2020 IEEE Region 10 Symposium (TENSYMP), 5-7 June 2020, Dhaka, Bangladesh

Early Depression Detection from Social Network Using Deep Learning Techniques

Faisal Muhammad Shah
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
faisal505@hotmail.com

Farzad Ahmed
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
farzadahmed6@gmail.com

Sajib Kumar Saha Joy
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
joyjft@gmail.com

Sifat Ahmed
Artificial Intelligence Engineer
Hiperdyne Corporation
Tokyo, Japan
sifat.ahmed@hiperdyne.com

Samir Sadek
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
samirsadek1009@gmail.com

Rimon Shil
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
rimonroy.cse@gmail.com

Md. Hasanul Kabir
Department of Computer Science and Engineering
Islamic University of Technology
Dhaka, Bangladesh
hasanul@iut-dhaka.edu

***Abstract*—Depression is a psychological disorder that affects over three hundred million humans worldwide. A person who is depressed suffers from anxiety in day-to-day life, which affects that person in the relationship with their family and friends, leading to different diseases and in the worst-case death by suicide. With the growth of the social network, most of the people share their emotion, their feelings, their thoughts in social media. If their depression can be detected early by analyzing their post, then by taking necessary steps, a person can be saved from depression-related diseases or in the best case he can be saved from committing suicide. In this research work, a hybrid model has been proposed that can detect depression by analyzing user's textual posts. Deep learning algorithms were trained using the training data and then performance has been evaluated on the test data of the dataset of reddit which was published for the pilot piece of work, Early Detection of Depression in CLEF eRisk 2017. In particular, Bidirectional Long Short Term Memory (BiLSTM) with different word embedding techniques and metadata features were proposed which gave good results.**

***Index Terms*—depression, social media, early detection, deep learning, BiLSTM, word embedding, metadata feature**

I. Introduction

The research, which was done by the World Health Organization (WHO) [1], states that there are three hundred million humans living with depression. In 2015 it was estimated that there are four percent of the earth's inhabitants who are depressed [1]. Among age of 15 to 29, suicide is the second main reason for dying. Depression is the top reason of committing suicide. Close to Eight Hundred thousand people die because of suicide every year.

Depression analysis is the process of identifying whether people are going through depression from their textual activity on social media. Identification of depression from social media has been framed as a classification problem which is in the domain of Natural Language Processing (NLP). In this work we study NLP approaches that can successfully extract information from textual data to enhance identification of depression. These NLP approaches perform different feature extraction to build document representations.

The main aim of our work is to detect depression early by analyzing the posts of Reddit users. We worked on the dataset which was published for the pilot piece of work Early Detection of Depression in CLEF eRisk 2017 [2]. We tried to implement deep learning based model to detect depression from Reddit users' posts. We evaluated several different combinations of word embedding techniques and metadata features with Bidirectional Long Short Term Memory (BiLSTM) using Latency-weighted F1 ($F_{latency}$) [3], Early Risk Detection Error (ERDE) [4] and F1 Score to measure the standard and speed of the model.

978-1-7281-7366-5/20/$31.00 ©2020 IEEE

II. Related works

There has been several research done on depression detection from text. In this section, we will see some of the works done from past years. Nguyen at al. used Live Journal to take the text to a higher dimension and collected important emoticon words in social media [5]. The features used were Emotion tags, Linguistic Inquiry and Word Count (LIWC) and affective feature. Traditional Machine Learning classifiers were used. Farig Sadeque, et al. had tried to detect early depression from user's post on Reddit [6]. The authors used depressive words and concept unique identifiers from the Unified Medical Language System as features. They used both deep learning and traditional machine learning for the prediction task. Tyshchenko et al categorized stop words and added Linguistic Inquiry and Word Count (LIWC) features to a pre-existing model and made a new model (Bag-of-Words (BoW)+Term Frequency–Inverse Document Frequency (TF-IDF)+ LIWC) [7]. The author used different combinations of features using Convolutional Neural Networks (CNN) as classifier. They got the best result by using the CNN+GloVe model. Farig Sadeque, et al. also had tried to solve many problems with ERDE metric [3]. The authors proposed a $F_{latency}$ metric which solves the problems. Then they evaluated this metric on several previously made model eRisk 2017 depression task and got a better result. They performed experiments on the data published for the pilot piece of work Early Detection of Depression in CLEF eRisk 2017. They used four feature sets such Words, Depressive Words, Depressive Embed and Unified Medical Language System. They used both Deep Learning and traditional machine learning techniques to classify data.

Many works had been implemented using machine learning techniques and deep learning techniques but no hybrid model using different word embeddings and LIWC was implemented to give better results of detecting depression early.

III. Dataset

We performed our experiments on the dataset of reddit which was published for the empirical work, Early Detection of Depression in CLEF eRisk 2017 [2]. The arrangers collected the maximum number of posts they could find for each user (maximum 2000 posts per user). Then the users were divided into two classes, depressed class and a control (non-depressed) class. For the depressed class the arrangers kept only those Reddit posters who clearly expressed that they were recognized with depression by a physician. For the control group, organizers collected Redditors who had no mention of depression in their posts and participated in depression forums or any other forums. The dataset contains 531,453 posts of 892 different users. These users were divided into test (401) and train data (486).

IV. Methodology

The workflow of our methodology is given in figure 1. The top part of the model architecture is all about feature selection and the bottom part of the model refers to the classification part.

A. Feature Extraction

We experimented with different feature sets. Then the combination of different feature sets were used to achieve a better result. The description of the different types of features that we used for our experiment is given below.

1) **TrainableEmbed Features**: We used trainable embedding layer to generate TrainableEmbed features.
2) **GloveEmbed Features**: We used Glove embedding technique [8] to get GloveEmbed features. We downloaded pre-trained Glove embedding model in which each word is represented with a vector of dimension 300. These word vectors were passed through a non-trainable embedding layer which essentially generates GloveEmbed features for each word in a given text.
3) **Word2VecEmbed Features**: Word2Vec embedding [9] technique was used to get Word2VecEmbed Features. We used gensim [10] library to build Word2Vec model. We trained the model with our train data. After that, we collected word vectors from the trained Word2Vec model. Here also each word is represented with a vector of dimension 300. The rest of the steps to get Word2VecEmbed features were same as GloveEmbed features.
4) **FastextEmbed Features**: Here Fastext embedding [11] technique was used. The procedure of getting FastextEmbed features is the same as Word2VecEmbed features.
5) **Metadata Features**: Meta Data is "statistics that provide facts about other data". For example, the count of the mentions of antidepressant drugs in user's posts is a numerical metadata feature. Moreover, depressed users tend to talk about themselves a lot so, they use phrases like 'I' a lot, talk about antidepressants. They use a lot of personal and possessive pronouns. This information was found out statistically by authors of [13] by finding out the most common words from depressive posts. These are called handcrafted metadata features. We used 40 Metadata features in our model. The first 9 features were count based which were occurrences of "I' in a sentence, Antidepressant Drugs, Diagnosis, Therapist, Depressive words, Past Tense Verbs in a sentence, Personal Pronoun, Possessive Pronoun, Length of Posts in user's posts. For Metadata features such as depression, therapist and diagnosis, we found all the similar words of the corresponding meta data using Word2Vec model and used them as metadata features. We normalized these features to convert the ranges of these features between 0 and 1. We had used minmax scaler for normalization. Additionally we used 31 relevant features from Linguistic Inquiry and Word Count (LIWC) [15] software. These features were selected from 10 categories of LIWC features such as Affect, Biological Process, Drives, Personal Concern, Function

Words, Summary Dimension, Social, Cognitive Process and Time Orientation.

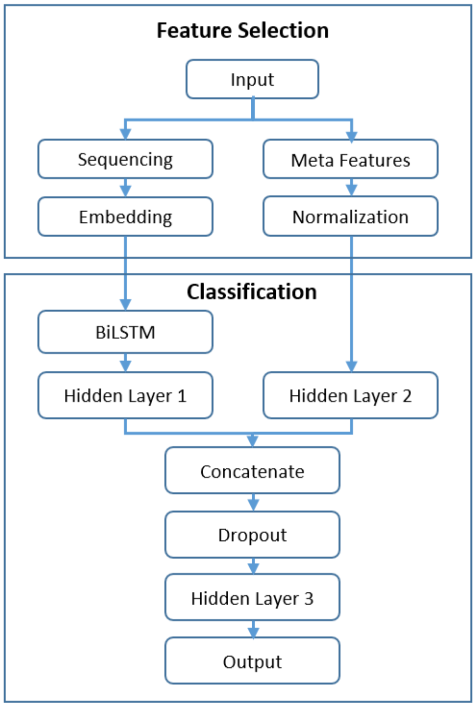

Fig. 1. Proposed Methodology

B. *Classification*

The Classification process started at the bottom part of the model. Embedded features were fed into Bidirectional Long Short Term Memory(BiLSTM) [12] layer of output dimension 600. Here BiLSTM was used because, at any time step, it had information about the past and future that helped the model to predict more accurately. The output of the BiLSTM layer was then fed into hidden layer 1 with an output dimension of 300. Normalized Metadata Features were fed into hidden layer 2 with an output dimension of 10. Rectified linear unit (ReLU) activation function was used in hidden layer 1 and hidden layer 2. After that, the vectors returned by these two hidden layers were concatenated. Thus the output dimension of the concatenation layer was 310. Then a dropout of 0.2 was added to avert overfitting. Finally hidden layer 3 was added whose output dimension was 1 and the activation function was sigmoid as we were doing binary classification. All the hidden layers were fully connected layers.

We set these dimensions by observing the result for a small amount of test data. The dimensions for which we got better result were kept. Keras API was used to build our model. We used adam optimizer [14] on batches of size 32 and the learning rate was 0.001. We did not change the other hyperparameter settings. We kept them as default values.

V. Training and Prediction

Our training phase had two stages. First, we built a dataset that contains 4656 posts of two categories, depressive posts and non depressive posts, equally distributed. To collect depressive posts, we used a sub portal of Reddit called depression. And for non depressive post we used textadventures portal. We manually observed posts that are more suitable for our task and assign them a class(i.e depressive or non-depressive), then train the model. In the CLEF eRisk 2017 dataset, class information for the posts of the users was not given. That is why, after training the model, we classified each post of train data as all the posts of a user may not be of the same class. Then in the second phase, we again trained our model with these classified data.

In case of predicting a user's state, we used the concept of risk window [3]. If the model makes a prediction that a post is depressive, a risk flag is raised. Then if the model continuously generates the identical (depressive) prediction for the consecutive n posts (size of the risk window is represented by n), or, if a person has less than n posts and the model continuously presages the identical prediction (depressive) for all the posts of that person, then the person is classified as a depressed. In the middle of the prediction process of the user, if a non depressive post arises, the risk flag is pulled down. If the model does not make n consecutive depressive prediction, then the user is eventually classified as a non depressive user.

VI. Result Analysis

Different measurement criteria are shown here to evaluate how well our model performed and also how quickly depressed users are predicted as depressed. Also, result comparison of different features is shown to find out which performed the best.

A. *Early Risk Detection Error (ERDE), Latency and Latency-weighted F1*

The commonly known classification metrics like F1-measure, Precision and Recall does not take time into account. To evaluate whether our model performed fast enough we used the metrics Early Risk Detection Error [4], Latency and Latency-weighted F1 [3].

Early Risk Detection Error (ERDE) measure is stated by:

$$ERDE_0(dc, ti) = \begin{cases} cs_{fp} & \text{if dc = p \& truth = n} \\ cs_{fn} & \text{if dc = n \& truth = p} \\ lcs_0(k) * cs_{tp} & \text{if dc = p \& truth = p} \\ 0 & \text{if dc = n \& truth = n} \end{cases} \tag{1}$$

where $lcs_o(ti)$ is the function that returns the cost and it is defined as

$$lcs_o(ti) = 1 - \frac{1}{1 + e^{ti-o}} \tag{2}$$

The delay is estimated by (ti), the number of texts observed before taking the binary decision (dc) which could be either '+' (p) or '-' (n). The o parameter acts as the "cutoff time" for making the decision. If the cutoff time is passed, the penalty starts rising exponentially. The authors of [4] set $cs_{fn} = cs_{tp} = 1$ and $cs_{fp} = 0.129$. The authors of

[3] proposed another metric to evaluate how fast the model performed. The latency is interpreted to be the median number of textual items that the model needs while predicting a depressed user. Latency is defined as

$$latency(User, s) = \underset{us \in User \wedge a(us)=+}{\text{median}} time(s, us) \quad (3)$$

'User' is the collection of users, a(us) is the actual class of the user, and Time(s,us) is the number of texts required by the model to classify user 'us'. Latency-weighted F1 ($F_{latency}$), is explained as the multiplication between the model's F1 score and the median of a set of penalties subtracted from 1. These penalties are measured by equation 4. After exactly 1 post if a prediction is made, the penalty is 0 and goes to 1 as the number of posts seen increases.

$$P_{latency}(us, s) = -1 + \frac{2}{1 + e^{-p(time(us,s)-1)}} \quad (4)$$

$$F_{latency}(User, s) = F_1(User, s)(1 - \underset{us \in User \wedge r(us)=+}{\text{median}} P_{latency}(us, s)) \quad (5)$$

$F_{latency}$ has a control variable p which controls how fast the penalty would rise. The authors of [3] set p to 0.0078 to give penalty 0.5 at median number of posts.

TABLE I
RESULT COMPARISON WITH DIFFERENT FEATURES USING RISK WINDOW SIZE 10

Features	F1 Score	$F_{Latency}$	$ERDE_5$	$ERDE_{50}$
TrainableEmbed	0.65	0.17	0.15	0.14
Word2VecEmbed	0.62	0.21	0.16	0.13
GloveEmbed	0.63	0.15	0.14	0.13
Meta Features	0.61	0.16	0.17	0.16
Word2VecEmbed+Meta	0.49	0.46	0.20	**0.10**
FastTextEmbed+Meta	0.69	0.34	0.17	0.13

TABLE II
RESULT COMPARISON WITH DIFFERENT FEATURES USING RISK WINDOW SIZE 15

Features	F1 Score	$F_{Latency}$	$ERDE_5$	$ERDE_{50}$
TrainableEmbed	0.57	0.21	0.17	0.18
Word2VecEmbed	0.53	0.20	0.14	0.12
GloveEmbed	0.70	0.18	0.14	0.13
Meta Features	0.62	0.15	0.13	0.16
Word2VecEmbed+Meta	0.71	**0.59**	0.15	0.12
FastTextEmbed+Meta	0.64	0.33	0.18	0.13

TABLE III
RESULT COMPARISON WITH DIFFERENT FEATURES USING RISK WINDOW SIZE 23

Features	F1 Score	$F_{Latency}$	$ERDE_5$	$ERDE_{50}$
TrainableEmbed	0.52	0.19	0.16	0.17
Word2VecEmbed	0.43	0.20	0.13	0.12
GloveEmbed	0.65	0.19	0.13	0.13
Meta Features	0.55	0.14	0.15	0.14
Word2VecEmbed+Meta	**0.81**	0.56	0.13	0.12
FastTextEmbed+Meta	0.77	0.26	0.15	0.13

As our dataset was not balanced, and we want to evaluate our model based on cost associated with false positive and false negative, we used F1 score, which depends on precision and recall, to evaluate our model. Also $F_{latency}$ is dependant on F1 score. Word2VecEmbed+Meta feature set gives the highest F1 Score of 0.81, with precision of 0.78 and recall of 0.86 at Risk Window 23. The same feature set gave the highest $F_{Latency}$ of 0.59 at Risk Window 15 and $ERDE_{50}$ of 0.10 at Risk Window 10.

VII. CONCLUSION

Depression detection from text is one of the toughest problems to solve. Yet we have tried several approaches to detect depression from the text. Our main focus was not only to correctly classify depressed users but also to reduce the amount of time to predict the state of the users. After experimenting with several approaches, we found out that, Word2VecEmbed+Meta features performed well. Our limitation is that though the users are correctly classified, it takes too long time to detect them as depressed. Further work can be done to solve this problem.

REFERENCES

[1] "Depression and Other Common Mental Disorders: Global Health Estimates." World Health Organization, 2017.

[2] D. Losada, F. Crestani, and J. Parapar,CLEF 2017 eRisk Overview: "Early Risk Predictionon the Internet: Experimental Foundations. In Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum." 2017.

[3] F. Sadeque, D. Xu, and S. Bethard,"Measuring Latency of Depression Detection in Social Media." WSDM'18, February 5-9, 2018, Marine Del Rey, CA, USA, 2018.

[4] D. E. Losada and F. Crestani, "A Test Collection for Research on Depression and Language Use". International Conference of the Cross-Language Evaluation Forum for European Languages. Springer International Publishing., 2016.

[5] T. Nguyen, D. Phung, S. V. B. Dao, and M. Berk, "Affective and content analysis of online depression communities." IEEE Trans. Affect. Comput., vol. 5, no. 3, pp. 217-226, 2014.

[6] Fariq Sadeque and Dongfang Xu and Steven Bethard, "UArizona at the CLEF eRisk 2017 Pilot Task: Linear and Recurrent Models for Early Depression Detection",CEUR Workshop Proc. Author manuscript; PMC 2017

[7] Y. Tyshchenko, "Depression and anxiety detection from blog posts data". 2018.

[8] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning."Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.

[9] Mikolov, Tomas, Kai Chen, Gregory S. Corrado, and Jeffrey A. Dean. "Computing numeric representations of words in a high-dimensional space." U.S. Patent 9,037,464, issued May 19, 2015.

[10] "Gensim - Deep learning with word2vec". "https://radimrehurek.com/gensim/models/word2vec.html". Accessed: 2019-09-28.

[11] "FastText". Facebook Research."https://research.fb.com/downloads/fasttext/". Accessed: 2019-12-25

[12] Schuster, Mike Paliwal, Kuldip. "Bidirectional recurrent neural networks.", Signal Processing, IEEE Transactions on. 45. 2673 - 2681. 10.1109/78.650093,1997.

[13] Trotzek, M., Koitka, S., Friedrich, and C.M., "Linguistic Metadata Augmented Classifiers at the CLEF 2017 Task for Early Detection of Depression". Working Notes Conference and Labs of the Evaluation Forum CLEF 2017, Dublin, Ireland, 2017.

[14] Diederik Kingma and Jimmy Ba. "Adam:A method for stochastic optimization.International Conference on Learning Representation", 2015

[15] Tausczik, Y.R., Pennebaker, J.W. "The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. Journal of Language and Social Psychology", Vol. 29(1), pp. 24–54. 2010