Text-based depression detection

Evangelia Baou
University of Piraeus
Department of Digital Systems

Dimitris Konstantakopoulos
University of Piraeus
Department of Digital Systems

September 20, 2021

Abstract

Depression is a common worldwide mental disorder, which brings enormous challenges to the patient. Due to the huge increase of awareness of mental health well-being, the detection of mental illness itself is starting to become a major concern.Nowadays the problem of early depression detection is one of the most important in the field of psychology and the task of automatic depression detection from speech has gained popularity. Text-based depression detection is a field of great value in the age before the fourth industrial revolution and is not yet fully explored. Previous text-based depression detection is commonly based on large user-generated data. Sparse scenarios like clinical conversations are even less investigated. To predict patients' depression state, this study investigates recurrent neural networks model architectures with different embedders and proposes the best candidate for this binary classification task on the DAIC-WOZ dataset. These architectures engulfs pre-trained word embeddings of Glove and FastText embedders in conjunction with Bidirectional GRU layers and an architecture of multi-task neural network with two outputs using as extra output the PHQ8 metric due to its interdependence nature with the binary detection task.

Index Terms—*Neural networks, depression detection, text-embeddings, RNN.*

I. Introduction

DEPRESSION is a disease that affects millions of individuals around the world, whether they are aware of it or not. Automatic depression diagnosis that is both efficient and successful can be quite beneficial. At its core, classic depression detection is a binary classification issue, using classifiers drawn from traditional methods such as SVM[1], naive Bayes[2], decision tree[3], and neural networks such as long short term memory (LSTM)[4] and convolutional neural network (CNN) [5.] Text-based depression detection, in particular, has been extensively researched on user-generated data; for example, a task in the CLEF eRisk challenge tries to predict depression severity from data obtained online, such as questionnaire answers [6] and written social media texts [7]. Our work approaches the problem from a different perspective: how to detect depression in text sparse data scenarios.

Conversational data between an interviewee and a professional therapist is of particular relevance to us. It's worth noting that self-generated data affects a huge number of people and could result in vast amounts of data, whereas clinical talks for depression diagnosis can be limited. The dataset used in our study is part of a larger corpus, the Distress Analysis Interview Corpus (DAIC) that contains clinical interviews designed to support the diagnosis of psychological distress conditions such as anxiety, depression, and post-traumatic stress disorder.

Data collected include audio and video recordings and extensive questionnaire responses; this part of the corpus includes the Wizard-of-Oz interviews, conducted by an animated virtual interviewer called Ellie, controlled by a human interviewer in another room. Data has been transcribed and annotated for a variety of verbal and non-verbal features. This published dataset only includes 107 participants for the training part and 35 participants for the evaluation part; thus, the training scheme greatly differ from those in a large data setting.

Regarding the severity prediction, it can either be seen as a multi-class classification or a regression problem, usually associated with a psychological questionnaire score like Patient Health Questionnaire PHQ-8. In this work, we develop a multi-task neural network with two labels: a binary diagnosis of depressed/healthy and the patient's eight-item Patient Health Questionnaire score (PHQ-8) metric. However, this approach is not of great importance to us since the depression state and PHQ-8 score are correlated but one characteristic does not necessarily predict the other.

Our study, mainly, focuses on the binary depression detection task and investigating the usage of pretrained word embeddings to alleviate sparse-data depression detection problems. We analyze the dataset to understand the difficulties involved when modeling this task and a Bidirectional RNN with GRU architecture is developed.

II. Methodology

i. Data modelling

As mentioned earlier, the DAIC-WOZ dataset encompasses three major media: video, audio and transcribed text data. In this work, we only incorporate the text data for the purpose of neat real-world application. Three different modeling settings are commonly employed in text-based depression analysis [8]:

In **Context-free** modelling each response of the participant is used as an independent sample while no information about neither the question nor the time it was asked is provided. Due to the fact that predictions can be formed from single sentences, this approach has the advantage of being simple to implement in real-world applications.

Context-dependent modelling requires the use of question-answer pairs, where each sample consists of a question asked and its corresponding answer.

Sequence modelling only models the patients responses in succession, without knowledge of the particular question asked.

In our work the Sequence modelling is implemented. Our sequential modeling approach treats an entire paragraph spoken by a patient as a single sample.

ii. Data Preprocessing

Text preprocessing is traditionally an important step for natural language processing (NLP) tasks. It transforms text into a more digestible form so that deep learning algorithms can perform better.

The raw text was firstly preprocessed, where tailing blanks were removed and every letter was set to be lowercase. Furthermore, techniques such as removal of punctuations, stop words, most common and rare words were applied. Moreover, it was observed that some of the patients' answers were in informal language, so custom preprocessing were applied for replacing the informal speaking with a formal one. The last step of the preprocessing concerns the lemmatization (reduction of each word to its root word).

Regarding the removal of the most frequent and rare words, we firstly count the frequency of each word in the corpus and then the first two and the last thirty words of the ranking list are deleted. Prior to data augmentation, the preprocessed training set contains an overall 64261, the development set 22136 written words. The training set contains 107 while the development set 35 patients. Meta information such as <laughter> or <sigh> are possibly helpful to the model, thus were not removed.

iii. Data Augmentation

Data augmentation is used in our work due to the limited size of the training dataset. Different techniques were applied for augmenting our data but only one of them was finally kept as an option since in NLP field, it is hard to augmenting text due to high complexity of lan-

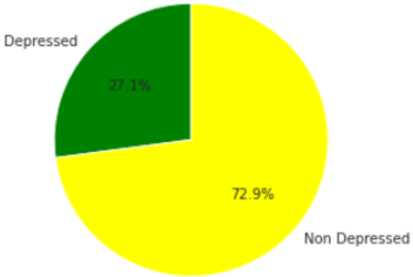


Figure 1: *Words' Percentage per Class (Training Data)*

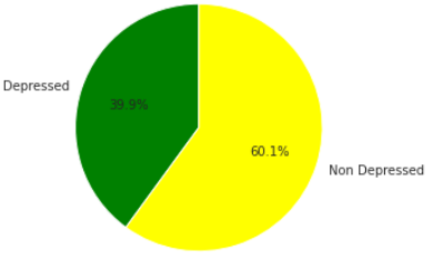


Figure 2: *Words' Percentage per Class (Validation Data)*

guage.

In particular, some of the most common techniques that are used are Random Insertion, Synonym Replacement, Random Swap etc. In our work, we end up with Synonym Replacement that randomly chooses n words from the sentence that are not stop words and replaces each of these words with one of its synonyms chosen at random. The number of n is set equal to five (n = 5) and the WordNet (part of NLTK Python module) lexical database is used to find the synonyms.

It is worth mentioning that the augmentation technique was not applied to a specific class but on the whole dataset, so after that the initial dataset is doubled from 107 training samples to 214 samples.

iv. Text embeddings

Usually when working with NLP tasks, it is necessary to use word vectors to represent the words presents in a vocabulary in a dense format. These representations are called word embeddings. Despite of being not a new idea, word embeddings became popular after Google releasing pretrained Word2Vec in 2013. Since that time, others Word Embeddings have emerged such as GloVe and the fastText. In previous depression detection research, context-free word embeddings are usually used, either trained from scratch or a simple pretrained word embedding. In this study we experiment with the pretrained fastText and GloVe.

FastText is a fast and effective method to learn word representations and perform text classification. The main objective of the fastText embeddings is to take into consideration the internal structure of words instead of learning word representations. This is remarkably useful for morphologically rich languages, so that the representations for different morphological forms of words would be learnt independently. FastText works by sliding a window over the input text and either learning the center word from the remaining context also known as continuous bag of words, or all the context words from the center one as skipgram. Learning can be viewed as a series of updates to a neural network with two layers of weights and three layers of neurons, in which the two outer layers each have one neuron for each word in the vocabulary, and the middle layer has as many neurons as there are dimensions in the embedding space.

This approach is very similar to Word2Vec. However, unlike Word2Vec, fastText might also learn vectors for sub-parts of words: so-called character n-grams. This ensures that for instance the words love, loved and beloved all have similar vector representations, even if they tend to show up in different contexts. This feature enhances learning on heavily inflected languages.

Apart from FastText embedder, GloVe embedder is also used in our experiments. The GloVe model is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. Populating this matrix requires a single pass through the entire corpus to collect the statistics. For instance, given a corpus having V words, the co-occurrence matrix X will be a V x V matrix, where the ith row and jth column of X, X_{ij} denotes how many times word i has co-occurred with word j. The main disadvantage of GloVe in compasiron to FastText is that GloVe fails to provide any vector representation for words that are not in the model dictionary.

III. Deep Neural Network Architectures

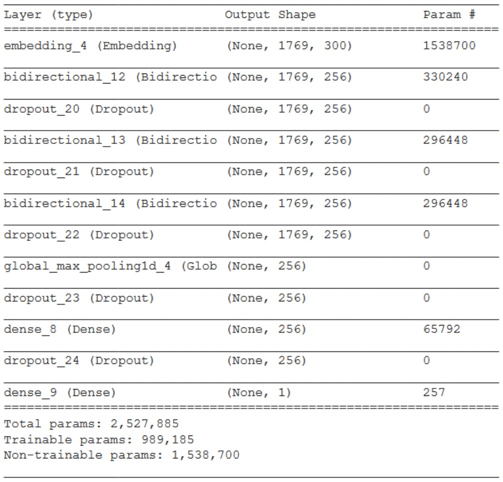
RNN-based models view text as a sequence of words, and are intended to capture word dependencies and text structures. However, vanilla RNN models do not perform well, and often underperform feed-forward neural networks. Among many variants to RNNs, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are the most popular architectures. In our experiments two variations of neural networks are used. The first one concerns a Bidirectional RNN with GRU architecture. As input to its embedding layer both GloVe and FastText embedders are used. The second variation is a multi-task neural network with two outputs (binary detection of depression as a classification task and the PHQ8 metric as regression task).

The multi-task neural network was designed to achieve better loss only to binary classification and the results of the regression task are not counted. Due to the co-dependence of the PHQ8 score with the binary classification of depression the two characteristics are correlated, but one cannot necessarily predict the other. Hence, both information sources can be effective in order to ascertain the patients' state. Multi-task architecture did not achieve encouraging results compared to the binary classification task as we expected and fastText embedding neural network achieved lower scores than its counterpart with the GloVe embedder. Thus our proposed model for this specific classification task is our first approach.

i. Architecture of BGRU model

Our proposed model was developed with Tensorflow Keras. It consists of an embedding layer with an embedding dimension of 300. In the case of pre-trained embedders, the embedding layer is set as not trainable while when the embedders are not used then it is set as trainable. Furthermore, three bidirectional GRU layers with 128 units each and a Tahn activation function are used. A globalMaxPooling layer without any mask, a dense 256 neuron layer, several dropout layers, some of them with relatively high values due to the sparse data scenario, and a sigmoid activation function for the binary classification task are parts of our model.

Training was done by running Adam optimizer for at most 10 epochs and a batch size of 15. The initial learning rate was set to be 0.0004 which was reduced by a factor of 10 if the cross-validation loss did not improve for at most 2 epochs. Finally, binary crossentropy is used as loss function. The model's summary is provided in Figure 4.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_4 (Embedding)      (None, 1769, 300)         1538700
_________________________________________________________________
bidirectional_12 (Bidirectio (None, 1769, 256)         330240
_________________________________________________________________
dropout_20 (Dropout)         (None, 1769, 256)         0
_________________________________________________________________
bidirectional_13 (Bidirectio (None, 1769, 256)         296448
_________________________________________________________________
dropout_21 (Dropout)         (None, 1769, 256)         0
_________________________________________________________________
bidirectional_14 (Bidirectio (None, 1769, 256)         296448
_________________________________________________________________
dropout_22 (Dropout)         (None, 1769, 256)         0
_________________________________________________________________
global_max_pooling1d_4 (Glob (None, 256)               0
_________________________________________________________________
dropout_23 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_8 (Dense)              (None, 256)               65792
_________________________________________________________________
dropout_24 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_9 (Dense)              (None, 1)                 257
=================================================================
Total params: 2,527,885
Trainable params: 989,185
Non-trainable params: 1,538,700
_________________________________________________________________
```

Figure 3: *BGRU model summary*

IV. Results

Due to the imbalance of the dataset, our main goal is to achieve a low loss score while accuracy of the model is of second priority to our experiments. Apart from loss, more evaluation metrics are chosen such as precision, recall and f1-score. Moreover, a stratified 5-Fold cross-validation is used in order to ensure the effectiveness of the model and especially for tackling overfitting and underfitting. The BGRU-fastText has achieved a loss of 0.51 while the BGRU-Glove has achieved a loss of 0.62 on the test dataset. Since the BGRU-fastText model performs better than the BGRU-Glove we present its classification report and confusion matrix in Tables 1 and 2 respectively, along with its ROC curve in Figure 4.

	Precision	Recall	F1-score
Class 1	0.70	0.58	0.64
Class 0	0.80	0.87	0.83
Macro avg	0.75	0.73	0.73
Weighted avg	0.77	0.77	0.77

Table 1: *Classification report of BGRU-fastText model.*

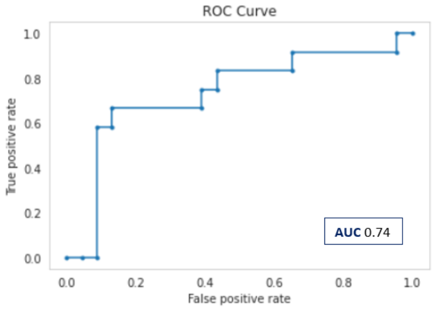


Figure 4: *ROC curve of BGRU-fastText model*

V. Conclusion

This work proposed the use of stacked RNN BGRU model in conjunction with word embeddings, namely fastText and GloVe. Moreover the same model is also trained without using the aforementioned embedders and these two approaches are compared. Our results indicate that using pretrained embedders on a large, unrelated dataset , can lead to encouraging results and lead to a faster training and a lower final training loss. It can be interpreted that the model could pick up more semantic signals from the pre-trained embeddings than it did from the training data through the embedding layer. Among fastText and GloVe, the first seems to have more superior results in our problem regarding the evaluation metric used. This lies on the fact that the way fastText works, out of vocabulary words can be handled while GloVe does not have this ability.

VI. Future Work

Conversational data can potentially reveal more information on the participant's linguistic ability and cognitive function,therefore can provide a different angle towards depression detection. In our future work, we aim to improve our two-output model in order to obtain better results not only in the binary classification problem but in the severity prediction as well. Moreover, since a plethora of embedders is available, it is important to experiment using some of them apart from fastText and GloVe.

References

[1] J.F. Cohn, T. S. Kruez, I. Matthews, Y. Yang, M. H. Nguyen, M. T. Padilla, F. Zhou, and F. De la Torre, "Detecting depression from facial actions and vocal prosody," in 2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops. IEEE, 2009, pp. 1–7

[2] M. Deshpande and V. Rao, "Depression detection using emotion artificial intelligence," in 2017 International Conference on Intelligent Sustainable Systems (ICISS). IEEE, 2017, pp. 858–862.

[3] L. Yang, D. Jiang, L. He, E. Pei, M. C. Oveneke, and H. Sahli, "Decision tree based depression classification from audio video and language information," in Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge, 2016, pp. 89–96

[4] M.-H. Su, C.-H. Wu, K.-Y. Huang, and Q.-B. Hong, "Lstm-based text emotion recognition using semantic and emotional word vectors," in 2018 First Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia). IEEE, 2018, pp. 1–6.

[5] A. H. Orabi, P. Buddhitha, M. H. Orabi, and D. Inkpen, "Deep learning for depression detection of twitter users," in Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic, 2018, pp. 88–97.

[6] Losada, David E. and Crestani, Fabio and Parapar, Javier, "erisk 2020: Self-harm and depression challenges," in European Conference on Information Retrieval. Springer, 2020, pp. 557–563.

[7] "erisk 2017: Clef lab on early risk prediction on the internet: experimental foundations," in International Conference of the Cross- Language Evaluation Forum for European Languages. Springer, 2017, pp. 346–360.

[8] T. Alhanai, M. Ghassemi, and J. Glass, "Detecting Depression with Audio/Text Sequence Modeling of Interviews," Tech. Rep. [Online].