Identifying Depression on Social Media

Kali Cornn
Department of Statistics
Stanford University
Stanford, CA 94305
kcornn@stanford.edu

Abstract

Millennials often turn to social media forums for mental health support. Looking at comments and posts on such platforms can give insight into how people self-disclose and discuss mental health issues such as depression. Using a dataset of scraped Reddit comments, this project aims to classify depression in comments. Focusing on the setting of social media, this project explores methods of machine learning and neural network architectures for identifying depression in digitally shared text entries. This project developed machine learning (logistic regression, support vector machines), a BERT-based model, and neural networks with and without word embeddings (CNN) for this classification task. It was found that the CNN model without word embeddings performed the best, with approximately 92.5% accuracy after 4 epochs, followed by the BERT-based model with 85.7% accuracy.

1 Introduction

Knowledge about disclosure of mental health issues on social media is limited. This project focuses on the website Reddit, which seems to fill a gap between other social media platforms such as Twitter or Facebook - which are often associated with permanent online identities - and health forums. Reddit is a unique platform in that users can choose to create "throwaway" accounts that are not associated with their main account in order to make posts or comments disclosing sensitive information.

1.1 Problem Definition

Depression has been shown to affect the language of individuals [1]. This project aims to use natural language processing, machine learning techniques, and neural network architectures to build, tune, and evaluate models that classify Reddit text comments as "depressed" or "non-depressed."

2 Background and Related Work

Natural language processing and machine learning have been used to perform sentiment analysis of social media posts. For example, previous work has involved building models predicting depression using the tweets of depressed Twitter users. It also has been found that Facebook status updates can reveal symptoms of major depressive episodes.

One paper aimed to address early detection of depression utilizing Reddit comments [1]. In this paper, models utilizing such pre-trained word vectors as GloVe and fastText were used in order to create simple CNN models consisting of a single layer.

Another paper utilized a deeper CNN on a wider variety of texts, such as Yelp reviews (polarity and full), Amazon reviews (polarity and full), and responses on Yahoo! answers. It was found that the

deeper CNN worked well on user-generated data, such as Amazon reviews, and less well on text that is more carefully curated, such as responses on Yahoo! answers [2].

3 Dataset

We use a custom dataset created by scraping two subreddits: /r/depression and /r/AskReddit. The dataset after pre-processing resulted in a total of 239,521 Reddit comments, consisting of 111,962 depressed comments and 127,559 non-depressed comments.

The data were concatenated, randomly shuffled, and split in a 80%-20%-20% ratio for training (143,712 comments), dev (47,904 comments), and testing (47,905 comments) sets, respectively. The final dataframe consisted of one column of text comments and another column of labels for the corresponding comments. Each comment was labeled with 1 or 0 for "depressed" or "non-depressed", respectively, depending on the subreddit it was posted on.

3.1 Dataset Collection

"Top" and "hot" comments were scraped from /r/depression and /r/AskReddit using the Python Reddit API Wrapper (PRAW) to create a custom dataset. All comments from /r/depression were considered "depressed" and those /r/AskReddit were "non-depressed."

We follow the reasoning of Wolohan et al. in choosing these two subreddits. /r/depression is described as "a supportive space for anyone struggling with depression" and many threads are of question-and-answer format. /r/AskReddit is similar in that it is also of question-and-answer format and can be a useful control for the mental health community. We also choose /r/AskReddit since the types of questions asked on this subreddit can provoke answers of different types of emotions other than sad/depressed [3].

An initial scraping of comments yielded 121,355 depressed comments and 178,277 non-depressed comments, all of which were not labeled.

An example of a comment from /r/depression and is therefore considered to be "depressed" is:

"I just really want to give up on life right now."

An example of a comment from /r/AskReddit and is therefore considered to be "non-depressed" is:

"Everyone now thinking about the last time they pooped. Nicely played."

3.2 Dataset Pre-processing

Some Reddit comments were deleted or removed, but were still considered a comment when extracted using PRAW. The dataset was thus filtered to only include actual comments.

Remaining comments were converted to lowercase letters, and further cleaned to remove such irrelevant text as subreddit and user mentions, and extra whitespace tokens. Comments were not trimmed for length: the inclusion of short words is particularly relevant to depression identification because recent work suggests depression is correlated with use of first-person pronouns [4].

4 Models

4.1 Baseline Models

We created two models to serve as a baseline for this project. Comments were converted to a matrix of **tf-idf** features to serve as inputs for these models. This was done by using **scikit-learn's TfidfVectorizer** with parameters **norm = L2** and **min_df = 2** (e.g. the vectorizer ignores terms that appear in less than 2 comments).

First, we developed a logistic regression model. For this model, we used Python's **scikit-learn's LogisticRegression** with the inverse of regularization strength set to $C = 1$.

Next, we built a support vector machine (SVM) model. For this model, we used Python's **scikit-learn's LinearSVC** using a linear kernel and the penalty parameter of the error term set to $C = 0.25$.

4.2 BERT-Based Model

We developed a Bidirectional Encoder Representations from Transformers (BERT)-based model, which is a new language representation model as described in [5]. As the name suggests, it was designed to pre-train deep bidirectional representations that can be fine-tuned with an additional output layer. For this project, this output layer - a pooled output - was used for the binary classification of the comments. From the many pre-trained models available, we chose the English-language uncased (all lowercase before tokenization) model of BERT, as case information is not particularly important to the task of social media comment classification.

For this model, we used a dropout probability of 0.2, learning rate of 2e-5, batch size of 32, and 3 epochs.

4.3 Character-Based CNN Without Embeddings

The next model we developed was a character-level convolutional neural network (CNN) inspired by [2]. While CNNs are most often used in computer vision applications, they are also common in NLP tasks.

Here, we take a character-based approach rather than a word-based approach since writing patterns common in social media comments, such as uncommon words and slang, misspellings, and emojis, can be learnt by looking at characters. The dataset was preprocessed similar to the baseline models.

The input text was one-hot encoded into a 2D matrix as described in the paper, utilizing a custom alphabet consisting of letters, digits, and certain special characters. Therefore, any character in the text that was not in the alphabet was an all-zero vector in this matrix.

This model contains 6 convolutional layers utilizing rectified linear units (ReLU) activation followed by 3 fully-connected layers. Some layers also included one-dimensional max-pooling with a kernel size of 3. Two dropout modules are also used between the fully connected layers for regularization purposes. Weights were initallized using a Gaussian distribution, and with the same parameters as in the paper (mean = 0, standard deviation = 0.02).

An overview of the architecture of the model is depicted in the below diagram, from [2].

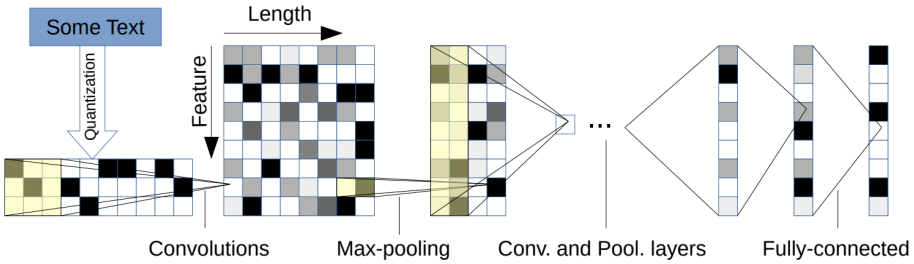

Figure 1: Representation of CNN model

In this model, we used a minibatch size of 128, a dropout probability of 0.5, and Adam optimization with a learning rate of 0.001. The model was also trained for 5 epochs.

4.4 Character-Based CNN With Pre-Trained Word Embeddings

Pre-trained neural word embeddings, which use distributed representations of words, are useful in representing interactions between words and are popular for use in text classification tasks. Therefore, another character-level CNN model was built similar to the one previously described, this time utilizing Stanford's 50-dimensional GloVe embedding weights. These word vectors were trained on a corpus of 6 billion words extracted from Wikipedia and the Gigaword 5 news corpus, and has a vocabulary of 400 thousand.

In this model, the input text embedded into a GloVe matrix, which was in turn fed into the neural network described in the previous section.

5 Experiments and Results

Evaluation of models involved use of the following metrics.

$$\textbf{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
$$\textbf{precision} = \frac{TP}{TP + FP}$$
$$\textbf{recall} = \frac{TP}{TP + FN}$$
$$\textbf{F1-Score} = 2 \cdot \left(\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \right)$$

In the above formulas, note that TP = true positive, TN = true negative, FP = false positive, and FN = false negative.

Training and dev accuracy and loss were also measured while developing, training, and tuning all models.

Model	Acc	Precision	Recall	F1 Score
LogReg	0.848	0.856	0.813	0.834
SVM	0.850	0.854	0.821	0.837
BERT	0.857	0.849	0.844	0.847
Char-CNN (no embed)	0.925	0.915	0.956	0.935
Char-CNN (w/ embed)	0.776	0.799	0.702	0.747

Table 1: Quantitative results of all models.

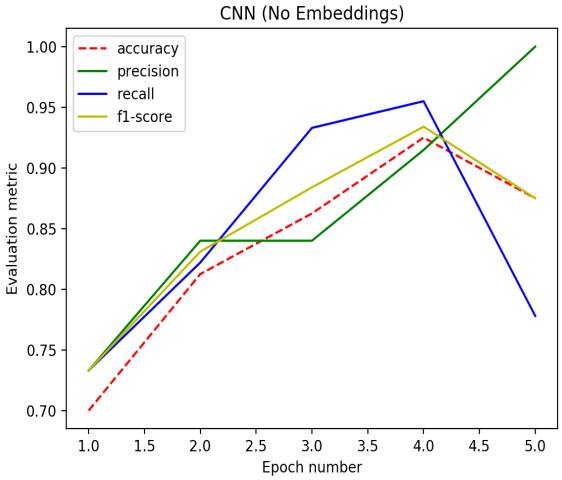

Figure 2: Evaluation metrics by epoch for CNN model with no embeddings.

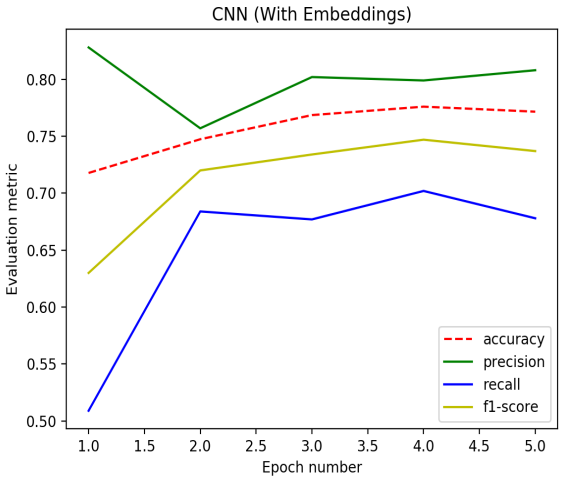

Figure 3: Evaluation metrics by epoch for CNN model with embeddings.

6 Analysis, Conclusion and Future Work

6.1 Analysis and Conclusion

In theory, the baseline accuracy for this classification task is 50%; namely, there is a 50-50 chance of classifying a comment as "depressed" or "non-depressed," as our training set was roughly evenly split between "depressed" comments and "non-depressed" comments. All models developed in this project performed better than this theoretical baseline.

With an accuracy of 85.7%, our BERT model performed slightly better than did the baseline models. An advantage of using BERT is that it is bidirectional; namely, this model is able to access context from "past" and "future" directions.

Our CNN model without embeddings performed the best, with an accuracy of 92.5% after just 4 epochs. Utilizing one-dimensional convolutional layers was likely helpful in classifying social media text since such layers only consider the most important features. In contrast, a RNN model can consider all words in a sentence, not all of which may be relevant, particularly in a social media context. So, 1D convolutional layers can help in reducing noise, which can be useful when classifying longer comments.

The CNN model with embeddings performed worst out of all models, but evaluation metrics in general consistently increased over time (as seen in Figure 3), unlike those of the other CNN model, the evaluation metrics of which were more variable.

An advantage of using a character-based CNN approach, as opposed to a word-based CNN approach, is that social media text can be very unique to the user and need not be formal: e.g. users can use emojis and repeat words and letters how they please. Using the word embeddings was likely a disadvantage as the GloVe word embeddings are not very representative of social media texts: for example, writings on Wikipedia tend to be formal with better grammar than is the average social media post, where there may be little to no editing of comments or posts.

A notable disadvantage of the CNN models, particularly the model with embeddings, is the significant increase in training time compared to the other models due to their depth and complexity. In the case of CNN with word embeddings, the training time is high as the input into the neural network is very large.

6.2 Issues and Possible Improvements

One issue with our custom dataset is that we immediately classified /r/depression comments as "depressed" and /r/AskReddit comments as "non-depressed," as was done in similar research. However,

this may not always be the case: it is certainly possible that a comment in /r/depression could be more lighthearted and therefore not be completely representative of depression. For example, one of the comments misclassified by the CNN model without embeddings to be "non-depressed" where it was actually "depressed" was

"this is really insightful. i think youre right."

which has a more positive tone than does the typical comment in /r/depression that may be seeking advice on mental illness. To improve this dataset, we could have manually labeled each comment through human evaluation (such as by psychologists or by finding confirmation that users have depression, e.g. a comment in their post history explicitly stating this) rather than utilizing automatic classification based on a general reading of each subreddits' comments. While this dataset would be much smaller than the one used for this project, it not only would give a better representation of depression on social media, but also would be more applicable for broader analysis of mental health and social media.

6.3 Future Work

We could improve our current CNN models by training for a longer time (e.g. running for more epochs). The accuracy went down by roughly 5% between epochs 4 and 5, so it would be interesting to see trends over more time. We could also use a different optimizer: in this project, we used Adam optimization with a fixed learning rate of 0.001, but another option is to use stochastic gradient descent and tune the learning rate by decreasing it over time.

While the focus of this project was to develop and improve CNN models, given their success in other works for similar text classification tasks, it would also be good for us to look deeper into other neural network models, such as RNNs and GRUs.

We could also experiment with simpler CNN models, as the one used in this project was fairly complex. Another avenue worth exploring could be building another complex model such as a character-level CNN-LSTM (long short-term memory) encoder-decoder model. This model has been effective in learning Tweet embeddings, and could potentially be expanded to other social media texts such as Reddit [6], which encodes inputs by using convolutional layers to extract features from characters and then passing these features through a LSTM layer, then decodes by using two LSTM layers to predict the next character at each time step.

6.4 Additional information

My mentor for this project was Pratyaksh Sharma. This project is shared between CS224N and CS229A (Applied Machine Learning).

References

[1] Marcel Trotzek, Sven Koitka, and Christoph M. Friedrich. "Utilizing Neural Networks and Linguistic Metadata for Early Detection of Depression Indications in Text Sequences". In: *CoRR* abs/1804.07000 (2018).

[2] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level Convolutional Networks for Text Classification". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 649–657. URL: http://dl.acm.org/citation.cfm?id=2969239.2969312.

[3] JT Wolohan. "Detecting Linguistic Traces of Depression in Topic-Restricted Text : Attending to Self-Stigmatized Depression with NLP". In: 2018.

[4] Johannes Zimmermann et al. "First-person Pronoun Use in Spoken Language as a Predictor of Future Depressive Symptoms: Preliminary Evidence from a Clinical Sample of Depressed Patients". In: *Clinical Psychology & Psychotherapy* 24.2 (2017), pp. 384–391. DOI: 10.1002/cpp.2006. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpp.2006. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpp.2006.

[5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018).

[6] Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. “Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder”. In: *SIGIR*. 2016.