# An Investigation of Deep Learning Systems for Suicide Risk Assessment

**Michelle Morales[1], Danny Belitz[2], Natalia Chernova[1], Prajjalita Dey[1], Thomas Theisen[3]**
[1]IBM Chief Analytics Office
[2]IBM GBS Client Innovation Center Benelux
[3]IBM Rochester Data Science Team
{michelle.morales, prajjalita.dey, thomas.theisen}@ibm.com
danny.belitz-cic.netherlands@ibm.com
ncherno@us.ibm.com

## Abstract

This work presents the systems explored as part of the CLPsych 2019 Shared Task. More specifically, this work explores the promise of deep learning systems for suicide risk assessment.

## 1 Introduction

In the United States alone, on average, approximately 1 person every 11 minutes kills themselves (Drapeau and McIntosh, 2017). In addition, the situation is worsened by the fact that 124 million Americans live in areas where there is a shortage of mental health providers (Bureau of Health Workforce, 2017). Meta-studies have shown that the ability to predict suicide attempts has been near chance for decades, and researchers have argued for the necessity to dedicate research efforts to approaches based on machine learning (Walsh et al., 2017). Machine learning systems which predict suicide risk have the potential to improve identification of people with heightened suicide risk.

This work is part of the 2019 CLPsych Shared Task[1](Zirikly et al., 2019), which focuses on predicting someone's degree of suicide risk using posts they have made on the public forum Reddit. In this paper, we present our team's results from the Shared Task. Specifically, in this work, we focused on two main objectives. The first objective is the exploration of deep learning systems for this particular task. Deep learning systems have demonstrated high performance in various NLP tasks, including text classification, however as is highlighted in past work (Shing et al., 2018), have yet to outperform more shallow machine learning models, such as Support Vector Machines (SVM). In this work, we focus on exploring various deep learning architectures, including convolutional neural networks, long short-term memory networks, and neural network synthesis. We find that deep learning models can outperform more traditional machine learning systems for suicide risk assessment. In addition to exploring the promise of deep learning for risk assessment, we also present results for novelly tested features for this particular task.

[1]http://clpsych.org/shared-task-2019-2/

## 2 Dataset

This work leverages the data provided by the 2019 CLPsych Workshop organizers (Zirikly et al., 2019). Our team's use of this data and participation in these tasks met the ethical review criteria discussed in Zirikly et al. (2019). The dataset includes a series of Reddit users who have posted on the r/SuicideWatch subreddit, with annotations from one of the following four categories: (a) No Risk, (b) Low Risk, (c) Moderate Risk, and (d) Severe Risk. For any models performing within the scope of Task A, the dataset solely includes r/SuicideWatch posts. The Task B dataset includes all of the r/SuicideWatch posts as well as each of the users' posts on any other subreddit. The Task C dataset only looks at the non-SuicideWatch posts for these same users. The dataset includes a post identifier, a user identifier, timestamp, subreddit name, title of the post, and body of the post.

## 3 Feature Engineering

### 3.1 Preprocessing

Preprocessing steps were dependent on task and model necessity. However, an overview of general preprocessing steps adopted across many of the systems included the following: joining of text title and body, lowercasing text, removal of excess punctuation/URLs/additional symbols, stop word removal, and lemmatization.



*Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 177–181
Minneapolis, Minnesota, June 6, 2019. 

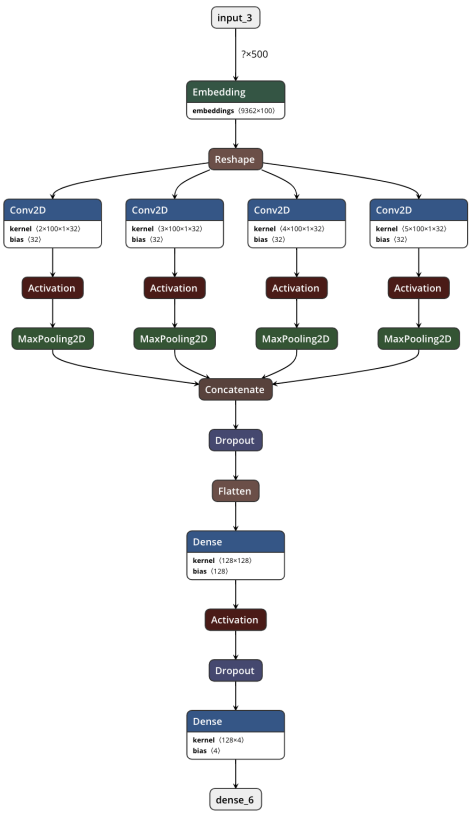


Figure 1: NeuNetS synthesized CNN architecture for Tasks A and B. The only architectural difference between both models is the input dimension.

### 3.2 Bag of Words

We first apply the above preprocessing steps, and then represent the concatenated post and title as a bag of words vector, including unigrams and bigrams with tf-idf weighting.

### 3.3 Topics

We use Gensim's LDA library to create topic models for each of the documents, one document being one post. This gave each document a topic distribution, and those distributions were used as features for the final model. We tested a range of number of topics (specifically 10, 20, 30, 40, 50, 100, 150, 200, and 250 topics) and found the macro-average precision, recall, and f1 score to remain the same, so the LDA model is ultimately trained on 50 topics.

### 3.4 Syntax & Named Entities

We leverage SpaCy's syntactic parser [2] to generate part-of-speech tags (POS) and named entities (NER). POS tags include both coarse-grained POS tags (Google's Universal POS tagset) and fine-grained POS tags (Penn Treebank POS tagset). Counts of each type of tag (for both sets) are taken across each post, and normalized by the word count. For NER tags, counts are taken and normalized by the number of named entities in the document.

### 3.5 Word Embeddings

Various word embedding architectures are explored. For each type, the same data is used for training specifically the entire task dataset (annotated and unannotated).

**Skip-gram**: We compute 100-dimensional embeddings for the entire Reddit corpus using a Skip-gram model (Mikolov et al., 2013), window size 5, and ignoring occurrences of words fewer than 5 times.

**Retrofitted Skip-gram**: For this representation, the trained Skip-gram word embeddings are optimized using the WordNet lexicon. This retrofitting approach is taken from Faruqui et al. (2014), where it was found to help improve performance on text classification tasks.

**FastText**: We also compute FastText embeddings (Joulin et al., 2016) for the entire Reddit corpus. FastText is an extension to the Word2Vec Skip-gram model. However, instead of training on individual words, FastText breaks words into several n-grams (sub-words). This helps capture morphological patterns and overcomes the limitation of Skip-gram when facing out-of-vocabulary words.

### 3.6 Novel Features

To the best of the our knowledge, the following set of features have yet to be explored for suicide risk assessment and/or screening.

**Personality features**: We leverage the IBM Watson Personality Insights API [3] to extract raw scores and percentiles for a variety of personality characteristics, including the Big Five (agreeableness, conscientiousness, extraversion, emotional range, and openness), as well as Needs (e.g. excitement, harmony, etc.) and Values (e.g. conservation, hedonism, etc). Important to note, that the API requires a sufficient amount of data to be provided about a user to extract personality features, namely at least 100 words per user to receive any results, at least 300 words to receive statistically significant results, but preferably even more

[2] https://spacy.io/

[3] www.ibm.com/watson/services/personality-insights/

| System | P | R | F1 |
|---|---|---|---|
| **Task A** | | | |
| SVM (Skip-gram) | .41 | .38 | .36 |
| CNN (Skip-gram) | .38 | .35 | .34 |
| NeuNetS | **.51** | **.64** | **.57** |
| **Task B** | | | |
| kNN (Personality) | .33 | .33 | .32 |
| LSTM (Tone) | .42 | .40 | .41 |
| NeuNetS | **.49** | **.47** | **.48** |
| **Task C** | | | |
| RF (Big 5 only) | .38 | .34 | .31 |
| kNN (Big 5 only) | .33 | .33 | .32 |
| kNN (Big 5 + Values) | .33 | .33 | .32 |

Table 1: Evaluation phase results. Results are reported on a 20% held out portion of the training dataset. Macro precision (P), recall (R), and F1-score reported. Only top 3 systems are reported.

- 600 or 1200 words per user. Given this limitation, these features are only explored for Task C, the screening task, where the most data about a user is given.

**Tone features**: We leverage the IBM Watson Tone Analyzer [4] to extract tone measures with corresponding weights (13 measures in total). The tone measures fall into 3 categories: emotion (anger, disgust, fear, joy, sadness), language (analytical, confident, tentative), and social (openness, conscientiousness, extroversion, agreeableness, emotional range). The tone measures include both the document and sentence level. The document level measures are an aggregation of the individual sentence level tone measures. Analysis on the sentence level provides insight into the range in each tone weight across the whole text body.

## 4 Systems

Systems are trained for three specific tasks. Two of the tasks (Task A and Task B) focus on risk assessment. The third task (Task C) focuses on screening. In addition, all tasks focus on predicting risk at the user level.

### 4.1 Linguistic & Personality Classification Models

Four sets of features are included in the linguistic-based system: topic distributions, syntax features, NER features, and tf-idf vectors. The various feature sets are concatenated together to train models at the post level. Majority voting is then used to aggregate the post predictions to the user level. Various machine learning algorithms are explored including: Random Forest (RF), Naive Bayes, k-Nearest Neighbors (kNN), and linear SVM. Given the imbalanced distribution across class labels, oversampling of the minority classes are performed using the SMOTE technique (Lemaître et al., 2017). During the evaluation phase, the RF model performs marginally better than the rest of the models and is therefore used as the model in the final linguistic-based system. These models are explored for Task A only. For the Personality-based models similar algorithms are explored with different subsets of the personality features tested.

### 4.2 Deep Learning Classification Models

#### 4.2.1 Convolutional Neural Network

The goal of this system was to explore the potential of a Convolutional Neural Network (CNN) for risk assessment. As is highlighted in the task dataset paper (Shing et al., 2018), CNNs have been shown to be effective in many NLP tasks, especially in text classification problems. However, in past work, CNNs have not outperformed more shallow systems for suicide risk assessment. We evaluate the potential of CNN models for this task and explore the impact of various different word embedding inputs. The systems we built using CNNs focus solely on Task A, as this task presents the most challenging problem for a deep learning model, i.e. the smallest data size per user, on average $\sim$1.8 posts per user. CNNs are built using Keras [5] and parameters are optimized using Hyperas [6]. All CNN models are trained on the post-level; user level predictions are made by averaging across the classes' probability distributions, choosing the risk label with the highest probability.

#### 4.2.2 Long Short-Term Memory Network

The goal of this system is to transform a Reddit user's history of posts into a sequence of tone weights over time. This system was used solely for Task B. Tone data was extracted at the document level. The date/time range in post activity for each user varied widely. Some users appeared to be new to the website, while other users had been active on Reddit for years. To partially correct for

[4] www.ibm.com/watson/services/tone-analyzer

[5] https://keras.io/

[6] https://github.com/maxpumperla/hyperas

| System | Accuracy | Macro F1 | Flagged F1 | Urgent F1 |
|---|---|---|---|---|
| **Task A** | | | | |
| CNN (Skip-gram) | .52 | .31 | .89 | .83 |
| NeuNetS | .43 | .18 | .86 | .79 |
| RF (Linguistic) | .40 | .15 | .83 | .76 |
| **Task B** | | | | |
| LSTM (Tone) | .42 | .30 | .79 | .75 |
| NeuNetS | .42 | .21 | .82 | .74 |
| kNN (Personality) | .34 | .28 | .75 | .67 |
| **Task C** | | | | |
| kNN (Big 5 + Values) | .44 | .17 | .55 | .46 |
| kNN (Big 5) | .42 | .18 | .49 | .41 |
| RF (Big 5) | .44 | .12 | .51 | .47 |

Table 2: Results on CLPsych 2019 test set.

this issue only the 10 most recent posts were considered for each user. Another issue arises that the length of time between a users' most recent post and their $10^{th}$ most recent post is not uniform. Thus, any relationship between a tone feature and time is not easily explained. Ultimately, for each user their features are the set of tone weights extracted on their set of maximum 10 posts. Many users had fewer than 10 posts, thus their input data was padded with zeros to maintain a constant input shape. Sequence classification modeling was performed by way of Long-Short Term Memory (LSTM) neural network. The model was utilized to predict user risk of suicide based on each users series of tone data and the corresponding risk level label for the user.

#### 4.2.3 Neural Network Synthesis

In addition to exploring CNNs and LSTMs, we also explore Neural Network Synthesis (NeuNetS). The main objective of NeuNetS (Sood et al., 2019) is to speed up the design of a deep neural network architecture for text or image classification by synthesizing the best deep learning model for a particular dataset. NeuNetS has two main stages: Coarse-grained synthesis and fine-grained synthesis. Based on the data provided, coarse-grained synthesis automatically optimizes and determines the overall architecture of the network - how many layers there should be, how are they connected and so on. The novel step of fine-grained synthesis enables NeuNetS to take a deeper dive into each layer optimizing the individual neurons and connections, e.g. what kind of convolution filter should be applied, and which neurons and edges should be optimized. NeuNetS is explored for both Tasks A and B. Specifically, the goal of these systems were to explore the potential for leveraging a model like NeuNetS to build a strong system for these particular tasks. As model input, the NeuNetS models take the full text (title and body) of users and generates its own word embeddings. The system is trained on the post-level; therefore, predictions for all posts of one user are aggregated into one final label to assess risk for a specific user by majority voting and choosing the higher risk label in case of a tie. The final model architecture can be seen in Figure 1.

## 5 Results

Results from the evaluation phase can be seen in Table 1. Although various combinations were explored, only the Top 3 systems are reported. In the evaluation phase, we explored various feature sets as well as standard and deep learning type classification models. We also explored post level vs. user level training. For both Tasks A and B, we found the NeuNetS systems to perform the highest, reporting a macro F1-score of .57 and .48 respectively. In addition, we found systems trained at the post level to outperform user-based systems.

To further test the robustness of our systems, the Top 3 performing systems are evaluated on the test set. Results from the test phase can be seen in Table 2. These results are reported for predictions made on an unseen test set which were evaluated by the Shared Task organizers. We find the CNN and the LSTM models to perform best across Tasks A and B. Unexpectedly, NeuNetS reports a low F1-score. Although NeuNetS has many procedures in place to prevent overfitting, such as

dropout and regularization, it seems that it still faces the same challenges as more manually designed deep learning architectures. We believe, by design, NeuNetS is more suitable for classification tasks trained on large and balanced data sets (e.g. for text classification the training file size limit is 5GB). For Task A the training data for each label was below the minimum required to train a robust model using NeuNetS. Furthermore, the training data provided for Tasks A and B was imbalanced, providing almost 5 times more labelled posts for label *d* than for label *b*. During training this might cause the model to steer in the wrong direction. This, plus the fact that NeuNetS trains word embeddings on the input alone might be a reason that the resulting model overfits to the training data. Even though various techniques are included in NeuNetS to reduce overfitting, the training data might just be too imbalanced and too small to be a suitable use case for NeuNetS. Also interestingly, for the NeuNetS system, majority voting did not allow for any predictions of labels *b* or *c* although they appeared as intermediate results for some posts. Hence the macro-average F1 score for tasks A and B are rather low. Alternative ways to aggregate might improve these results, e.g. by averaging the confidence scores that are returned for each label. Although we see unexpected results for NeuNetS, we find other deep learning designs to perform well in the tasks, such as the results for the CNN and LSTM systems. These results suggest there is still promise in pursuing deep learning systems for tasks that face data size challenges, such as suicide risk assessment.

## References


Bureau of Health Workforce. 2017. Designated health professional shortage areas: Statistics, first quarter of fiscal year 2018, designated hpsa quarterly summary. *Health Resources and Services Administration (HRSA) U.S. Department of Health and Human Services*.

Christopher W Drapeau and John L McIntosh. 2017. Usa suicide 2016: Official final data. *American Association of Suicidology*.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Han-Chin Shing, Suraj Nair, Ayah Zirikly, Meir Friedenberg, Hal Daumé III, and Philip Resnik. 2018. Expert, crowdsourced, and machine assessment of suicide risk via online postings. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 25–36.

Atin Sood, Benjamin Elder, Benjamin Herta, Chao Xue, Costas Bekas, A. Cristiano I. Malossi, Debashish Saha, Florian Scheidegger, Ganesh Venkataraman, Gegi Thomas, Giovanni Mariani, Hendrik Strobelt, Horst Samulowitz, Martin Wistuba, Matteo Manica, Mihir R. Choudhury, Rong Yan, Roxana Istrate, Ruchir Puri, and Tejaswini Pedapati. 2019. Neunets: An automated synthesis engine for neural network design. *CoRR*, abs/1901.06261.

Colin G Walsh, Jessica D Ribeiro, and Joseph C Franklin. 2017. Predicting risk of suicide attempts over time through machine learning. *Clinical Psychological Science*, 5(3):457–469.

Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. 2019. CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*.