A Performance Comparison of Latest Models for Twitter Sentiment Analysis

Xue Yuanfeng,CSEE,University of Essex, yx20778@essex.ac.uk

Abstract—Natural language processing (NLP) is a prevalent technology in text mining, mainly applied to analyze conversations, opinions, and opinion sharing to determine business strategy, political commentary, and evaluate public action. As data science and technology keep unfolding, an increasing number of models that appear later than BERT (Bidirectional Encoder Representations from Transformers) are available to be applied on Twitter sentimental analysis. The latest pre-trained models trained on massive data can achieve a higher score on text classification than traditional models, with more precise predictions. This paper aims to find the advantages and limitations of the latest text classification models and provides a new benchmark after estimating the performances on the datasets (hate, irony, and offensive). Also, the result of this project can be compared with the standard evaluation framework provided by Cardiff University. Finally, we involve other transformed models (GPT-2 and XLNet) into comparison and conclude their benchmarks.

Index Terms—Sentiment Analysis, BERT, GPT-2, XLNet Tensorflow, Keras

1 Introduction

This paper is intended to exploit the latest models for analyzing Twitter users' sentiment on datasets from TweetEval[1]. Although convolutional neural networks and recurrent neural networks have proven to be considerably effective for various natural language processing, there are a number of problems with them. For example, data can only be read in a sequential manner in one direction. Recently, latest transformed models, such as BERT, GPT-2 and XLNET, can be imported as an additional output layer to achieve a more precise prediction of the natural language. This is because these models are trained on a vast dataset, they can output an amount of unlabeled text data with different weightage for further processing. This project will focus on comparing performance of the latest text classification models. This project uses three datasets from TweetEval, including irony detection [8], hate speech detection[2], and offensive language identification[10]. The ironic tweets were collected using irony-related comments (i.e., irony, not irony), Hate Speech Detection was one of the leading binary sub-tasks for detecting the presence of hate speech (hateful, not hateful). Offensive language identification is a project devoted to distinguishing the discriminate between offensive and non-offensive posts.

2 Literature Review

(i.) BERT (Bidirectional Encoder Representations from Transformers) is published by researchers at Google AI Language [4] , which combines previous algorithm and experience on natural language processing and thus obtains new state-of-the-art. Because of their contributions, we can fine-tuning model based on a pre-trianed mode to improve the precision of prediction.
(ii.) Twitter as a Corpus for Sentiment Analysis and Opinion Mining[5], A.Pak and P.Paroubek have tried SVM and Naive Bayes to obtain up to 70% of an accuracy on the test set and compared their performance with N-gram. However, with the release of BERT, N-gram is no longer popular, because transformed models pre-trained from a large amount of samples can achieve better predictions through generating sentence embeddings.
(iii.) A practical guide to sentiment analysis[3].This book discuss the latest developments in sentiment analysis. The authors gave us the best practice to classify sentiments at the sentence level and then investigated several strategies to estimate the sentiment.
(iv.) Fine-Tuning BERT for Multi-Label Sentiment Analysis in Unbalanced Code-Switching Text [6]. This work is proposed to fine-tune BERT models in the code-switching text that contains both Chinese and English, which gives us a practicable reference for optimizing a model.
(v.) TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification [1].The project is created by Barbieri et al. at 2020, the TweetEval TweetEval consists of seven tasks in Twitter, which provides a set of baselines as well as shows the effectiveness of starting off with existing pre-trained generic language models, but it has not given the benchmark for GPT-2 and XLNet .
(vi.) Word2vec [9] is a technique for natural language processing, word2vec represents each distinct word with a particular list of numbers called a vector. . In this paper, we will look a sentence as an entire vector rather than analyzing words separately. (vii.) Well-Read Students Learn Better: On the Importance of Pre-training Compact Models [7] this paper presents the performance of the Pre-trained BERT trained on different training contents.

3 Methodology

3.1 Data Pre-processing

Pre-proecessing is a series of steps for building a pipeline that converts text to numbers because the classifiers can only work with digital forms.

3.1.1 Normalization

Text normalization is the process of transforming a text into a standard form. It involves following steps: (i.). transform all upper case letters to lower case
(ii.). remove special characters and eliminate unwanted characters such as HTML tags, punctuation marks
(iii.). remove special characters, white spaces and symbols
(iv.). remove stopwords like: that, that'll, these, am, is, are, was, were, be, been, etc.,

Example: "@user @user real talk do you have eyes or were they gouged out by a rapefugee? " will be changed to "real talk eyes gouged rapefugee"

3.1.2 Tokenization And Word Embedding

Tokens are words separated by spaces in a text. Transformed model encoding differs from traditional vector approaches as they provide contextualized embedding approaches that will transform words into a sentence based on their vocabulary. If the word does not exist in their vocabulary, they can separate words into several sub words. During the tokenization process, tokenized words are converted to numbers. As a result, the process of Word Embedding determines the actual "vocabulary" of the training dataset because the machine learning algorithm works with numbers rather than text.

3.2 Training and Classification

After downloading the TweetEval project from GitHub, the datasets (hate, irony, offensive) are transfer into data frames using pandas. Afterward, The algorithm can be learning the relationship between texts and labels using training and validating dataframes. Once the model is built, it can predict the unknown sentiment for the text at the test dataframes.

3.2.1 Evaluating Model Performance

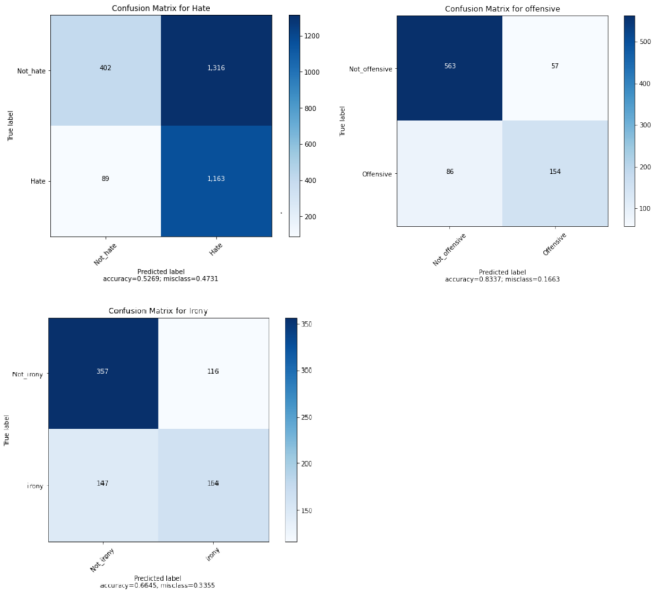
At this stage, the predicted sentiment is then compared with the actual sentiment in the test set. We can use confusion matrix, accuracy, recall and F1 measures to evaluate the model performance. To get a quick result as reference, a initial text classification model "BERT Classifier" is created as a baseline model in order to validate the effectiveness of latest model with these datasets. Also, the benchmark given by TweetEval [1] will be taken as reference.

3.2.2 Classification

A pre-trained BERT model that was trained on a vast dataset is performed here as a baseline. In this project, the pre-trained model "bert_en_uncased" is imported as a deep neural network layer. Fitting the training datasets and validating datasets, we can have a new model to predict labels in the test datasets.

4 Results

According to the result, the pre-trained BERT model achieve 83.4% on 'offensive', and 66.5% on the 'irony' datasets, whereas the result on 'hate' is not satisfied, with only 52.6%. The confusion plots demonstrate that most of labels are correctly predicted on offensive which is better than that on irony.
Noticeably, 1316 'not hate' comments are mis-classified into 'hate', however it works quite good on 'hate', only 89 'hate' comments are wrongly classified.



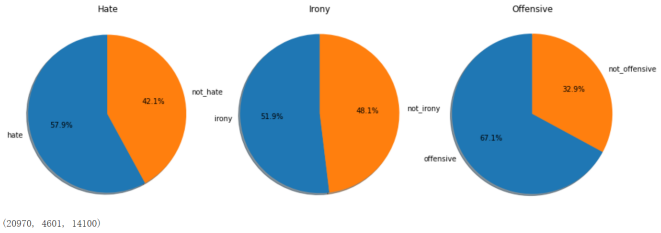
Compared to the models given by TweetEval, we have a relatively better evaluation.

Accuracy Reference List			
Model	Hate	Irony	Offensive
BERT*	52.6	66.5	83.4
LSTM	52.6	62.8	71.7
SVM	36.7	61.7	52.3

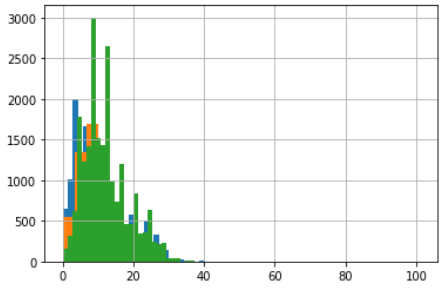
Generating wordcloud plot can visualize the frequency of the words so that to help us analyse the characters of the words.

As we can see from the figure, the most popular

words for hate and not-hate are ”refugee” and ”bitch” respectively, while ”liberal” is the most popular words for both offensive and non-offensive with respect to the word frequency in offensive datasets. Likewise, ”love” is the word which has the highest frequency in both sides of irony. Overall, the graph indicates that individual words cannot represent the sentimental analysis appropriately. In this paper, transformed models are applied since they use word embedding, which can separate words and analyze similarity comprehensively.



The pie charts compare the proportion of sentimental samples in the datasets. The dataset of hate contains the largest number of samples(20970), followed by datasets of irony and Offensive, with 14100 and 4601 respectively. The data distribution of irony is comparatively balanced, 51.9% of comments is 'hate' and 48.1% of them is 'not hate'. By contrast, datasets of hate (%57.9 hate and 42.1% not hate) and offensive (67.1% offensive and 32.9%) are imbalanced.



The histogram shows the distribution regarding to the length of sentences. Most of the sentences in the datasets have around 15 words, while very few sentences have more than 30 words.

5 Discussion

When it comes to NLP-Based sentiment analysis, it is indispensable to convert text to numbers in order to apply different types of classifiers since the machine learning and deep learning work with numbers. Thus, the natural language process heavily relies on data Pre-processing like normalization, tokenization, word embedding. These operations are inevitable to be performed before applying the algorithm on sentimental analysis. According to my theoretical knowledge, a lot of common word embedding approaches, including word bag, N-Grams, TFIDF Approach, Word2Vec, the similar disadvantage of them is context-independent. In other words, there is just one vector representation for each word in these methods. Different senses of the word combined into one single vector is not conducive to be recognized by the computer. By contrast, the embedding measures allow us to have multiple vector, overcoming the shortcomings of the historical method of vectorization.
Text classification is one of the crucial and typical tasks of sentimental analysis. Obviously, machine-based classifiers work far more better than rule-based classifiers, BERT, GPT-2 and XLNet are the most popular pre-trained model amongst all deep learning models. All of them are trained on a massive amount of data. GPT-2 relies on transformer decoder blocks, while BERT uses transformer encoder blocks. Similarly, XLNet is a larger bidirectional transformer that may achieve better than BERT prediction. Owing to the time limitation, I have to stop on BERT and am going to continue my research in the coming weeks.

6 Conclusion

This paper compares the performance of the latest text classification models on Twitter sentimental analysis. By applying the latest deep learning models (BERT, GPT-2 and XLNet), predictions can be more precise than the traditional models such as SVM. Besides, Pre-processing with different tokenizations can also affect the result of classification. With the enhancement of computing capacity, adding huge amounts of samples into a more complex model for training will significantly improve the accuracy of predictions in the future.

References

[1] Francesco Barbieri et al. “TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification”. In: *Proceedings of Findings of EMNLP*. 2020.

[2] Valerio Basile et al. “SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter”. In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, pp. 54–63. DOI: 10.18653/v1/S19-2007. URL: https://www.aclweb.org/anthology/S19-2007.

[3] Erik Cambria et al. *A practical guide to sentiment analysis*. Springer, 2017.

[4] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[5] A. Pak and P. Paroubek. “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”. In: *LREC*. 2010.

[6] T. Tang, X. Tang, and T. Yuan. “Fine-Tuning BERT for Multi-Label Sentiment Analysis in Unbalanced Code-Switching Text.” In: *IEEE Access, Access, IEEE* (2020), pp. 193248–193256.

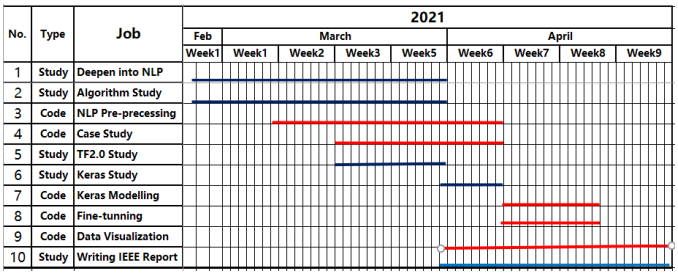
[7] Iulia Turc et al. *Well-Read Students Learn Better: On the Importance of Pre-training Compact Models*. 2019. arXiv: 1908.08962 [cs.CL].

[8] Cynthia Van Hee, Els Lefever, and Véronique Hoste. “Semeval-2018 task 3: Irony detection in english tweets”. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. 2018, pp. 39–50.

[9] Wikipedia contributors. *Word2vec — Wikipedia, The Free Encyclopedia*. [Online; accessed 6-February-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Word2vec&oldid=1001737145.

[10] Marcos Zampieri et al. “SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval)”. In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019, pp. 75–86.

7 PLAN



Mastering the neutral language process requires aspiration and time on cycling between theory, practice and assignments. According to the previous exploration, rule-based approaches, for instance, comparing the number of positive and negative words to estimate the Twitter sentiment, may not be reliable for predicting a sentimental classification. Hence, my further research will focus on deep learning, prioritizing BERT, GPT-2, XLNET amongst all supervised learning models. I have already outlined a roadmap to explore the NLP world’s principal rather than completing the assignment with ”black boxes”(library).

Deep into NLP (week1-week4): It is impossible to be an expert of NLP through reading a couple of blogs or theses. I will follow several books to enhance my fundamental understanding of natural language processing.

Algorithm Study (week1-week4): Algorithms can be considered as the vital building blocks for Machine Learning. Realizing the computational process is an indispensable way to improve the accuracy of results. Only in this way can I achieve an optimal model to unlocking the value of machine learning. I will root into the mathematics behind the machine learning models (BERT, GPT-2, XLNET).

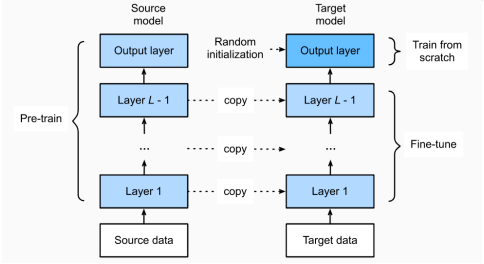
NLP Pre-processing (week2 - week4): Once I have learned more NLP concepts, I should embark on making a comprehensive comparison of pre-processing collocations to improve the precision without modifying the model during this period.

Case Study(week3 - week6): The most convenient way to acquire NLP knowledge is to study and conclude someone else’s code,learning by doing and learning by imitating.

TF2.0 Study(week3 - week 6): Tensorflow is an open-source artificial intelligence library, which can optimize a range of machine learning models by tuning parameters and parsing results. From the study, I expect to know how to build a neural network and how to train, evaluate and optimize it with hyperparameter.

Keras Study and Keras Modelling (week6 and week7-week8): Keras follows best practices for reducing cognitive load for human beings. In this time frame, I will practice a range of deep learning algorithms(BERT, GPT2, XLNET) and finally compare the performance of all models. Hopefully, I can find a better model with higher accuracy based on my knowledge.

Fine-tunning (week7-week8): Besides creating a new model, fine-tunning is another approach to create a new model by selecting part of layers from a pre-trained model and adding several layers into it. Fine-tunning can be a feasible approach for optimizing the original model if we can find normalities from the classified data.



Visualization(week6-week9): A good data visualization not only helps me find patterns and relationships from data but also gives a clear and convincing voice to the audience. Being familiar with presenting data is conducive to concluding my report and explaining the details of the data.

Writing IEEE Report(week6-week9): I will perfect my report by referring to your suggestions and my findings.My next article will focus on the comparison of the latest models for twitter sentimental analysis.