Contextual Representation of Self-Disclosure and Supportiveness in Short Text

Chandan Reddy Akiti[1], Sarah Rajtmajer[1], and Anna Squicciarini[1]

Pennsylvania State University, University Park, PA 16804, USA

Abstract. As user engagement in online public discourse continues to richen, voluntary disclosure of personal information and its associated risks to privacy and security are of increasing concern. Users are often unaware of the sheer amount of personal information they share across online forums, commentaries, and social networks, as well as the power of modern AI to synthesize and gain insights from this data. We develop a novel multi-modal approach for the joint classification of self-disclosure and supportiveness in short text. We take an ensemble approach for representation learning, leveraging BERT, LSTM, and CNN neural networks.

1 Introduction

As public discourse facilitated through social media and online forums grows increasingly commonplace, voluntary disclosure of personal information has been normalized. But users are often unaware or under-aware about the threat of self-disclosure to their privacy and security. We argue that public self-disclosure is often *encouraged* and even *primed* by a false sense of intimacy, as well as topics and tone of conversations. Accordingly, we aim to leverage contextual representations afforded by deep neural language models for the detection of self-disclosure and supportive text.

The application of deep learning to NLP is made possible by representing words as vectors in a low-dimensional continuous space. Traditionally, these word representations were static. Each word was represented by a single vector, regardless of the context. However, this approach had fundamental deficits for tasks like sentiment analysis where the representation of a word in context is critically important. Instead, recent work, e.g., [1], has shown that contextual word representations increase performance on NLP tasks.

Deep neural language models such as BERT [2] and GPT-2 [3] represent successful attempts to create contextualized word representations. Replacing static with contextualized representations has led to significant improvement in a number of NLP tasks [2]. In this work, we use the BERT pre-trained model from the huggingface [4] library.

Following, we present our approach and results for the CL-Aff Shared Task on the OffMyChest conversation dataset put forth in the AAAI-2020 workshop on Affective Content Analysis. The task involves multi-class classification of sentences with 6 labels: emotional disclosure; information disclosure; support; general support; information support; and, emotional support.

We propose two alternative models for the task of detection of self-disclosure and supportiveness in online users' generated comments. The first model is classification using BERT, a bi-directional transformer. We use this model to fine-tune our word representations and for classification using sentence representations and hidden attentions obtained from the model. The second model is classification using a CNN, where we replace the typical embedding layer with the pre-trained BERT model. We apply multiple convolutions using different window sizes (number of words).

2 Problem Definition

The OffMyChest conversation dataset consists of 12,860 labeled sentences and 5,000 unlabeled sentences sampled from comments on subreddits within the OffMyChest community, with the following tags: "wife"; "girlfriend"; "gf"; "husband"; "boyfriend" and "bf". For example, the following training sentence is labeled for emotional disclosure and emotional support: *I hope this chapter results in a better, healthier, more fulfilled you!!*
Of particular interest in this dataset is the General Support label. This label is intended for sentences offering support through catch phrases or quotes, which are exceptionally difficult to distinguish using automated approaches.

Label	Frequency
Emotional Disclosure	0.44
Information Disclosure	0.61
Support	0.35
General Support	0.06
Information Support	0.11
Emotional Support	0.08

Table 1. Label frequency for each label in training data.

As indicated in Table 1, sub-labels for supportiveness suffer from class imbalance. We address this issue with weighted batch sampling. That is, we assign a weight to each sample given by the inverse frequency of occurrence of the label assigned to it, and then draw each mini-batch from the multinomial distribution based on the sample weights. Thus, highly-weighted samples are sampled more often for each mini-batch.

3 Modeling Approach

Our model leverages the generalizability of Bidirectional Encoder Representations from Transformers (BERT). As BERT models tend to generalize better for the diverse set of NLP tasks [5], we use this capability in our transfer learning approach. Sentence-level representations obtained from the BERT model can also be fine-tuned on the unlabeled dataset to better generalize on unseen data for our task.

BERT has its origins in Semi-supervised Sequence Learning [8], Generative Pre-Training [13], ELMo [6], and ULMFit [7]. Unlike previous models, though, BERT is a deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus [2]. This makes it particularly suitable for our task, as it allows us to input training text *as-is*, without imposing predefined and possibly biased features or setting hyper-parameters that would require further analysis.

Context-free models such as word2vec [11] and GloVe [12] generate a specific word embedding representation for each word in the vocabulary. That is, the same word used in different contexts have the same word-representation. While, the BERT model accounts for context of a sentence using the words both preceding and following. BERT has achieved state-of-the-art performance on eleven NLP tasks [2].

We present two classification models. The first model is a BERT model for sequence classification fine-tuned on the unlabeled text corpus. The second model is a CNN model where we use BERT word embedding instead of a typical embedding layer for words. We use the CNN model to visualize which parts of the sentences are contributing to the classification.

3.1 Model 1: BERT model with fine-tuning

BERT-base model has 12 layers with a hidden size of 768 and 12 self-attention heads. This model is pre-trained on BooksCorpus (800M words) [10] and English Wikipedia (2,500M words). We fine-tune the BERT model using a Masked Language Model (LM) and the post and comment data provided as the unlabeled dataset.

For fine-tuning using Masked LM, we prepare a text corpus with all the posts and comments and feed it to the model. The model masks 15% of the words in each sentence and tries to predict them using the other 85% words in the sentence. Thus we fine-tune the language model to predict the words in the corpus accurately in the context of surrounding words. For this task, we use a learning rate of *5e-5*, weight decay of *0*, Adam epsilon of *1e-8* and gradient clipping at *1.0*. We train the Masked LM for 1 epoch.

Sentence tokenization We add *[CLS]* and *[SEP]* tokens at the start and end of every sentence. These token are used in the BERT model to indicate the start and end of a sequence. We then generate word indices for all the tokens in the sentence using a BERT tokenizer.

Sentence padding The sentences in the training data and test data have an average length of 17 words and maximum length of 171 words. We pad all sentences to 200 words in length to avoid truncating the tokenized sentences.

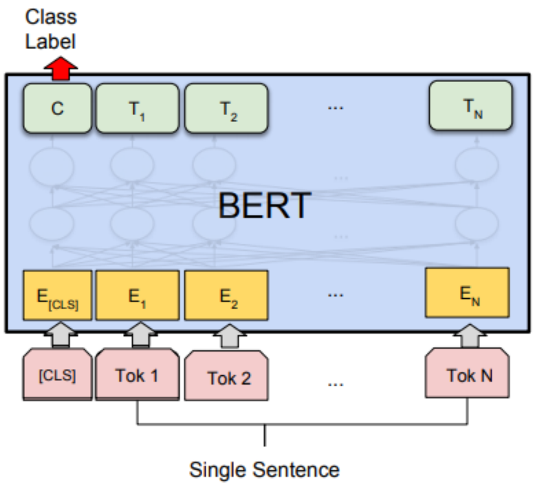


Fig. 1. Sentence classification using Bidirectional Transformers [2].

Sentence classification For sentence classification, we add a linear layer on top of the pooled output of *[CLS]* token. We then apply softmax on the output of the linear layer to obtain the classification probability for the label.

parameter	value
optimizer	Adam
lr	1e-5
weight decay	0
pre-trained model	bert-base-uncased
batch size	32
epochs	2

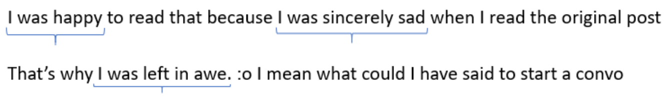
Table 2. Training parameters for Model 1.

Training parameters are set according to Table 2 We measure the cross-entropy loss for each mini-batch and back-propagate the loss.

3.2 Model 2: CNN with BERT embedding

As an alternative model, we deploy the CNN model presented in [9] for text classification. The intuition behind this model is to process window of words in the contextual representation of sentences and aggregate the decision from all the windows. For this purpose we represent the sentence in a matrix and design convolution kernels of different sizes, each size corresponding to a window size of words in the sentence.

We tokenize the sentences in a similar fashion as we did with BERT model. If the tokenized sentence has n tokens where $(n < 200)$, we pad the tokens


I was happy to read that because I was sincerely sad when I read the original post

That's why I was left in awe. :o I mean what could I have said to start a convo

Fig. 2. Sample sentences for Emotional Disclosure. The word windows, e.g., "I was happy" and "I was left in awe" activate for the classification of Emotional Disclosure.

with zero tokens to get standardised length of $N = 200$ tokens. We then obtain word representations of the N tokens from BERT-base pre-trained/fine-tuned model. The BERT-base model has a hidden length of $H = 768$. Thus we take the representation of a sentence as an $N \times H$ matrix.

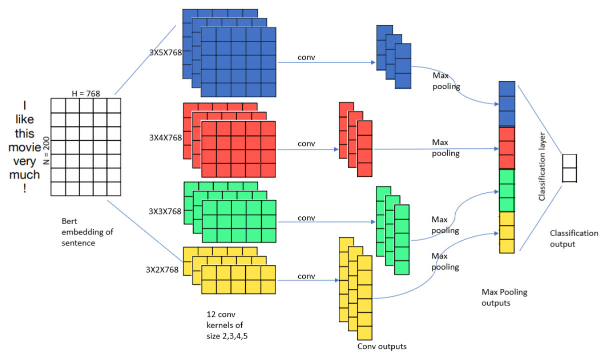


Fig. 3. CNN model with Bert embedding, modified from [9].

parameter	value
optimizer	Adam
lr	1e-3
weight decay	0
batch size	32
epochs	30

Table 3. Training parameters for Model 2.

We set 4 types of convolution kernels of filter size 2 through 5 and filter width equal to the length of word representations, H. Suppose a kernel w_h with filter size of h is used for convolution on the sentence A on the window of i^{th} token to $(i + h - 1)^{th}$ token, we obtain kernel output o_{hi} as $o_{hi} = w_h \cdot A[i : i + h - 1]$ where o_h is the convolution output of kernel w_h on sentence A. We add a bias term b_h and activation function f to each o_{hi}, then we have $c_{hi} = f(o_{hi} + b_h)$

In this way, for each kernel, we obtain an $N \times 1$ output c_{hi} after a padded convolution and activation. The output c_{hi} is an indicator for the weight of the window i of length h for the classification task. The output of the convolution layer can also be visualized as a heat-map for the sentence classification task.

To maximize learning, we use 12 kernels: 3 of each kernel type. We apply max-pooling on each of the convolution output to obtain 4 outputs for each of the kernels. Finally, a linear layer is applied on the outputs for classification. Loss is measured using cross entropy loss function and back-propagated for training. We use an Adam optimizer with learning rate of 1e-3, a batch size of 32 and training for 30 epochs and we choose the model with best validation f1-score.

Visualization of activations We visualize activations in the CNN network during classification. Given a sentence A, we pass the input as a single element batch to the model and extract activations from c_h for each window size h. In Figure 6, we plot the heat map for each word. For a window size h, for a word A_i, we measure the heat map as the sum of activations of all the windows in which the word A_i is included. We then normalize the heat maps across the sentence to plot activations with respect to windows size h.

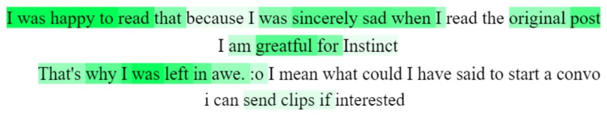

I was happy to read that because I was sincerely sad when I read the original post
I am greatful for Instinct
That's why I was left in awe. :o I mean what could I have said to start a convo
i can send clips if interested

Fig. 4. Activations in the CNN model for Emotional Disclosure. $h = 2$ for first two sentences and $h = 4$ for the third sentence. The fourth sentence does not contain emotional disclosure and has low activation values.

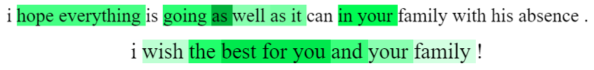

i hope everything is going as well as it can in your family with his absence .
i wish the best for you and your family !

Fig. 5. Activations in CNN model for Emotional Support. $h = 2$ for both sentences.

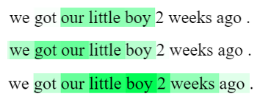

we got our little boy 2 weeks ago .
we got our little boy 2 weeks ago .
we got our little boy 2 weeks ago .

Fig. 6. Activations in CNN model for Information Disclosure. $h = 2$, 3 and 4 respectively for the sentences.

4 Results

We measure accuracy against our labeled dataset using standard accuracy metrics, i.e. precision, recall, F1-score and AuROC.

To avoid false high accuracy on the imbalanced dataset, we measure precision and recall of only true values for the labels. AuROC measures how well the model is able to distinguish the true labels from false labels.
Model 1 results Results for our BERT model are reported in Table 4. As shown, BERT provides a mean F1-score of 0.525 with mean precision 0.45 and mean recall 0.655. Emotional Disclosure, Information Disclosure and Support labels had no data imbalance problem, thus accounting for reasonable precision. The model performs well on these three labels with a mean F1-score of 0.60.

However, General Support, Information Support and Emotional Support labels have high data imbalance with approximately 10% true labels. Low precision for these labels indicate high false positive rate. The model performs very poorly on *General Support*. This is expectable, given the difficulty in distinguishing catch phrases and quotes from usual text. Performing weighted sampling based on the labels increased the recall but considerably decreased the precision.

Label	Precision	Recall	F1-score	AuROC
Emo Disclosure	0.48	0.68	0.57	0.74
Info Disclosure	0.60	0.64	0.62	0.74
Support	0.55	0.78	0.61	0.82
General Support	0.26	0.42	0.32	0.73
Info Support	0.39	0.72	0.49	0.84
Emo Support	0.44	0.69	0.54	0.85

Table 4. BERT fine-tuned model: average scores on 10-fold cross validation.

Model 2 results The CNN model provides a mean F1-score of 0.485 with precision 0.417 and recall 0.592. Emotional Disclosure, Information Disclosure and Support labels had no data imbalance problem, thus accounting for reasonable precision. The model performs well on these three labels with a mean F1-score of 0.58.

Label	Precision	Recall	F1-score	AuROC
Emo Disclosure	0.43	0.67	0.53	0.69
Info Disclosure	0.56	0.67	0.61	0.72
Support	0.57	0.70	0.62	0.83
General Support	0.20	0.39	0.26	0.75
Info Support	0.38	0.56	0.45	0.83
Emo Support	0.36	0.56	0.44	0.83

Table 5. CNN model with BERT embedding: average scores on 10-fold cross validation.

Comparing the two models, BERT performs better as we fine-tune the model while performing the classification task itself. In contrast, the CNN model takes as input static word embeddings from BERT. We do not propagate loss from the BERT model; we only train the CNN.

5 Conclusion

In this work, we have shown that the use of contextual embedding performs well on complex sentence classification tasks. We have also tested an alternative model with a CNN classification layer. We found the contextual embedding performed with comparable accuracy, despite lack of fine-tuning. We have presented visualizations of the convolution activations for the classification task.

Currently we use the BERT-base pre-trained model with only 12 layers. However, there are other variations of BERT models with additional parameters. In future work, we plan to vigorously validate model parameters such as number of layers and add text pre-processing, e.g., stemming, to further clean the data. We suspect that fine-tuning with pre-processed text data would improve generalization on test data.

We also plan to dissect bi-directional transformers like BERT and XLNet to enhance contextual word representations for tasks specific to disclosure and support classification in spoken dialogue systems. Finally, we would like to model effects of peer influence on self-disclosure.

References

1. Zhilin Yang, Ruslan Salakhutdinov and William W. Cohen. Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks, 2017; arXiv:1703.06345.
2. Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018; arXiv:1810.04805.
3. Radford, Alec and Wu, Jeff and Child, Rewon and Luan, David and Amodei, Dario and Sutskever, Ilya: Language Models are Unsupervised Multitask Learners (2019)
4. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing, 2019; arXiv:1910.03771.
5. Yaru Hao, Li Dong, Furu Wei and Ke Xu. Visualizing and Understanding the Effectiveness of BERT, 2019; arXiv:1908.05620.
6. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee and Luke Zettlemoyer. Deep contextualized word representations, 2018; arXiv:1802.05365.
7. Jeremy Howard and Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification, 2018; arXiv:1801.06146.
8. Andrew M. Dai and Quoc V. Le. Semi-supervised Sequence Learning, 2015; arXiv:1511.01432.
9. Ye Zhang and Byron Wallace. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification, 2015; arXiv:1510.03820.
10. Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba and Sanja Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books, 2015; arXiv:1506.06724.
11. Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, 2013; arXiv:1301.3781.
12. Sakakibara, Y. (1992). Efficient Learning of Context-Free Grammars from Positive Structural Examples. Inf. Comput., 97, 23-60.
13. Alec Radford: Improving Language Understanding by Generative Pre-Training (2018)