Using Contextual Representations for Suicide Risk Assessment from Internet Forums

Ashwin Karthik Ambalavanan, Pranjali Dileep Jagtap, Soumya Adhya, Murthy Devarakonda*

Arizona State University, Tempe, AZ

Abstract

Social media posts may yield clues to the subject's (usually, the writer's) suicide risk and intent, which can be used for timely intervention. This research, motivated by the CLPsych 2019 shared task, developed neural network-based methods for analyzing posts in one or more Reddit forums to assess the subject's suicide risk. One of the technical challenges this task poses is the large amount of text from multiple posts of a single user. Our neural network models use the advanced multi-headed Attention-based autoencoder architecture, called Bidirectional Encoder Representations from Transformers (BERT). Our system achieved the 2nd best performance of 0.477 macro averaged F measure on Task A of the challenge. Among the three different alternatives we developed for the challenge, the single BERT model that processed all of a user's posts performed the best on all three Tasks.

1 Introduction

Social media has become an important part of everyone's life, and in particular online discussion forums related to mental health provide opportunities for individuals to share their feelings and their state of mind. These self-documented posts are valuable in assessing suicidality and potentially offering interventions. Since the volume of posts and the time sensitivity of potential interventions, automation is critical for monitoring the forums.

The CLPysch 2019 shared task[1,2] used the University of Maryland Reddit Suicidality Dataset, which was developed using data from Reddit, a well-known online site for anonymous discussion forums on a wide variety of topics. As described in Shing et al[3] the data was extracted from the 2015 Full Reddit Submission Corpus, including postings from one specific *r/SuicideWatch* subreddit forum (denoted as SW here), to identify suicidality risk of the post subject. The data contained post id, anonymous user id, timestamp, subreddit name, post title, and post body. The training data contained four labels, denoted by a to d, with increasing risk of suicidality of low risk to the highest risk. The organizers also provided a separate control group of users and their subreddit posts with no known suicidality risk. The challenge involved three different subtasks, and they were:

- Task A: Assess the subject's suicide risk (a, b, c, or d) from a user's SW postings only;
- Task B: Similar to the Task A, but the user's posts from other subreddits are also used;
- Task C: Assess the subject's suicide risk from a user's Reddit posts other than from SW;

Note that the prediction is not for a post but for a user (actually for the subject of a user's posts). The training dataset contained 496 users with as an average of 1.85 posts per user. The test dataset contained 125 users. The gold standard was based on a consensus of human annotators.[3]

The challenge used macro average of the F measures of the four labels, a, b, c, and d as the system performance indicator. The evaluation also provided *accuracy* (number of correct predictions divided by the number of all predictions), and F measures of *flagged* and *urgent* predictions. The flagged predictions measure performance of identifying b, c, d, out of the four labels, and the urgent predictions measure the performance of identifying

*Contact author: mvd@acm.org

Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology, pages 172–176
Minneapolis, Minnesota, June 6, 2019.

c and d out of the four labels. We proposed three different methods based on BERT (Bidirectional Encoder Representations from Transformers).[4] An important hypothesis we considered here was if a model that is built for general domain NLP and only fine-tuned with the suicide-related training data performs on suicide prediction. As such, we did not make use of the suicide literature or the theories of suicide in our methods.

2 Methods

2.1 Pre-Processing and Data Preparation

We pre-processed the text from the posts and presented the resulting text (a sequence of words) as the input to the model. This processing was common to all three methods and Tasks, although some steps are only relevant to certain Tasks, as described below.

- Removed stop words (am, the, for, etc.) and punctuations
- Expanded contractions like *couldn't* to *could not* for easier interpretation and to avoid awkward splitting of words.
- Concatenated words from all posts in a sequence of decreasing order of timestamp so that the most recent post is considered first based on the intuition that the latest psychological state of a user is based on his or her most recent post. The first word in most recent post occupies the first word location in the sequence.
- For Task B and C the subreddit name was prepended to the corresponding posts. The intuition was that the subreddit name might provide a clue to the model.

In addition, we also made the following adjustments to the data:

- Class "b" was over sampled to class "a" instance count by random oversampling class b instances. This was because, given its low frequency in the given training data, our preliminary models couldn't identify the class "b" very well.
- For Task B, posts of the control users were ignored from the training set for the simplicity and also because it was not necessary for the model to predict them in the test set.
- For Task C, a None label was used for the control users and the model was trained using 5 labels instead of 4. Then post processing converted all None labels to class 'a' label as per the challenge requirements.

2.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT[4] is a new exciting development in neural network models research, demonstrating significantly improved state-of-the-art performance on various general domain NLP tasks, including text classification. BERT pre-trained model produces sequence (i.e. sentence) level and word level representations, which can be fine-tuned for task-specific outcomes. BERT includes an advanced auto-encoder architecture to generate the representations. The pre-trained BERT model, after fine-tuning for a task, has been shown to perform well on multiple general domain NLP tasks, using only an additional, simple feed-forward network with a softmax layer.[4]

The BERT approach is distinctly different from the most biomedical NLP architectures where word2vec and similar representations of words are used as input that is processed by complex task-specific, heavily-engineered architectures containing Bidirectional LSTMs, RNNs, and CNNs. BERT provides a broadly applicable pre-trained model which only need to be fine-tuned using task-specific training data. BERT uses layers of neural network components known as Transformer encoders to generate representations of input words and sequences. See Figure 1. The Transformer encoders contain layers of bidirectional multi-headed self-attention[5] encoders with residual connection around each layer. Intuitively, an Attention layer produces output (say, a sentence representation) that is based on any arbitrary word positions of the input sequence. See Figure 2. Multi-headed attention can simultaneously optimize for various input combinations.

2.3 BERT-Fine-Tuning

The pre-processed (combined) posts of each user is given as input to the BERT Model with a Linear classifier (SoftMax) to predict the output labels (Fig 1). Fine tuning helps to tune the initial embeddings of BERT to the CLPsych downstream task with the help of error backpropagation. The BERT Implementation in PyTorch[6] was used to implement this architecture. The configuration that gave the best results on a validation subset of the training data was a maximum sentence length of

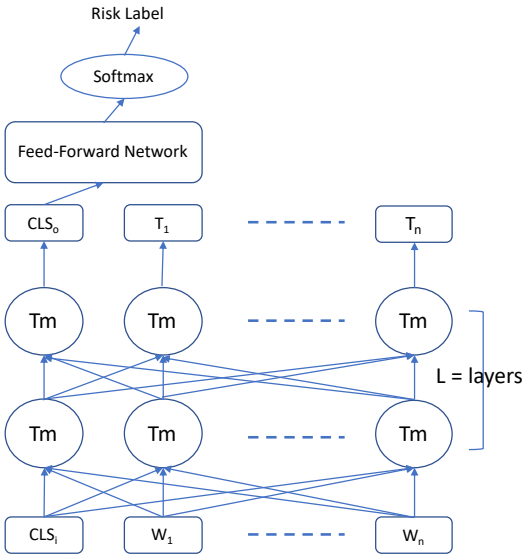

Figure 1. BERT architecture along with the feed-forward network and the softmax layer

384 tokens with a batch size of 16 and 75 epochs. Note that no theories of suicide or lexicons specific to suicide were used in the model. Instead we let BERT learn task-specific lexical clues through fine tuning on all of a user's text and make predictions on unseen posts.

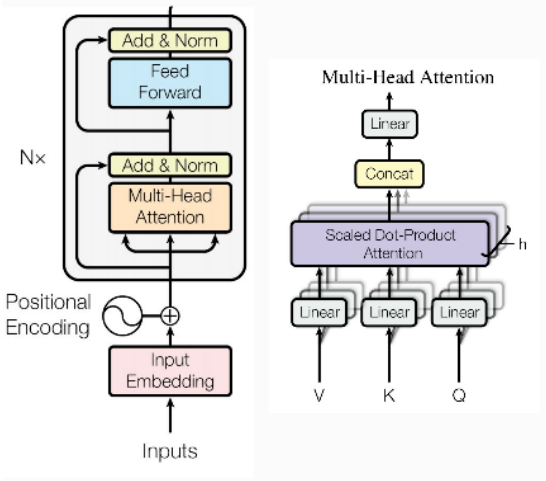

Figure 2. The transformer architecture used in BERT, with acknowledgements to Vaswani et al[5]

2.4 BERT-Sentence-Embedding + BiLSTM + ATTN

The sentences of the combined posts were extracted using NLTK. The representation for each of the sentences was obtained from the pre-trained BERT model. The [CLS$_o$] BERT output, as shown in Figure 1, provides this representation. All sentence representations of a user were concatenated as a sequence and fed into a BiLSTM+Attention[7] model with a linear projection layer and a softmax layer. The maximum number of sentences per user was set as 50 and the model was run for 200 epochs with a batch size of 10. The intuition behind this model is to see if sentence level aggregation at the input results in better learning and prediction from a user's posts.

2.5 BERT-Multiple-Instance-Learning

Pre-Processing of a user's posts was the same as before except that the posts, ordered in decreasing order of timestamps, were kept separate. Each post was separately processed by the fine-tuned BERT model (from Task A) and a post-level representation was produced at the [CLS$_o$] output. Multiple independent BiLSTM+Attention[7] models analyzed the post-level representations. The output vectors from the BiLSTM+Attention models were concatenated, fed to a Linear projection layer with ReLU activation and dropout of 0.4, and then finally to a linear projection layer with a softmax for classification.

We configured this multi-instance learning model with five BiLSTM+Attention models because an average user had at most 5 posts. If a user had more than 5 posts, all older posts (after 5 posts) were ignored and if user had less than 5 posts, nulls were fed as input to the corresponding models.

The model takes time to fit to the data and gives poor results after the first 5 epochs but at around 10 epochs it tends to learn better and then overfits rather quickly. Thus, the use of the dropout layer was very important in this implementation to prevent overfitting. Our best configuration for this model ran 20 epochs with a batch size of 16. The intuition behind this model is to retain the word level input but to aggregate decisions from separate models each of which analyzes a single post.

3 Results

The results, determined by the organizing team from our system output, for our models and the Tasks are shown in Table 1. Generally, across the broad, the BERT-Fine-Tuning model achieved the best results in our experiments. BERT-Multi-Instance-Learning model performed close to the fine-tuning model, and in fact outperformed it on the

flagged metric. We used the BERT-Sentence-Embedding model only for Task A, and its performance was the lowest of our models.

Table 1. Evaluation results for the methods proposed in this this study on the test dataset as reported by the Challenge organizing team.

Tasks	Measures	BERT-Fine-Tuning	BERT-Sentence-Embedding + BiLSTM+Attn	BERT-Multi-Instance-Learning
Task A	Macro F meas.	**0.477**	0.418	0.421
	Accuracy	**0.544**	0.512	0.504
	Flagged F meas.	0.882	0.875	**0.891**
	Urgent F meas.	**0.826**	0.795	0.812
Task B	Macro F meas.	**0.261**	---	0.169
	Accuracy	**0.368**	---	0.200
	Flagged F meas.	**0.765**	---	0.577
	Urgent F meas.	**0.691**	---	0.286
Task C	Macro F meas.	**0.159**	---	0.143
	Accuracy	**0.597**	---	0.153
	Flagged F meas.	**0.630**	---	0.455
	Urgent F meas.	**0.575**	---	0.342

Table 2. **Task A** results for all participants. The highest scores in each metric are underlined, and our system performance was shown in bold-italic.

Team	Macro F meas.	Accuracy	Flagged F meas.	Urgent F meas.
CLaC	0.481	0.504	0.922	0.776
ASU	***0.477***	***0.544***	***0.882***	***0.826***
HLAB	0.459	0.560	0.842	0.839
Text2Knowledge	0.445	0.544	0.852	0.789
CAMH	0.435	0.528	0.897	0.783
ttu	0.402	0.504	0.902	0.844
Affective_Computing	0.378	0.592	0.920	0.862
cmu	0.373	0.472	0.876	0.773
jxufe	0.364	0.464	0.882	0.779
UniOvi-WESO	0.312	0.512	0.897	0.821
usiupf	0.291	0.376	0.753	0.707
ibm_data_science	0.178	0.432	0.861	0.788

In the challenge our BERT-Fine-Tuning model achieved the 2nd place on Task A (See Table 2). However, our models did not fare well on Task B and C, in part because we could not invest enough time and resources to analyze and/or enhance our models for these tasks in the short time that was available. Across all participating teams, the best performance on Task B was 0.451 macro F measure and Task C it was 0.268. So, the performance of all participating systems was low on Task C.

4 Discussion and Conclusion

The methods we proposed here did not make use of any suicide-specific domain knowledge and yet our system finished close second on Task A. This is significant because it indicates that the BERT model that uses transfer learning from general domain NLP can perform well on a domain-specific dataset[8] after only fine-tuning for the specific domain. It suggests that the new generation of auto-encoder architectures, with pre-trained models, can potentially reduce the need for domain-specific features, lexicons, and pre-training. However, it would be interesting to explore the possibility of further customization to the domain and improvement such models can achieve from it.

In developing the methods for this task, which we mapped the task to text classification, one of the challenges was how to deal with large input text. For example, for the Task A, 27% of users had more than one post, and 4% of users had more 5 posts. Maximum length of a post (in the Task A training set) was 8457 words, and 124 posts had more than 512 words. Task B input is even larger since all posts are included not just SW. So, since our primary method (BERT-Fine-Tuning) used stop-word eliminated, truncated sequences up to 384 words as the input, we attempted to include larger parts of a user's posts with the other two methods. However, the results did not show improvement over the primary method, indicating that either the most recent words are the most important or our secondary methods do not represent larger text well.

In conclusion, this study applied the state-of-the-art, general domain, pre-trained neural network model, BERT, and achieved good performance on a domain-specific task. Future research includes error analysis, improvement of our methods with and without the use of domain-specific knowledge.

References

1. CLPsych 2019. http://clpsych.org/shared-task-2019-2/. Published 2019.
2. Zirikly A, Resnik P, Uzuner Ö, Hollingshead K. 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*. ; 2019.
3. Shing H-C, Nair S, Zirikly A, Friedenberg M, Daumé III H, Resnik P. 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*. ; 2018:25-36.

4. Devlin J, Chang. M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv Prepr arXiv181004805 (accepted to NAACL 2019)*. 2018.
5. Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need. In: *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA; 2017. doi:10.1017/S0952523813000308
6. PyTorch Pre-Trained BERT. https://github.com/huggingface/pytorch-pretrained-BERT. Published 2019.
7. Dandala B, Joopudi V, Devarakonda M. Adverse Drug Events Detection in Clinical Notes by Jointly Modeling Entities and Relations using Neural Networks. *Drug Saf*.:0-2. doi:10.1007/s40264-018-0764-x
8. CLPsych 2017: Triaging content in online peer-support forum. http://clpsych.org/shared-task-2017.