a next step.

3.2 Text Classification

Our primary task was to build a classifier that could effectively discriminate between posts made by depressed and non-depressed users of Reddit. Towards this goal, we tested a variety of approaches including traditional machine learning methods as well as transformer-based/deep learning methods. Specifically, we tested a multi-layer perceptron classifier, a linear support vector classifier, an LSTM-based classifier (Hochreiter and Schmidhuber, 1997), and a BERT-based classifier (Devlin et al., 2018).

3.2.1 Multi-Layer Perceptron Classifier

We trained a multi-layer perceptron (MLP) classifier in the interest of reproducing the results found in (Tadesse et al., 2019) as a baseline. In particular, the best combination of features and classification model found by the authors was an MLP trained on a combination of LIWC, LDA, and bigram features, which they reported as achieving an accuracy of 91%, an F1 of 0.93, a precision of 0.90, and a recall of 0.92 on the Reddit dataset. Since the LIWC lexicon is proprietary, we were unable to acquire it due to a lack of financial resources. An attempt was made to include LDA features but due to complications explained further in Section 4.3, they were not included in the MLP model we trained. Ultimately, we used a combination of unigram and bigram features instead to produce our experimental results, which were used as input to an MLP classifier implemented by scikit-learn (Pedregosa et al., 2011).

3.2.2 Linear Support Vector Classifier

For the purposes of producing the results seen in Section 3.4, three linear support vector classifiers implemented by scikit-learn (Pedregosa et al., 2011) were trained on the Text Classification task. The three classifiers were trained on unigram, bigram, and trigram features respectively. Although not primarily intended for Text Classification, we include them in this section for the sake of completeness.

3.2.3 LSTM Classifier

In order to capture the semantic features implied in word sequences, we built two different LSTM-based classifiers (Hochreiter and Schmidhuber, 1997). As a base model, we considered only word sequences that appeared in each post. For the input, sentence length was constrained as maximum 50, and words in all sentences are tokenized. In the model, all input sentences are fed to the embedding layer, Bidirectional LSTM, and a single linear layer which returns scalar value for binary classification. Additionally, since we already verified the effectiveness of n-gram features in MLP baseline experiments, we tried to incorporate the unigram and bigram features in the LSTM model. In this case, we pre-determined the n-gram features which are weighted by TF-IDF transformation and concatenated this n-gram feature to the Bidirectional LSTM output. For this modification, we added two-dense layers with ReLU activation function for n-gram feature and one final layer for concatenation.

3.2.4 BERT Classifier

The last model we trained on the Text Classification task was a BERT-based classifier (Devlin et al., 2018). In particular, we leveraged the pre-trained `bert-base-uncased` implementation by Huggingface[1] and fine-tuned it on our dataset. Rather

[1] https://huggingface.co/bert-base-uncased

than the n-gram features used in many of the previous models, in order to make our inputs compatible with BERT, we followed the following preprocessing steps:

1. Add tokens marking the beginning and ending of each sentence ([CLS], [SEP])
2. Pad sentences to the same length
3. Create an attention mask representing which tokens should be considered in the model's learned representation (e.g. padded tokens were masked out).

We trained for 2 epochs using a batch size of 16 and an AdamW optimizer with a learning rate of 5e-5.

3.3 Topic Modeling

As mentioned in Section 3.2.1, we attempted to reproduce the results found in (Tadesse et al., 2019) as a baseline for our Topic Modeling task. The authors used an LDA model to generate 70 topics from the Reddit dataset; however they did not include any metrics regarding the quality of these topics such as topic diversity and coherence, only results based on their predictive power in classification. Thus, we trained an LDA model implemented by the package OCTIS[2] to generate 70 topics from the Reddit dataset and then evaluated the results based on normalized pointwise mutual information (NPMI, (Bouma, 2009)) to measure topic coherence and the percentage of unique words for all topics (Dieng et al., 2020) to measure topic diversity. We also trained the BERTopic model (Grootendorst, 2022) in a similar fashion and evaluated it according to the same metrics. Notably, BERTopic does not accept a fixed number of topics as a parameter, but rather generates as many topics as it sees fit, which can then be reduced to a manually-specified number.

3.4 Predictive N-grams

Due to the results found in the Topic Modeling task during our midway report, we decided to try a different approach for discovering keywords and phrases characteristic of depressed and non-depressed posts. As mentioned in Section 3.2.2, we trained three linear support vector classifiers on unigram, bigram, and trigram features of our dataset, then extracted the coefficients from these models in order to find the most predictive n-grams for depressed and non-depressed posts.

[2]https://github.com/MIND-Lab/OCTIS

4 Experimental Results

4.1 Data Analysis

Interestingly, the most common bigram in posts by both depressed users and non-depressed users was "feel like", appearing 555 times in the 1293 depressed posts and 152 times in the 548 non-depressed posts. We present the WordClouds generated of the most common unigrams and bigrams across depressed and non-depressed posts in Figures 1, 2, 3, and 4.

We believe the difference in most common bigrams between depressed and non-depressed posts are the most illustrative, with examples such as "anyone else" and "want die" being more common in depressed posts while bigrams like "best friend" and "would like" being more common in non-depressed posts.

4.2 Text Classification

Each of the classifiers we trained was evaluated based on accuracy, precision, recall, and F1 score. The results of our experiments are detailed in Table 1.

4.2.1 Multi-Layer Perceptron Classifier

The MLP classifier trained on unigram and bigram features achieved an accuracy of 90.24%, an F1 of 0.93, a precision of 0.93, and a recall of 0.93. These results are very close to the accuracy of 91%, F1 of 0.93, precision of 0.90, and recall of 0.92 reported by (Tadesse et al., 2019), despite using a different combination of features. Hence, we conclude that these results serve as an adequate baseline to compare the rest of our experiments again.

4.2.2 Linear Support Vector Classifier

The linear support vector classifier trained on unigram features performed surprisingly well, exceeding the performance of the multi-layer perceptron classifier on all metrics. The performance of the classifier decreased with larger n-grams; however all three classifiers performed sufficiently well to consider analyzing their coefficients in Section 4.4 a valid approach.

4.2.3 LSTM Classifier

Based on sentences and n-gram features, two different Bidirectional LSTM classifiers were trained

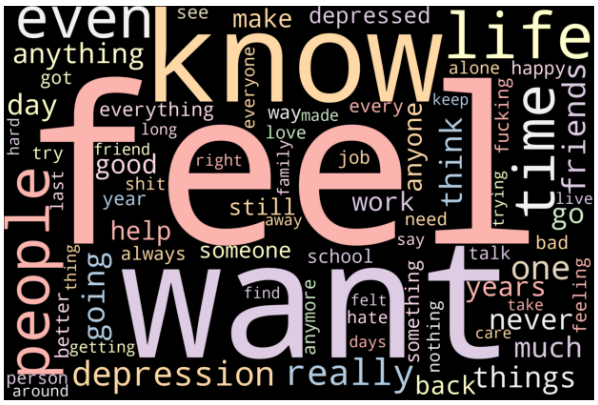

Figure 1: Most Common Unigrams in Posts By Depressed Users

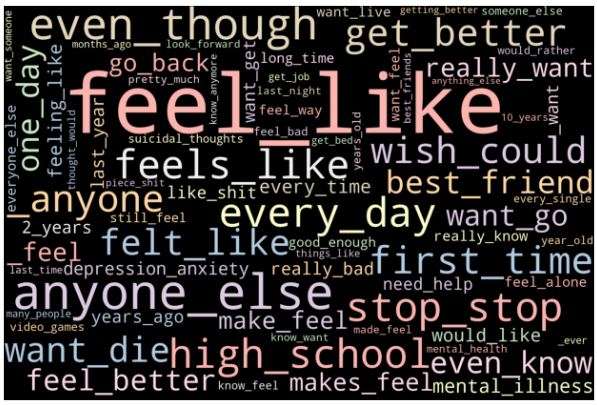

Figure 3: Most Common Bigrams in Posts By Depressed Users

Figure 2: Most Common Unigrams in Posts By Non-Depressed Users

Figure 4: Most Common Bigrams in Posts By Non-Depressed Users

and achieved an accuracy of 90.3% and 94.6% respectively. From this, we observed that n-gram features weighted by TF-IDF improved the overall model performance. Also, we found that explicit n-gram features as well as word sequence can help to detect the depressed users' posts.

4.2.4 BERT Classifier

The BERT classifier exhibited strong performance on the dataset, beating the MLP classifier and all three SVC models in accuracy and F1 score. However, it attained slightly lower accuracy and F1 than the LSTM that used unigram and bigram features.

4.3 Topic Modeling

The LDA model we trained achieved an npmi of -0.0065 and a diversity of 0.086, which implies extremely poor performance. A manual analysis of the topics generated by the model reflect this poor performance, as many of the topics feature extremely similar sets of keywords (e.g. ['friend', 'know', 'feel', 'like'], ['friend', 'want', 'feel', 'get']). The BERTopic model we trained faced the opposite problem: rather than generating a large number of very similar topics, it simply generated one topic and treated all documents that didn't fall into that topic as outliers. As a result, it achieved an npmi of 0.0163 and a diversity of 1.0, though clearly this model could not be described as a well-performing model.

4.4 Predictive N-grams

We extracted the coefficients from the three linear support vector classifiers described in Section 3.2.2 and translated them to the corresponding n-grams. The largest positive coefficients are the n-grams most predictive of depressed posts, and the largest negative coefficients are the n-grams most predictive of non-depressed posts. The top 10 most predictive n-grams for depressed posts are listed in Table 2, and the top 10 most predictive n-grams for non-depressed posts are listed in Table 3.

The results from this experiment are much more interpretable and logical than the results of our

failed Topic Modeling experiments. N-grams such as "depression", "anyone else", "want die", and "ever feel like" are clearly indicative of and characteristic of the language and attitudes of depressed people, whereas n-grams such as "friend", "family members", and "let us call" clearly reflect the language of non-depressed people. There are some abnormalities such as "talked behind back" being predictive of non-depressed posts, though it may be necessary to examine these cases in their original context to be able to make clear judgments.

It is worth noting that we experimented with filtering out words such as "depression" and "depressed" from our input data before training our text classifiers and found that they were able to achieve similar performance compared to the unfiltered data, so while these words are certainly predictive of depressed users, our models are not reliant on them.

5 Conclusion and Future Work

Our Text Classification experiments proved generally successful, despite the inability to directly reproduce the exact setup followed in (Tadesse et al., 2019). Considering that we were able to achieve similar performance even with a different combination of features, we suspect that the results we achieved represent the extent of an MLP's ability to discriminate between depressed and non-depressed posts in this dataset, implying a more complex model is needed to achieve higher performance. To this end, the BERT classifier as well as the LSTM classifier augmented with n-gram features both managed to exceed the performance of the baseline, which fell in-line with our expectations.

The Topic Modeling experiments were not very successful, showing both poor quantitative and qualitative performance. The WordClouds presented in Section 4.1 were comparatively more interpretable, but were based on absolute frequency rather than predictive power, leading to cases like the bigram "feel_like" and "even_though" appearing prominently in both depressed and non-depressed posts. In contrast, the results of our Predictive N-gram experiments yielded highly interpretable results backed by well-performing models as shown in Table 1. We believe that these results best represent the differences in the kind of language used by depressed and non-depressed users.

In terms of future work, there are several extensions of this work we could consider. Due to time constraints, we were unable to implement the CNN-based model mentioned in our midway report, so this represents a potential extension for a future project. Additionally, while we found that the LSTM model augmented with n-gram features performed better than the base LSTM model, we did not have the chance to test if the same applied to the BERT model. We also believe additional efforts into topic modeling may yield better results than the results we found during our preliminary experiments. Finally, it may be worth considering alternative datasets that might comprise a broader/more diverse audience and be more representative of the experiences of the depressed population as a whole.

6 Contributions

Adrian performed all of the experiments that contributed to the text classification section (except for the two LSTM models), the topic modeling section, and the predictive n-grams section. Jongseok preprocessed the dataset, generated the WordClouds, and implemented/evaluated the two LSTM models. Both members contributed to the report and presentation.

References

Amna Amanat, Muhammad Rizwan, Abdul Rehman Javed, Maha Abdelhaq, Raed Alsaqour, Sharnil Pandya, and Mueen Uddin. 2022. Deep learning for depression detection from textual data. *Electronics*, 11(5):676.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40.

Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. In *Seventh international AAAI conference on weblogs and social media*.

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

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Akshi Kumar, Aditi Sharma, and Anshika Arora. 2019. Anxious depression prediction in real-time social data. *arXiv preprint arXiv:1903.10222*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Inna Pirina and Çağrı Çöltekin. 2018. Identifying depression on reddit: The effect of training data. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*, pages 9–12.

Guozheng Rao, Yue Zhang, Li Zhang, Qing Cong, and Zhiyong Feng. 2020. Mgl-cnn: a hierarchical posts representations model for identifying depressed individuals in online forums. *IEEE Access*, 8:32395–32403.

Michael M Tadesse, Hongfei Lin, Bo Xu, and Liang Yang. 2019. Detection of depression-related posts in reddit social media forum. *IEEE Access*, 7:44883–44893.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Effective lstms for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*.

Table 1: Text Classification Results

Model	Accuracy	Precision	Recall	F1
MLP (w/ unigram and bigram)	90.24	0.93	0.93	0.93
SVC (w/ unigram)	91.11	0.93	0.95	0.94
SVC (w/ bigram)	82.43	0.83	0.94	0.88
SVC (w/ trigram)	71.58	0.73	0.95	0.82
LSTM (base)	90.27	0.93	0.94	0.93
LSTM (w/ unigram and bigram)	94.59	0.95	0.98	0.96
BERT (base)	93.23	0.97	0.93	0.95

Table 2: N-Grams Most Predictive Of Depressed Posts

Unigram	Bigram	Trigram
depression	feel like	diagnosed depression anxiety
depressed	anyone else	anyone else feel
life	want die	ever feel like
alone	wish could	put foster care
feel	feel better	even feel like
anyone	help get	still feel like
birthday	feel alone	look mirror see
people	piece shit	things get better
hate	want get	making feel like
ever	makes want	remember last time

Table 3: N-Grams Most Predictive Of Non-Depressed Posts

Unigram	Bigram	Trigram
friend	best friend	let us call
friendship	years ago	make new friends
family	family members	one best friends
said	please help	never long term
friends	another friend	well months ago
mom	months ago	talked behind back
phone	right thing	like nothing happened
father	hold back	sorry long post
us	mom dad	one else talk
dad	thanks advance	would like someone