A Deep Learning Approach to Classify Users' Mental States from Social Media Posts.

Nathanael L. Yoewono Jonathan A. Prasetiyo Shawn E. Samudra
Department of Mathematics and Statistics
University of Melbourne

October 25, 2020

Abstract

Depression is a major contributor to the overall global burden of diseases that affects people of all ages and gender. It is a serious issue as depression is a leading cause of suicide. In this paper, we utilized social media posts to train deep learning classifiers to detecting users' mental states, which is normal, depressed, or suicidal. The result shows that Bidirectional-LSTM(BiLSTM) with GloVe embedding layer has the highest performance with accuracy and AUC of 79.3% and 0.83 respectively, and it is an improvement over the baseline Support Vector Machines(SVM) model with TF-IDF word vectorizer.

1 Introduction

Suicide is one of the leading causes of death especially in today's youth, according to the WHO, it ranks third in deaths among 15-19 year-olds globally [2]. Furthermore, they also state that mental health disorders, particularly depression and substance abuse are associated with 90% of all suicide cases[2].

Concurrently, with the growing connectivity through the internet, more people are sharing their stories through social media posts[1]. In May of 2020, Reddit.com has 1.5 billion users[3]. Listening to their stories can be a way to help people suffering from depression.

On the other hand, Natural Language Processing(NLP) techniques are rarely implemented for this problem[4]. Having a good classifier can assist mental health organizations to identify and help people affected. Hence, a depression classifier can be useful as a means to help tackle the global problem of depression. However, text is a form of unstructured data; thus, feature engineering steps are needed. Nonetheless, due to the lack of domain knowledge, it might be best for the model to learn the features itself. Ergo, a deep learning approach is used to detect these depressing posts.

2 Dataset

For the datasets, we have chosen to scrape from r/depression, r/SuicideWatch, r/happy, and r/CasualConversation subreddits. These subreddits consist of posts with topics according to their title. Each subreddit has rules that are regulated by community moderators [5]. We acknowledge that not all users posting on each subreddit may be depressed, suicidal, neutral, or happy. However, due to the scarcity of labelled data, we assumed that they are labelled according to their respective subreddit.

2.1 r/depression

Posts in the depression subreddit only allow text data which are stories of depressed people that are both self-diagnosed and clinically diagnosed[5]. This subreddit allows people to share their experiences and support one another. The subreddit has strict guidelines that do not allow content that is irrelevant to the subject.

2.2 r/SuicideWatch

Similar to the r/depression subreddit, the r/SuicideWatch is also a place to share stories and experiences[6]. This subreddit is created to support users that are having suicidal thoughts and only allow posts that relate to this topic.

2.3 r/CasualConversation

The r/CasualConversation subreddit is a space for daily life conversations. They discourage serious sensitive and agenda-driven topics. Conversation topics can range from food, drinks, books, etc.

2.4 r/happy

r/happy subreddit is a space where users share happy stories and experiences. They strictly only allow straightforward happy posts without any double meaning or stories that may seem happy but is sad or depressing.

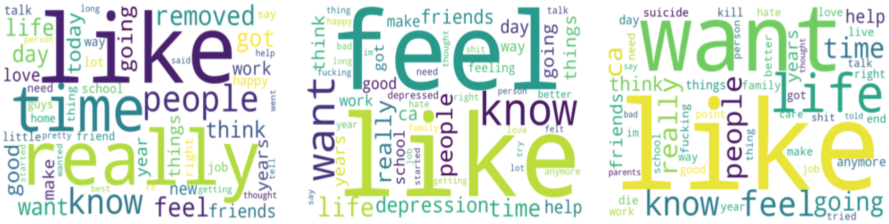
3 Preprocessing

There are around 1.9 million posts in total for the subreddit posts. The distribution of each subreddit is as follows: 49.11% Depressed, 23.18% SuicideWatch, and 27.71% CasualConversation and happy. These posts were all taken from 2012-2020 with 100 max posts in each day.

The preprocessing consists of several steps. Firstly, all missing posts were replaced with empty strings, and the title and text of each post were concatenated together into a single text column. Secondly, all emoticons in the subreddit posts were translated [8], as this may add to the sentiment context of the posts. Thirdly, each word in the posts was lower-cased and tokenized using NLTK's tokenizer [9]. Next, all stop words were removed using our customized stop words list. This customized list preserves all pronouns, as past research has shown that depressed posts tend to use more first-person pronouns [20]. Moreover, all non-alphabetical words were also removed, as they may not be an effective indicator to identify the users' mental state. Lastly, all spam posts were fixed by only preserving the unique words in the post. Spam posts are posts that continuously repeat the same phrases for a considerable amount of length.

Additional preprocessing was done for the depressed post by excluding posts with less than 15 words in length, as most of the short-length posts are quite ambiguous. This was not done in the other subreddits due to the lack of posts. Finally, the posts were labeled based on their subreddit origins. The label are as following: 0 = normal, 1 = depressed and 2 = suicidal. The dataset was then split into 4:1 train test split, where the train data will be used for both train and development. This split was done using a stratified split to preserve the class distribution [21].

4 Exploratory Data Analysis



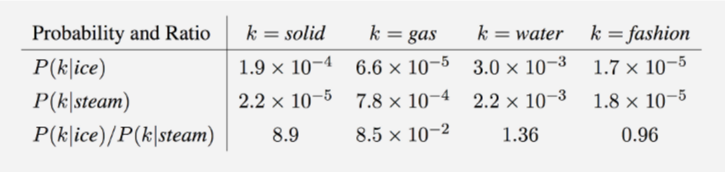
(a) Normal posts (b) Depressed posts (c) Suicidal pots

Figure 1: Word Cloud visualisation for the mental state labels

The figures above visualize the top 50 most frequent words that are used in each mental state class, which was done using a count vectorizer method [10]. The frequencies act as the weight for the word size in the word cloud. Generally, it can be seen that there are some words that are specifically used in certain classes, such as "depression" for depressed, and "suicide" or "kill" for suicidal. Nonetheless, it is interesting to see that "happy" and "love" are frequent in each mental state. Henceforth, relying only on word frequency may not be enough in identifying the state. As a result, learning the context of the words may be necessary. For instance, the word "love" in depressed and suicidal may refer to past memories of their loved ones.

5 Word Embedding Models

Word embedding is an unsupervised technique in NLP to map each unstructured text data into vectors of real numbers. For our paper, we used pre-trained Stanford GloVe 42b as it is one of the state of the art pre-trained word embedding models[11].



Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Figure 2: GloVe Overview[11]

GloVe word embedding method generates an embedding matrix for each feature by counting the word co-occurrences globally[11]. It tabulates the relationship of word by how frequent words co-occur in a given corpus as shown in figure 2. Generating the embedding matrix can be computationally expensive. Hence, we will rely on the pre-trained Stanford GloVe 42b[11].

6 Deep Learning Models

There are three different neural network models that will be trained. The first model is the static convolution neural networks(CNN), and the other two are recurrent neural network based models: bidirectional long short term networks(LSTM) and LSTM with attention layer.

6.1 Convolution Neural Network Static

This neural network uses convolution layers to extract a combination of important information from high-dimensional data, which is followed by a max-pooling to project them in a lower dimension. This combination of important information is the new feature for each input. In the context of text sentences, the convolution layer may learn the context of some phrases that might be a decent indicator of the users' mental state.

This experiment implemented Kim Yoon's static CNN model [18], as it is one of the state of the art CNN architecture model for text classifications [19]. The general architecture for the model can be seen in figure 3.

The 1D convolution filters will use a window size of 3, 4, and 5 with 100 filters each; thus, the total filters would be 300. Global max pooling is applied to each of these filters, which results in 300 new scalar features. These features were passed to the dropout layers with a scale of 0.2. Finally, the dropout layer

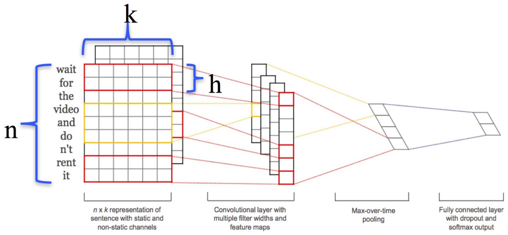

k
wait
for
the
video
and
do
n't
rent
it
n
h
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 3: Kim Yoon CNN architecture [22]

is connected to the output layer. The term static means the word embedding was not updated during training. We chose the static model due to lack of time and computing resources, as static will speed up the training process. Moreover, the GloVe embedding was highly trained on a large text; thus, it may not need further updating process.

6.2 LSTM with Attention Layer

The LSTM was first proposed by Hochreiter and Schmidhuber at 1997 in order to tackle the vanishing gradient problem[12]. It replaces the regular RNN cells with LSTM cells that have gates mechanisms. The gate mechanism allows each cell to decide whether or not to keep the previous states and also learn features from the current input data.

Attention was presented by Dzmitry Bahdanau, et al. in their paper "Neural Machine Translation by Jointly Learning to Align and Translate"[14]. It was first proposed as a solution to encoding and decoding long input sequences that had to be compressed into a fixed-length vector especially for sequences longer than the training corpus. Using the LSTM layer allows the model to identify parts of the input sequence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors.

For this model, we will use 64 units for the LSTM layer. The weight sequences are then given to the attention layer where it will compute the importance of each step of the sequence. In this model, we will not be updating our embedding layer like our CNN model.

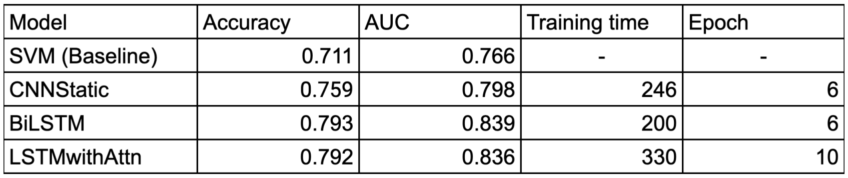
6.3 Bidirectional LSTM

Bidirectional LSTM learns from two networks of LSTMs in different directions, forward direction and reverse direction. Since networks are trained from both directions, it is able to capture information or patterns that a unidirectional LSTM would not have learned. In most cases, bidirectional model learns faster than the regular unidirectional model[13]. Furthermore, Bidirectional LSTM is one of the state of the art models for text classification.

Our BiLSTM model used glove for its embedding layer followed with a bidirectional LSTM layer with 64 units which will output. Finally, it is passed through a relu dense layer and a softmax dense layer. This model is also “static” much like the CNN model as it also does not update the embedding while training.

7 Results

The output layer for the deep learning models consists of three neurons, which represent the three mental state class. Moreover, softmax activation with a categorical cross-entropy loss function was used as the models are dealing with multi-class classification [15]. Adam optimizer was used as it is one of the states of the art optimizer used in most deep learning models [16]. Additionally, the input for the model was padded to 150 since the mode for the post length is around 107 words. The training was done in 10 epochs with an early stopping mechanism to avoid over-fitting.



Model	Accuracy	AUC	Training time	Epoch
SVM (Baseline)	0.711	0.766	-	-
CNNStatic	0.759	0.798	246	6
BiLSTM	0.793	0.839	200	6
LSTMwithAttn	0.792	0.836	330	10

Figure 4: Table of result on the test data

The results from the figure above indicate that BiLSTM has the highest performance, with accuracy and AUC score of 79,3% and 0.839 respectively. It also has the fastest training time with only 33.3 minutes/epoch. In addition, LSTM with the Attention layer was actually on par with BiLSTM, with 0.001 and 0.003 discrepancy in accuracy and AUC respectively. However, it can be seen that CNN’s performance was inferior compared to the LSTM models. This could be due to some information loss during the max-pooling process. CNN also requires more training time as there are more parameters to be trained compared to the LSTM models.

Furthermore, all of the deep learning models generate a higher result compared to the SVM classifier. This baseline SVM was trained using the tf-idf word vectorizer. We chose SVM as our baseline as it has been proven to be a capable model for classifying mental disorder by previous papers [17].

In addition, table 1 describes the performance of each model in classifying the users’ mental states. Interestingly, the CNN-Static model was superior in classifying the normal and depressed mental state compared to the other models, with 0.91 and 0.84 recall results. However, it performed poorly in detecting suicidal posts, with only 0.44 recall. Despite having a lower performance for both normal and depressing posts, LSTM-based models had a more stable performance for each of the mental states. The precision result for each class in each model was satisfactory. This may suggest that the model was not making any bias classifications towards certain mental states. This notion can be further supported by the AUC result for each model, where the lowest AUC is achieved by CNN with 0.798. A 0.5 AUC score means that the model is indifferent to its true positive and false positive performance. Hence, AUC greater than 0.798 indicates that the model has a higher proportion in detecting a true-positive compared to the false-positive classification.

Model	Mental State	Precision	Recall
CNN-Static	0	0.82	0.91
	1	0.73	0.84
	2	0.74	0.44
LSTM with Attention	0	0.87	0.90
	1	0.79	0.80
	2	0.70	0.64
BiLSTM	0	0.88	0.89
	1	0.80	0.79
	2	0.68	0.68

Table 1: Precision and Recall

8 Discussion

8.1 Model Discussions

Our results indicate that deep learning models perform better with a significant margin when compared to the SVM baseline. However, this may be because of our feature engineering steps. Due to the model limitations, SVM is unable to uncover hidden patterns like a deep learning model. Thus, it requires a more rigorous feature engineering process to help improve performance.

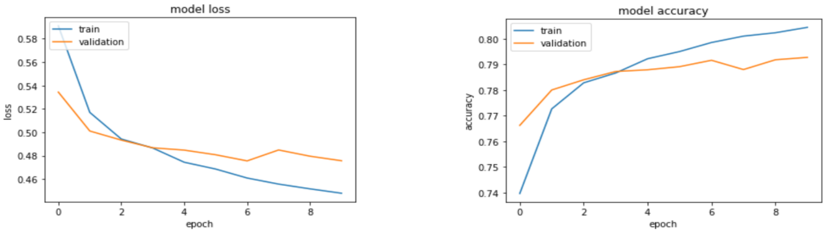


Figure 5: Model loss and accuracy on each epoch.

When comparing our three model results, it is clear that BiLSTM generated the highest valuation output based on the given number of parameters and epochs. It is higher than our CNN model and baseline, but is on par with our LSTM with the attention layer model. We can conclude that our sequential models work better for our classification problem. However, from figure 5, we can see that the LSTM with the attention layer's performance metrics has not plateaued. Hence, there may be a possibility that it will perform better than the BiLSTM model given more time.

8.2 Constraints and Limitations

Our main constraint for this paper is that labeled datasets for this particular topic are relatively scarce because of its sensitivity. As a result, we had to gather our data from subreddits and make several assumptions. We assumed that all posts belonging to a subreddit should be labeled accordingly, where in reality there are posts that are misplaced especially between r/depressed and r/suicidewatch. Another limitation is the number of resources that were available to us and were not able to add more layers or train for more epochs on some of our deep learning models as a result.

8.3 Future Improvements

For future work, we would like to try other deep learning models. Intuitively, BiLSTM with the Attention layer may improve our performance, as both of our LSTM models performed well. Implementing different word embeddings might also improve the model's accuracy, for instance, Google BERT. Moreover, tuning hyper-parameters may maximize each of our model's performance further. Lastly, we would also like to get more data and test our model on other social media platforms, such as Twitter and Facebook.

9 Conclusion

In conclusion, all of our deep learning models produced significantly higher performance compared to our baseline SVM. They all showed exceptional performance in differentiating between neutral and non-neutral posts. When comparing between the deep learning models, CNN is better at classifying neutral and depressing posts but is lacking when predicting suicidal posts. Although the BiLSTM did not perform as well in classifying neutral and depressing posts, it made up for it when classifying suicidal posts and had the overall best performance among all models with an accuracy of 79.3% and an AUC score of 0.836.

References

[1] Luxton, D., June, J. and Fairall, J., 2012. Social Media and Suicide: A Public Health Perspective. American Journal of Public Health, 102(S2), pp.S195-S200.

[2] Who.int. 2020. Depression. [online] Available at: <https://www.who.int/news-room/fact-sheets/detail/depression> [Accessed 26 September 2020].

[3] Statista. 2020. Reddit Users: Unique Monthly Visits 2019 | Statista. [online] Available at: <https://www.statista.com/statistics/443332/reddit-monthly-visitors/> [Accessed 25 October 2020].

[4] CALVO, R., MILNE, D., HUSSAIN, M. and CHRISTENSEN, H., 2017. Natural language processing in mental health applications using non-clinical texts. Natural Language Engineering, 23(5), pp.649-685.

[5] Reddit.com. 2020. [online] Available at: <://www.reddit.com/r/depression> [Accessed 25 October 2020].

[6] Reddit.com. 2020. Peer Support For Anyone Struggling With Suicidal Thoughts.. [online] Available at: <https://www.reddit.com/r/SuicideWatch/> [Accessed 25 October 2020].

[7] Husseini Orabi, A., Buddhitha, P., Husseini Orabi, M. and Inkpen, D., 2018. Deep Learning for Depression Detection of Twitter Users. Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic,.

[8] PyPI. 2020. Emoji. [online] Available at: <https://pypi.org/project/emoji/> [Accessed 25 October 2020].

[9] Nltk.org. 2020. Natural Language Toolkit — NLTK 3.5 Documentation. [online] Available at: <https://www.nltk.org/index.html> [Accessed 25 October 2020].

[10] Scikit-learn.org. 2020. Sklearn.Feature_Extraction.Text.Countvectorizer — Scikit-Learn 0.23.2 Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html> [Accessed 25 October 2020].

[11] Pennington, J., 2020. Glove: Global Vectors For Word Representation. [online] Nlp.stanford.edu. Available at: <https://nlp.stanford.edu/projects/glove/> [Accessed 25 October 2020].

[12] Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. Neural Computation, 9(8), pp.1735-1780.

[13] Schuster, M. and Paliwal, K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), pp.2673-2681.

[14] Brownlee, J., 2020. How Does Attention Work In Encoder-Decoder Recurrent Neural Networks. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/how-does-attention-work-in-encoder-decoder-recurrent-neural-networks> [Accessed 24 October 2020].

[15] Peterroelants.github.io. 2020. Softmax Classification With Cross-Entropy. [online] Available at: <https://peterroelants.github.io/posts/cross-entropy-softmax/> [Accessed 25 October 2020].

[16] Kingma, D. and Ba, J., 2017. Adam: A Method For Stochastic Optimization. [online] ICLR. Available at: <https://arxiv.org/pdf/1412.6980.pdf> [Accessed 25 October 2020].

[17] Park, G., Schwartz, H., Eichstaedt, J., Kern, M., Kosinski, M., Stillwell, D., Ungar, L. and Seligman, M., 2015. Automatic personality assessment through social media language. Journal of Personality and Social Psychology, 108(6), pp.934-952.

[18] Yoon, K., 2020. Convolutional Neural Networks For Sentence Classification. [online] Doha: Association for Computational Linguistics. Available at: <https://www.aclweb.org/anthology/D14-1181.pdf> [Accessed25October2020].

[19] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. and Gao, J., 2020. Deep Learning Based Text Classification: A Comprehensive Review. [online] Available at: <https://arxiv.org/pdf/2004.03705.pdf> [Accessed25October2020].

[20] Pennebaker, J., 2011. The Secret Life Of Pronouns. [online] New Scientist. Available at: <https://www.newscientist.com/article/dn20848-the-secret-life-of-pronouns/> [Accessed 25 October 2020].

[21] Scikit-learn.org. 2020. Sklearn.Model_Selection.Stratifiedshufflesplit — Scikit-Learn 0.23.2 Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html> [Accessed 25 October 2020].

[22] Jamiekang.github.io. 2020. Convolutional Neural Networks For Sentence Classification · Pull Requests To Tomorrow. [online] Available at: <https://jamiekang.github.io/2017/06/12/cnn-for-sentence-classification/> [Accessed 25 October 2020].

A Appendix

Link to our github code: https://github.com/ElmoSamudra/NLP-DepressionClassifier