Twitter Sentiment Analysis Using Deep Neural Networks

Shubhra Bedi
University of Toronto
shubhra.bedi@mail.utoronto.ca

Saumya Bakshi
University of Toronto
saumya.bakshi@mail.utoronto.ca

Nazar Georgis-Yap
University of Toronto
nazar.georgis.yap@mail.utoronto.ca

Abstract

Sentiment analysis is referred to as text organization that is used to classify expressed sentiment polarities. While there are many models that can analyse sentiments, for the purpose of this paper, we will focus on an Attentional Encoder Network (AEN) that employs a BERT model and SMART model that includes a smooth-inducing adversarial regularization and a Bregman proximal point optimization. We will also compare the results with an off-the-shelf transformer. Our experiments show that the AEN and SMART models outperform an off-the-shelf transformer.

1 Introduction

Social networking cites like Twitter and Facebook are rapidly gaining popularity as they allow people to express their views about topics, have discussions or post messages across the world. This paper focuses mainly on sentiment analysis of twitter data which is helpful to analyze the information in the tweets where opinions are highly unstructured, heterogeneous and can display clear positive and negative sentiment. In this paper, we provide comparative analyses using existing techniques like SMART and AEN-BERT with evaluation metrics on twitter data streams. We have also discussed general challenges and applications of Sentiment Analysis on Twitter.

2 Related works

Numerous techniques have been proposed for the purpose of sentiment analysis. Recent times have shown a greater interest in neural network methods for the purpose of sentiment analysis. Recurrent Neural Networks (RNNs) with attention mechanisms are the most widely-used technique for the task of sentiment analysis [3]. The methods use attention to find the semantic relatedness between words, and then use the attention scores to find the aggregated contextual features for predictions [3].

In addition, the transformer models were proposed for neural machine translation. Due to their superior performance, a bidirectional transformer based language model known as BERT was proposed [4]. This model prompted many other works to further improve the pre-training of the model by introducing new unsupervised learning tasks [4].

In this paper, we propose an Attentional Encoder Network for the targeted sentiment classification task which employs BERT for the modeling between context and target. We also propose a SMART model which utilizes the Smoothness inducing Adversarial Regularization technique,

3 Method and algorithm

Implementing the algorithms requires several steps including tokenization, training and testing. The dataset we will be using is offered by Stanford and contains 1.6 million tweets labeled with a polarity of sentiment (0 = negative, 4 = positive).

Since twitter messages have a max of 140 characters, the vocabulary is very different from formal writing [1]. We will tokenize the sentences and compute vector word embeddings using BERT embeddings. These word embeddings will be the input to the AEN-Bert and SMART models.

The max length of an embedded tweet was assigned to 32, so if the tweet happens to be more than that, it would be cut out and if it is less than 32 then there would be padding token added to it. The start of sequence is [CLS] and end is [SEP]. The padding after would be denoted as [PAD] .

RNNs are models that involve sequences of data, such as sentences, that have internal memory. However, RNNs are difficult to parallelize and truncated backpropagation through time brings difficulty in remembering long-term patterns. To address this issue, this paper proposes an Attentional Encoder Network (AEN) - Bert which eschews recurrence and employs attention based encoders for the modeling between context and target [2].

$$Attention(\mathbf{k}, \mathbf{q}) = softmax(f_s(\mathbf{k}, \mathbf{q}))\mathbf{k} \tag{1}$$

We will also implement a SMART model, for robust and efficient fine-tuning of pre-trained language models [4]. The SMART framework uses pre-trained weights from BERT with two additional ingredients: Smoothness-inducing Adversarial Regularization and Bregman Proximal Point Optimization. We will use the Adam optimizer for training and tune hyper parameters to reach desired levels of accuracy and loss.

$$min_\theta \mathcal{F}(\theta) = \mathcal{L}(\theta) + \lambda_s \mathcal{R}_s(\theta) \tag{2}$$

```
for e = 1...EPOCHS:
   for input, targets in BATCH:

      logits = SentimentClassifier(inputs)
      loss = CrossEntropyLoss(logits, targets)

      ~inputs = Adversarial(inputs)
      ~logits = SentimentClassifier(~inputs)
      ~loss = CrossEntropyLoss( ~logits, targets)

      theta` -> AdamUpdate(loss + Lamda * ~loss)
      theta -> u * Bregman(theta`)
    end for
end for

return: theta
```

Figure 1: SMART Algorithm

The Smoothness-inducing Adversarial Regularization (2) adds a weight penalty in order to keep weights small. The penalty is calculated by computing the loss term $\mathcal{L}(\theta)$ plus a regularizer. The regularizer is calculated by computing the loss between the natural inputs and an adversarial input, which is the natural input slightly modified. $\lambda_s > 0$ is a tuning parameter, and $\mathcal{R}_s(\theta)$ is the smoothness-inducing adversarial regularizer. Here we define $\mathcal{R}_s(\theta)$ as

$$R_s(\theta) = \frac{1}{n} \sum_{i=1}^{n} \max_{||\tilde{x_i} - x_i||_p \leq \epsilon} l_s(f(\tilde{x_i}; \theta), f(x_i; \theta)) \tag{3}$$

4 Experiments and Discussions

We demonstrate the effectiveness of sentiment classification for the AEN, SMART in comparison to an off the shelf transformer[7].

The same training data was used to train all models. We used 100,000 labeled tweets with a 0.7, 0.15, 0.15 train, validation and test split. Each tweet was labelled with a 0 or 1, indication positive or negative sentiment. All tweets were embedded using BERT embedding and given a max length of 32 tokens.

4.1 AEN

The AEN model used Adam optimizer with a learning rate of 1e-5 and batch size 128. The number of epochs was set to 3.

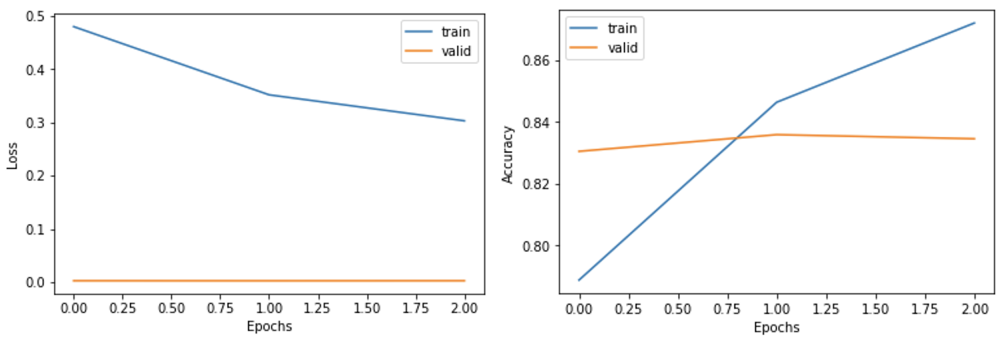

Figure 2: AEN train accuracy and validation accuracy

Over the 3 epochs of training, the AEN model achieved a training accuracy of 87%, and validation accuracy of 83%. The test accuracy was 83.8%. The model appears to be leaning towards overfitting, however, stopping at 3 epochs proved to be useful as the validation accuracy converged very earl

4.2 SMART

Our implementation of SMART is based on BERT [5], using their pre-trained weights. We followed specifications for hyper parameters described in the SMART paper, using ADAM [6] for the optimizer with a learning rate of 1e-5 and batch size 128. A larger batch then specified was used in order to speed up training. The number of epochs was set to 3. A dropout layer of 0.1 was added before the classification layer, and gradients were clipped to a max of 1.0. For SMART specific hyper parameters, we used $\epsilon = 1e-5, \mu = 1, \lambda = 3, \beta = 0.99$

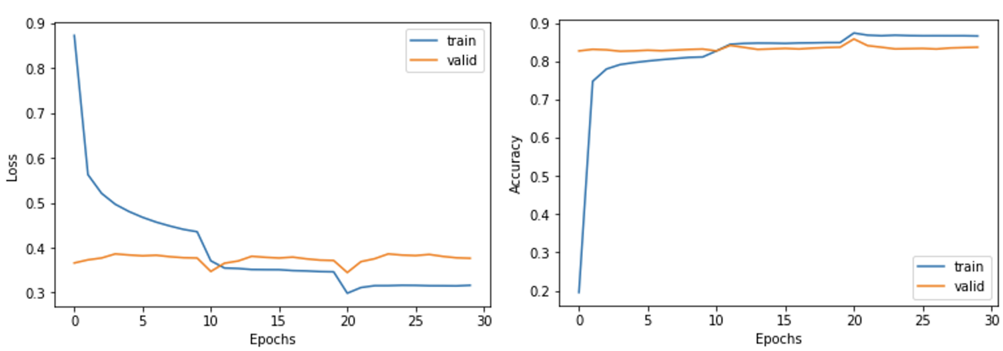

Figure 3: SMART train loss and validation loss

Over the 3 epochs of training, the SMART model achieved a training accuracy of 85%, and validation accuracy of 83%. The test accuracy was 83.5%.

In order to view the effects of SMART, a non-SMART model was trained and tested to compare the effectiveness of SMART.

Over the 3 epochs of training, the non-SMART model achieved a training accuracy of 92%, and validation accuracy of 81%. The test accuracy was 83.2%.

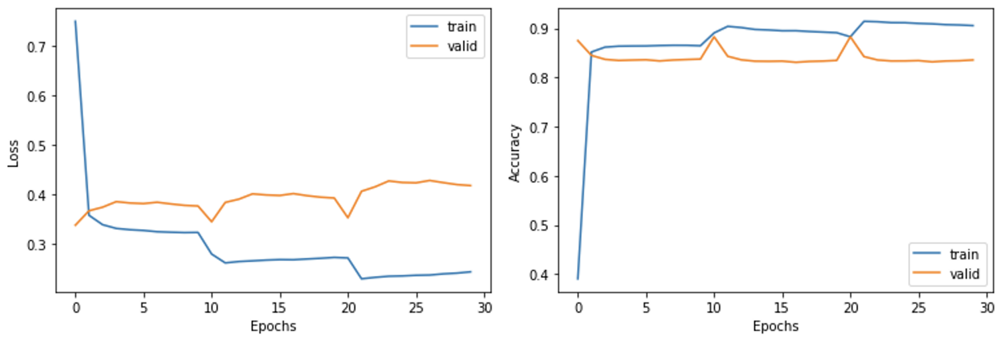

Figure 4: NON SMART train loss and validation loss

In the non-SMART model, the training accuracy is significantly bigger than the validation accuracy, showing that the model is over fitting. This also results in a slightly lower test accuracy, as the model is worse at generalizing. The SMART techniques cause for a reduction in the training accuracy and an increase in validation accuracy. By adding a weight penalty with an adversarial regularizer and using Bregman point optimization, the SMART model is able to reduce over fitting and achieve a better test accuracy.

4.3 Control

A control was used in order to compare the effusiveness of both models we implemented with an off the shelf available transformer. The transformer used was provided by Hugginface [7], using their sentiment classifier. We tested the same test data set on the control, which outputs a label in {Positive, Negative} and a confidence between {0-1}.

The off the shelf transformer achieved a test accuracy of 70%. This score is lower than expected, and is most likely due to the fact that twitter messages significantly differ from regular text. This control model was trained on regular text, so twitter data proves to be a more difficult problem for this model.

5 Summary

The demand of sentiment analysis is increasing due to the requirement of analyzing and structuring hidden information extracted from social media. This paper made use of AEN-BERT and SMART model to solve the variety of problems in this paper, which have achieved high accuracy in many related studies. Moreover, in this paper we focused on Twitter messages and implemented a machine learning model for sentiment analysis using those models and thereby showed its results. We found that AEN model performed better than the SMART model.

There are also some significant limitations of this method which makes twitter sentiment analysis a challenge. People sometimes express their views using multiple tweets, however the model does not take multiple twitter threads into account. Additionally, words or phrases can have different meanings based on context, which may not be well understood by the model. Finally, these models require large data sets and are tremendously costly to train. With better processing power, our models would have achieved better generalization and more accurate results

6 References

[1] Y. Yuan and Y. Zhou, *Twitter Sentiment Analysis with Recursive Neural Networks*, https://cs224d.stanford.edu/, 2015. [Online] Available: https://cs224d.stanford.edu/reports/YuanYe.pdf.

[2] [4]Y. Song, J. Wang, T. Jiang, Z. Liu and Y. Rao, "Attentional Encoder Network for Targeted Sentiment Classification", arXiv preprint arXiv:1902.09314, 2019 [Online]. Available: https://arxiv.org/pdf/1902.09314.pdf. [Accessed: 20- Apr- 2021]

[3] X. Li, L. Bing, W. Lam and B. Shi, "Transformation Networks for Target-Oriented Sentiment Classification", Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018 [Online]. Available: https://arxiv.org/pdf/1805.01086.pdf. [Accessed: 20- Apr- 2021]

[4] H. Jiang, P. He, W. Chen, X. Liu, J. Gao and T. Zhao, "SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization", Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020 [Online]. Available: https://arxiv.org/pdf/1911.03437.pdf. [Accessed: 20- Apr- 2021]

[5] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. Availiable: https://arxiv.org/pdf/1412.6980.pdf

[6] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Availiable: https://arxiv.org/pdf/1810.04805.pdf

[7] “Transformers,” Transformers - transformers 4.5.0.dev0 documentation. [Online]. Available: https://huggingface.co/transformers/.