

Forensic Suicidal Inquiry of Depressed Individuals using LSTM and Convolutional Neural Networks

Chandrika Prasad, Anurag Hakeem, Rishabh Malhotra, Shahroz Ahmad Jan,
Urvash Pratap Singh, Jagdish S. Kallimani

Abstract—Social media sites such as Twitter, Facebook, Tumblr etc., are vastly popular among the general population. People post updates, tweets etc., and almost 75% of the times, these posts are a combination of emotions. The idea is to analyze suicidal-depression tendencies in adults with traumatizing experiences or socio-economic difficulties. This makes the overall analysis of sentiments especially extremely complex, which we aim to resolve here in this project by breaking down all the sentences into individual words, and along with emoticons and hashtags, converting each one of them into tokens, and then applying deep learning algorithms on the same, to accurately determine the sentiments of given messages. The objective of the project undertaken is to determine the suicidal-sentiment of various depressed individuals, and how likely is it that they are inclined to commit suicide on the basis of their tweets.

I. INTRODUCTION

We start out by defining the term ‘social media’, which is basically the collective of online communications channels dedicated to community-based input, interaction, content-sharing and collaboration. Social Media platforms like facebook, twitter etc provide users with the ability to communicate their sentiments and standpoints, current affairs and debates, all of which carry information, which represents a gargantuan broad data set. All of these major social media platforms offer technical environments through which data can be gathered for personal reasons, or any sort of analysis.

The data collected from twitter is very heterogeneous when it comes to its material, which accounts for the size of the dataset to be humongous - many thousands of tweets and posts per diem. Twitter has additionally allowed the geo-tagging of every tweet and message, related to the exact geographical coordinates of a particular location, which gives each tweet a sense of unique identification.

Revised Manuscript Received on April 12, 2019.

Chandrika Prasad, Department of Computer Science and Engineering M S Ramaiah Institute of Technology, Bangalore. (Email: chandrika@msrit.edu)

Anurag Hakeem, Department of Computer Science and Engineering M S Ramaiah Institute of Technology, Bangalore. (Email: anuraghakeem@gmail.com)

Rishabh Malhotra, Department of Computer Science and Engineering M S Ramaiah Institute of Technology, Bangalore. (Email: rishabh1302@gmail.com)

Shahroz Ahmad Jan, Department of Computer Science and Engineering M S Ramaiah Institute of Technology, Bangalore. (Email: shahrozjan67@gmail.com)

Urvash Pratap Singh, Department of Computer Science and Engineering M S Ramaiah Institute of Technology, Bangalore. (Email: urvashpsingh@gmail.com)

Dr. Jagdish S. Kallimani, Department of Computer Science and Engineering M S Ramaiah Institute of Technology, Bangalore. (Email: jagadish.k@msrit.edu)

This paper deals with the extraction of unstructured data from the twitter feed, by taking the tweets from a data stream, the data is subsequently cleaned and then trained using various neural network techniques to accurately determine the suicidal tendency of the depressed individual being analyzed.

II. LITERATURE SURVEY

In [1], they performed sentiment analysis on Twitter by taking a biased and noisy dataset, by proposing a 2-steps sentiment classification analysis, which first classifies messages as subjective and objective, further classifying the subjective tweets as positive or negative in the step that follows. Instead of applying the conventional means of supervised classification approaches on regular training data, they leveraged sources of noisy data as their training data set, which were provided by a few sentiment detection websites over twitter data. They exploited a variety of features for the problem of sentiment analysis - unigrams, bigrams, part of speech (POS) tags etc. They have mentioned 3 data sources for their technique, namely Twendz, Twitter Sentiment, and TweetFeel. After cleaning and transforming the dataset from these 3 sources, the subjectivity of each data item is first predicted, followed by polarity prediction on the subjective data. After obtaining their final data set, the usual supervised classifiers are applied, thus forming their model for sentiment analysis.

In [2], they perform sentiment analysis on twitter data by using 3 experimental models: unigram model, feature-based model and a tree kernel based model, using a 2 classification tasks, the first task being classification of sentiment as positive and negative classes, the second being a 3-way task of classifying sentiment as positive, negative and neutral classes. The training data source taken by them was the manual extraction of 11,875 tweets from a commercial source, which was basically a collection of tweets in several languages, which was then translated to English using the conventional methodology of using google translate. For the preprocessing of tweets, an acronym dictionary and an emoticon dictionary is used to convert acronyms into sentences and classifying emoticons as extremely positive, extremely negative, positive, negative, and neutral. After classifying the tweets based on their polarities, the above 3 models are tested, where the tree kernel based approach gives the best accuracy of 60.60%.



In [3], he makes use of the “NLPWin Natural Language Processing system”, to perform linguistic analysis of his data, where a phrase tree structure and a logical form forstringsis dispensed, where features such as POS trigrams, constituent specific length measures, constituent structure in the form of context free phrase structure patterns for each constituent in the form of a parse tree etc, can be extracted, alongside extracting a binary value for each of these features. Following this, a procedure known as feature reduction was applied, which is important for perfecting the performance of a linear classifier by reducing the size of the feature vector to a point such that it was less than or equal to the size of the training data set. Combinations of these features were taken as four categories: those with every feature, those without the linguistic features, surface features, and linguistic features only, and a linear classifier was then applied to each of these categories to then achieve the final result.

“In [4],three strategies were put forward to steady down the process of “dynamic routing” to attenuate the disturbance of loud sound capsules which incorporate all of the “background” data or have not been trained correctly. A number of experimental tests were conducted on six text classification benchmarks with capsule networks: MR (Pang and Lee, 2005), MR SST-2 (Socher et al., 2013), Subj (Pang and Lee, 2004), TREC (Li and Roth, 2002), CR (Hu and Liu, 2004), and AG’s news corpus (Conneau et al., 2017). 4 out of 6 datasets established competitive responses over the compared baseline methodsfor the capsule networks, which showed the benefits and effectiveness of capsule networks for this kind of text classification.

In [5], they used Representation Learning which linked raw data to better forms. The model they used was a single layer multiplicative LSTM 4096 units. It was trained for only one epoch on miniature batches of 128 sequences of length amounting to 256 for a sum of 1,000,000 weight updates in which the states were kept 0 at the initialization of each shard and persisted across updates to simulate full-back propagation and allow for the forward propagation of information outside of a given sub-sequence. Dataset used was the Amazon product review dataset which was further split into 1000 shards which contained equal amount of reviews and 1 shard each for validation and testing. The model achieved 91.80%accuracy compared to the state of the art of 90.21%.

In [6], they presented modelsthat particularly points towards transfer learning to other NLP tasks for encoding sentences into embedding vectors. They scrutinized and recorded the relationship between model complexity levels, resource utilization, the ease of getting of training data, and task performance. Comparisons were made between word level transfer learning via pre-trained word embeddingsand sentence embeddings. The results presented that word level transfer was outperformed by transfer learning using sentence embeddings. They had used eight benchmarks for their datasets:MR (2005), CR (2004), SUBJ (2004), MPQA (2005), TREC (2002), SST (2013), STS (2017) and WEAT (2017).

In [7], they developed a deep learning model for sentiment analysis of tweets. The most important result was amodel for which initialized the parameter weights for the

CNN, which is necessary for training an accurate model. The model used was that of unsupervised neural language which then trained word embeddings. At the penultimate stage, the pre-trained parameters of the network were used to initialize the model. They trained the latter on the official system evaluation campaign dataset of Twitter sentiment analysis which is a supervised training data made. End result was that of this model performing better than other models on the dataset.

In [8], Sentiment analysis of Twitter messages and short sentences is difficult because of the limited 140 characters of information which is usually contextual. Solving this problem in a timely manner requires methods which combine the limited text content with earlier known information. In this work, they proposed a new deep CNN that uses from “character to sentence” level information to perform sentiment analysis of short texts. The datasets used were the Stanford Sentiment Treebank, which essentially contains sentences from motion picture reviews, alongside the Stanford Twitter Sentiment Corpus, which contains only Twitter messages. This model achieves for the stanford sentiment treebank corpus 85.68% accuracy, and also fine-grained classification, with 48.32% accuracy. For the stanford twitter sentiment corpus, this approach achieved a sentiment accuracy of roughly 86.41%.

In [9], they worked on a few experiments involving the use of convolutional neural networks (CNN) which were trained on top of trained word vectors for sentence-level classification. He showed that a simple straightforward CNN with little hyperparameter tuning and static vectors can be simply remarkable when associated with various benchmarks. Also, learning task-specific vectors using fine-tuning can offer a massive rise in performance. He proposed a change in the architecture on order to use both task specific and static vectors. One of the dataset used for training the model was the Stanford Sentiment Treebank and the CNN models showed improvements in sentiment analysis and question classification. It showed that even a simple model with static vectors can perform outstandingly.

In [10], they brought into play the use of Convolved Neural Networks and Long Short Term Memories for conducting sentiment analysis on data extracted from twitter. He starts the paper by describing and comparing the architectures of CNNs and LSTMs used in his model. The input taken for this model are individual words that were tokenized from the original tweets. For performing the task of sentiment analysis, a total of 99490 tweets were used as the training data set, and alongside this human labelled data set, an additional 100 million unlabelled tweets were also exploited. Pre-processingis performed on the data appropriately by replacing the URLs, emoticons, and words with a repetition of letters with the <url> token, <smile>, <sadface>etc appropriate smiley token and the same word with an occurrence of not more than 2 repetitions of the same letter in a row, respectively. Unsupervised training methods are then applied to this preprocessed data with the help of Google’s Word2vec, Facebook’s FastText, and Stanford’s

GloVe. Distant training is then performed on this dataset to classify the data based on its polarity as positive, negative, or neutral. Finally supervised training is performed with the help of CNNs and LSTMs and the accuracies of all the various approaches were compared and ranked according to the years they were performed in.

In [11], the analysis was supported by extraction of two features, acoustic (phonetic features) and linguistic aspect of the speech. This model was trained on speeches which included interviews of patients and the transcription of the interview to create bigrams and unigrams

In [12], they had presented a Multi-Task Deep Neural Network (MT-DNN) through multiple natural language understanding (NLU) tasks for learning representations by joining multi-task learning with language model for pre-training. MTDNN by included a pre-trained bidirectional transformer language model, known as BERT (Devlin et al., 2018) and expanded the model proposed in Liu et al. (2015). MT-DNN helped to reshape by new tasks and domains by not only benefiting from a regularization effect that lead to more general rendition. MT-DNN showed results on ten NLU tasks. These tasks included SNLI, SciTail, and eight out of nine GLUE tasks, taking the GLUE benchmark upto 82.2% (1.8% absolute improvement). They also presented that the representations learned by MT-DNN led to domain adaptation with comparatively fewer in-domain labels than the pre-trained BERT with the help of SNLI and SciTail datasets.

In [13], the academicians performed sentiment analysis on a RNN-Capsule which is a capsule model made from a Recurrent Neural Network (RNN). For a given issue, a single capsule was built for every category of sentiment e.g., 'negative' and 'positive'. Each capsule had an attribute associated, a state, and three modules: representation module, probability module, and reconstruction module. Two benchmark datasets (i.e., Film and Movie Dataset from Stanford) and one proprietary dataset (i.e., the Medical Feedback dataset) were acquired and used. They helped in concluding that RNN-Capsule achieved very good performance on sentiment classification. More importantly, without using any linguistic knowledge, the words were presented out by the RNN-Capsule with sentiment likelihood explaining the capsules' attributes. The words well reflected the domain specificity of the dataset.

In [14], there are numerous neural network models which are used for sentiment analysis where the authors changed words to 2D Matrices for predicting sentiment distributions. The tactic learns vector area illustration for multi-word phrases, a bit like a word2vec implementation, and makes use of this algorithmic nature of sentences. Their approach differs from these previous works in a way that it follows a feed-forward neural network method rather than a algorithmic one. Moreover, it doesn't want any input regarding the grammar structure of the sentence. In their work they used a deep CNN which was better in sentiment analysis of short text due to the use of character-to-sentence level information. The network developed, CharSCNN had 2 convolutional layers which it used on varying sentences and words to find out relevant options. This model was used on movie reviews and tweets for measuring the performance.

III. PROPOSED SYSTEM

Model description:

Convolutional neural network (CNN) is a type of deep neural networks, which is a regularized version of multilayer perceptron. Multilayer perceptrons are completely connected networks where every neuron in layer one has a connection to all neurons in the layer two and so on. CNN uses hierarchical pattern in data to assemble more complex patterns using smaller and simpler patterns which is possible with the use of filters on the input data.

Long Short Term Memory (LSTM) networks are an extension of Recurrent Neural Network (RNN), which are designed to avoid the long-term dependency problem. They enable RNN's for remembering data for a long period of time where the memory is actually a gated cell which decides whether to delete or store data based on importance which is decided through weights.

The Convolutional Neural Network Long Short Term Memory Networks (CNN LSTM) belong to the class of models that has spatial and temporal depth. Moreover, CNN LSTM is flexible enough to be applied to various vision tasks which has both sequential inputs and outputs. CNN LSTM, as the name suggests combines CNN layers for extraction of features on input data with LSTMs to support prediction of sequences. CNN-LSTMs are used when their inputs have spatial features in their input such as the two dimensional structure, pixels in an image or the one dimensional structure of words in a sentence which is the case we are dealing with in our proposed paper. This figure shows the architecture of a CNN LSTM model where the Dense layer is used because the output of a LSTM model is not softmax.

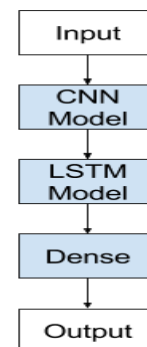


Figure 1: Basic Architecture of CNN LSTM

The model used in this study is designed using python's neural network library, Keras. Keras provides an API to design a sequential model, where different layers are added in the order of their execution. In our model, we use a combination of a 1d Convolutional Neural Network (CNN), along with a Long Short Term Memory (LSTM) unit, in a total of 7 layers.

REFERENCES

[1] Luciano Barbosa, JunlanFeng. Robust Sentiment Detection on Twitter from biased and noisy data; 2010

[2] ApoorvAgarwal, BoyiXie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau. Sentiment Analysis of Twitter Data; 2011

[3] Michael Gamon. Sentiment Classification on Customer Feedback Data: Noisy data, large feature vectors, and the role of linguistic analysis; 2004

[4] Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, Zhou Zhao. Investigating Capsule Networks with Dynamic Routing for Text Classification; 2018

[5] Alec Radford,RafalJozefowicz, IlyaSutskever. Learning to Generate Reviews and Discovering Sentiment; 2017

[6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope and Ray Kurzweil. Universal Sentence Encoder; 2018

[7] AliakseiSeveryn, Alessandro Moschitti, Twitter Sentiment Analysis with Deep Convolutional Neural Networks; 2015

[8] Cicero Nogueira dos Santo, MairaGatti, Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts; 2014

[9] Yoon Kim. Convolutional Neural Networks for Sentence Classification; 2014

[10] Mathieu Cliche. Twitter Sentiment Analysis with CNNs and LSTMs; 2017.

[11] John P. Pestian.Ph.d, Michael Sorter.MD, Brian Connolly.Ph.d, Kevin BretonnelCohen.Ph.d, Cheryl MccullumSmith.MD,Ph.d, Jeffrey T. Gee.MD, Louis-Philippe Morency.Ph.d : A Machine Learning Approach to Identifying the Thought Markers of Suicidal Subjects: A Prospective Multicenter Trial.

[12] Xiaodong Liu, Pengcheng He, Weizhu Chen, JianfengGao. Multi-Task Deep Neural Networks for Natural Language Understanding; 2019

[13] Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, Xiaoyan Zhu. Sentiment Analysis by Capsules; 2018

[14] Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts C´iceroNogueira dos Santos Brazilian Research Lab IBM Research cicerons@br.ibm.com Ma´iraGatti Brazilian Research Lab IBM Research mairacg@br.ibm.com.

[15] Ji, S., Yu, C. P., Fung, S. F., Pan, S., & Long, G. (2018). Supervised Learning for Suicidal Ideation Detection in Online User Content. Complexity, 2018.