A Comparison of Simple vs. Complex Models for Suicide Risk Assessment

Michelle Morales
Global Business Services
IBM

Prajjalita Dey
Global Business Services
IBM

Kriti Kohli
Corporate Technical Strategy
IBM

{michelle.morales, prajjalita.dey, kkohli}@ibm.com

Abstract

This work presents the systems explored as part of the CLPsych 2021 Shared Task. More specifically, this work explores the relative performance of models trained on social media data for suicide risk assessment. For this task, we aim to investigate whether or not simple traditional models can outperform more complex fine-tuned deep learning models. Specifically, we build and compare a range of models including simple baseline models, feature-engineered machine learning models, and lastly, fine-tuned deep learning models. We find that simple more traditional machine learning models are more suited for this task and highlight the challenges faced when trying to leverage more sophisticated deep learning models.

1 Introduction

Globally 800,000 people die from suicide each year, which makes it one of the leading causes of death (Hannah Ritchie and Ortiz-Ospina, 2015). Despite decades of substantial efforts to analyze risk factors for suicidal thoughts and behaviors (Franklin et al., 2017), models have produced predictions only slightly better than random chance (AUCs=0.56-0.58) (Ophir et al., 2020). Recent progress in Natural Language Processing (NLP) and Machine Learning systems to predict suicide risk have been shown to have higher AUC 0.9 (Coppersmith et al., 2018), however it is still a complicated task particularly due to the sensitivity and difficulty in obtaining high quality labeled datasets.

This work is part of the 2021 CLPsych Shared Task (Macavaney et al., 2021), which provides secure and ethical access to sensitive data in order to work on the problem of predicting suicide risk from social media data. The shared task has two main objectives: prediction of a suicide attempt 30 days prior, and prediction of a suicide attempt 6 months prior. In this paper, we present our team's results from the Shared Task using a variety of methods to improve performance. We focus on exploring various machine learning ensemble models, feature engineering approaches and compare to deep learning architectures and Transfer Learning methods in NLP. We find that baseline models such as Term Frequency, used in combination with simple machine learning models outperform fine-tuned deep learning Transformer-based models.

2 Methods

Our goal for this task was to compare the results of models across different levels of complexity, and see how they perform in the context of a small dataset in the mental health space. All Tweets were aggregated at the user level, and each of the classification methods were implemented and compared at that level.

2.1 Dataset

This work leverages the data provided by the 2021 CLPsych Workshop organizers (Macavaney et al., 2021). Data was provided for a series of Twitter users and all their Tweets for a certain timeframe of history: in Subtask 1 that timeframe was 30 days, while in Subtask 2 the timeframe was 182 days. The dataset also provided true binary labels about past suicide attempts as well as the date of attempt if applicable - a first for this type of shared task, only possible because of the secure computing environment that was provided. Real world binary outcomes have been used in other types of work (Coppersmith et al., 2018).

2.2 Baseline Model

The baseline model provided by the organizers involved a Term Frequency model in conjunction with a Logistic Regression classifier. This method involved simple preprocessing: cleaning hashtags, removing stopwords, and tokenizing Tweets. In

Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology, pages 99–102
June 11, 2021. ©2021 Association for Computational Linguistics

addition, all of the models described in Section 2.3 leveraged the same preprocessing approach.

2.3 Machine Learning Models

2.3.1 Gradient Boosting - Syntax Features

This model used a gradient boosting classifier with an emphasis on manually created grammatical features. Prior research in this space has shown that grammatical and syntactic patterns are a consistent differentiator between individuals characterized with suicide risk and those who are not (O'dea et al., 2017). The features created were intended to measure this, and focused on length and syntax patterns prominent within the user's Tweets. The length features comprised of both average word and sentence count. The syntax related features quantified pronoun usage, differentiating between first, second, and third-person pronouns as well as singular and plural pronouns.

2.3.2 Gradient Boosting - Character TF-IDF

This model used the same gradient boosting model as above, but used a different feature set. Also, this model stemmed the data as an additional preliminary preprocessing step. Instead of manually creating features from the text, this model utilized a character TF-IDF vector. Both gradient boosting models were applied to both Subtasks.

2.3.3 Ensemble Voting Classifier

Our third model used a voting method to create an ensemble machine learning model. Features were created using an n-gram Term Frequency with unigrams and bigrams, across the entire training set, with 5,000 maximum features. We then trained three machine learning models: a Logistic Regression classifier, a Multinomial Naive Bayes classifier, and a Random Forest classifer. We used a soft voting classifier - where the predicted class probabilities for each classifier are collected and averaged - and weighted each classifier equally. The final class label is then derived from the class label with the highest average probability between the three models. We picked conceptually different machine learning classifiers in order to balance out individual weaknesses in the average predicted probabilities.

2.4 Deep Learning Models

Lastly, we explored the effect of using NLP transfer learning methods and fine-tuning deep learning models. For this system, we used BERTweet (Nguyen et al., 2020) - a language model pre-trained on an 80GB corpus of 850M English Tweets - and fine-tuned it on the Shared Task dataset. BERTweet uses the same architecture as BERTbase (Devlin et al., 2018), with a pre-training procedure based on RoBERTa (Liu et al., 2019); it has generally proven to do better than its competitors on Tweet NLP tasks, including text classification. We only applied this deep learning system to Subtask 1, due to the limit on maximum sequence length at 512 and 128 for BERT and BERTweet respectively. Since Subtask 2 comprised of 6 months worth of Tweets its sequence length was above the maximum requirements of BERT and BERTweet, and therefore not included in this part of our investigation.

2.4.1 BERTweet Preprocessing

Before applying BERTweet to the classification task, we normalized the Tweets by following the same preprocessing steps applied to the BERT pre-training corpus. This included tokenizing the Tweets using TweetTokenizer from the NLTK toolkit and using the emoji package to translate emotion icons into text strings. In addition, raw Tweets were normalized by converting user mentions and web/url links into special tokens as provided through the normalization argument in the BERTweet Transformers package (Wolf et al., 2019).

2.4.2 Fine-tuned Model

We explored two fine-tuning methods. In Method 1, we created a BERTweet model instance with a randomly initialized sequence classification head on top of the encoder, of output size 2. In Method 2, we froze the entire architecture and attached a dense neural network layer, updating only the weights of the attached layers.

Both fine-tuning approaches used a maximum sequence length of 128 tokens, and models were optimized using AdamW (Loshchilov and Hutter, 2017), which implements gradient bias correction as well as weight decay. We followed the recommended hyperparameters for fine-tuning as described in Appendix A3 of (Devlin et al., 2018): batch size 16, fixed learning rate of 2e-5, 4 epochs for fine-tuning Method 1 and 10 epochs for Method 2.

In our fine-tuning Method 2, we kept all the weights of the pre-trained BERTweet model frozen and appended a dense linear layer, a dropout layer

	F1	F2	TPR	FPR	AUC
Subtask 1 (30 days)					
Task Baseline	0.636	0.636	0.636	0.364	0.661
Run 1: Char. TF-IDF GB	0.455	0.455	0.455	0.545	0.438
Run 2: Syntax GB	0.500	0.472	0.455	0.364	0.616
Run 3: BERTweet	0.571	**0.656**	0.727	0.818	0.413
Subtask 2 (6 months)					
Task Baseline	0.710	0.724	0.733	0.333	0.764
Run 1: Syntax GB	0.467	0.467	0.467	0.533	0.618
Run 2: Char. TF-IDF GB	0.516	0.526	0.533	0.533	0.591
Run 3: Voting Classifier	0.727	**0.769**	0.800	0.400	0.720

Table 1: Model results on CLPsych test set as compared to the task baseline system.

to reduce overfitting, and a softmax layer. The model was trained using a cross-entropy loss function. We computed the task performance after each training epoch on a validation set and selected the best model checkpoint to compute the performance on the test set.

3 Results

In Subtask 1, our models are as follows: Run 1 refers to the character TF-IDF gradient boosting model, Run 2 refers to the syntax gradient boosting model and Run 3 refers to the BERTweet model using fine-tuned Method 1. In the validation experiments, we found BERTweet fine-tuned Method 1 to outperform Method 2. In Subtask 2, Run 1 refers to the syntax gradient boosting model, Run 2 the character TF-IDF model, and Run 3 the voting classifier.

We see that in the case where the BERTweet model could be applied, it outperformed more simple machine learning models. However, although the BERTweet model had a high F1, F2, and TPR, it has a high FPR and a low AUC score - this implies that the model is overfitting, and has a tendency to predict 1s.

In the case where BERTweet could not be applied (Subtask 2), having an ensemble model fared better than the single gradient boosting models. The voting classifier outperformed the baseline in most metrics (F1, F2, TPR) but also had a nominally higher FPR and lower AUC score than the baseline. The increased FPR corresponds to misclassifying one negative sample as a positive sample. For assessing suicide risk though, we feel that it is better to overpredict suicide risk than underpredict, since the consequences of underpredicting are much more severe.

F2 score gives less weight to precision and more weight to recall therefore prioritizing the proportion of actual positives that were correctly identified. Both BERTweet (Subtask 1) and the voting classifier (Subtask 2) have higher F2 score than the baseline, however F2-score alone is an unsuitable metric as a classifier that predicts all 1s would have a recall of 1. The AUC is widely used to as a measure for predictive modeling accuracy, however, AUC is not recommended for small sample sizes (Hanczar et al., 2010). Overall, looking at all the metrics in Table 1 holistically is recommended.

4 Discussion

For Subtask 1, in the Transfer Learning methods, we tried two fine-tuning techniques. In the first approach, i.e. Method 1, we instantiate a BERTweet model with an added single linear layer on top for classification. In this approach, the entire pretrained BERTweet model and the additional untrained classification layer is trained on our specific task. The average accuracy with the validation set was 0.51 and 0.45 for the test set, suggesting overfitting of the model. For the second approach, i.e. Method 2, we freeze all the layers of BERTweet and only update the weights of the attached layers. While the training loss decreased for the first 4 epochs, it did not decrease further, suggesting that the model was trained for too long and is also overfitting on the training data. While both approaches suggested that such a small dataset caused overfitting, a simple fine-tuning approach through adding one fully-connected layer to BERTweet and training the whole model end-to-end for a few epochs (Method 1) showed better results than appending a custom architecture to the frozen BERTweet model (Method 2). As all Tweets were aggregated into one large Tweet at the user level and the sequence length was limited to 128, effectively this approach reduced the dataset from Tweets of the last 30 days to the last 1-3 days depending on the Tweet length. This causes loss of potentially valuable data and features that may be missed as these particular models cannot learn from the older Tweets. As the machine learning models do not have these limiting properties, they are more suitable for this task. A recommendation for future work is to transform the dataset in an alternate manner, for example, creating a classification task at the Tweet level instead of the aggregated User-Tweet level.

5 Conclusion

The main question we sought to explore in this paper was the following, *would a classical machine learning model approach outperform a more sophisticated deep learning model for the suicide risk assessment task?* Given past research in this space that struggled with this task as well as the small nature of the datasets, it was our hypothesis that keeping it simple would lead to better performance. Our findings support this hypothesis. We found that BERTweet struggled with overfitting and demonstrated limitations, such as sequence length, that made it difficult to leverage for this task. In our evaluations, we found that a simple baseline model, or an ensemble of machine learning models can outperform the more sophisticated models. In addition, the short time period inherent in building a model for a Shared Task made it difficult to investigate alternate data transformations that are more appropriate for a complex model like fine-tuned BERT/BERTweet. However, we do find some promise in the test performance of BERTweet for Subtask 1 and believe with more time and exploration a variation of Transfer Learning models can be built and leveraged in a task of this nature.

Ethics Statement

Secure access to the shared task dataset was provided with IRB approval under University of Maryland, College Park protocol 1642625.

References

Glen Coppersmith, Ryan Leary, Patrick Crutchley, and Alex Fine. 2018. Natural language processing of social media as screening for suicide risk. *Biomedical Informatics Insights*, 10:1–11.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Joseph Franklin, Jessica Ribeiro, KR Fox, KH Bentley, EM Kleiman, X Huang, KM Musaccchio, AC Jaroszewski, BP Chang, and MK Nock. 2017. Risk factors for suicidal thoughts and behaviors: A meta-analysis of 50 years of research. In *Psychol Bull.*, pages 143(2):187–232.

Blaise Hanczar, Jianping Hua, Chao Sima, John Weinstein, Michael Bittner, and Edward R. Dougherty. 2010. Small-sample precision of ROC-related estimates. *Bioinformatics*, 26(6):822–830.

Max Roser Hannah Ritchie and Esteban Ortiz-Ospina. 2015. Suicide. Https://ourworldindata.org/suicide.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Sean Macavaney, Anjali Mittu, Glen Coppersmith, Jeff Leintz, and Philip Resnik. 2021. Community-level research on suicidality prediction in a secure environment: Overview of the CLPsych 2021 shared task. In *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology (CLPsych 2021)*. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Bridianne O'dea, Mark E Larsen, Philip J Batterham, Alison L Calear, and Helen Christensen. 2017. A linguistic analysis of suicide-related twitter posts. *Crisis: The Journal of Crisis Intervention and Suicide Prevention*, 38(5):319.

Yaakov Ophir, Refael Tikochinski, Christa S. C. Asterhan, Itay Sisso, and Roi Reichart. 2020. Deep neural networks detect suicide risk from textual facebook posts. In *Scientific Reports. 10*, page 16685.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.