Insincere and Toxic Question Detection

Course project for CSE 6240: Web Search and Text Mining, Spring 2020

Aaron Reich
Georgia Institute of Technology
areich8@gatech.edu

Kavin Krishnan
Georgia Institute of Technology
kkrishnan9@gatech.edu

Thomas Hu
Georgia Institute of Technology
thu71@gatech.edu

ABSTRACT

Recently toxic content classification has become a very important problem for major websites such as Quora, Reddit, etc. which feature public forums for discussion. One significant area in toxic content classification is attempting to predict whether questions asked by users are sincere or not. This is specifically referring to if the question is trying to make a statement rather than actually looking for useful answers from other users. In this project, we are trying to predict the intention of questions asked on Quora as being either sincere or insincere. We have trained two models to serve as baselines. The first of which is an Attention-based Bi-LSTM Recurrent Neural Network. The other baseline model is a Support Vector Machine utilizing a Linear Kernel. Both model use a pre-trained GloVe embeddings. Bi-LSTM outperformed the SVM model in terms of F1-scoring and precision whereas the SVM model performed better in terms of recall. With the completion of these two baseline models, we proceeded to evaluate our proposed which utilized BERT and DistilBert, and they featured improved performance in the areas of precision and recall.

1 INTRODUCTION

Toxic content classification is a crucial task for major platforms such as Twitter, Youtube, and Reddit which grant users the ability to publicly post content. One unique area in toxic content classification is the issue of predicting whether posts from users are sincere or not. This particular dilemma is faced by websites such as the question and answer platform Quora. For example, some users ask questions which have the primary motivation to communicate an opinion such as: Why is it that European food is so terrible? or Why are Quora employees friendly and helpful?

This task of identifying the sincerity of questions was the focus of this project. This problem is more complex than usual toxic content classification where users are putting a statement or response to others. In this case, all the content is question based and it can be hard to distinguish between sincere and insincere questions because using a hateful word does not necessarily correlate to it being a insincere question. A insincere question can have one or more of the following characteristics: having a non-neutral tone, is disparaging or inflammatory, not grounded in reality, not trying to seek genuine answers so on and so forth. To perform our experiments, we utilized a dataset available in a challenge on Kaggle.com hosted by Quora. The dataset consisted of over 1.3 million questions either labelled insincere or not sincere. We split this dataset appropriately for testing and training our experimental models. Our approach to complete our objective was to first train an SVM and Bi-LSTM model with GloVe Embeddings to serve as a baseline to compare with our main approach which was to train a BERT and DistilBERT model. The GloVe embeddings used by the baseline models were produced though a context-free model, it solely generating a word embedding for each word in the vocabulary. This limited the amount of information that could be encoded compared to BERT which generates a representation for each word based on the other words found in each sentence. Due to this notion, it was believed that BERT and DisilBERT would allow for a significant improvement in performance. Both of these models in our experiments came to boast better results in terms of precision and F1-Score than the baseline models in both validation and testing. However, the baselines performed better in recall on both the validation and test set.

2 LITERATURE SURVEY

Support Vector Machine was used for finding the maximum-margin hyperplane in order to maximize the distance between different classes [13]. SVM is one of the traditional machine learning algorithms for toxic content classification. They used unigrams and bigrams as features and weighted by Term Inverse Document Frequency. Support Vector Machine is suitable for the toxic content classification problem due to its ability to work well in classification problems because of its ability to handle a large number of features and because it stays robust even in the case when sparse examples are provided to it to learn from [12]. Zhao et al. [13] then used more recent deep learning techniques such as CNN's and LSTM's for the problem. They did not use pre-trained word embeddings for the problem and instead had an embedding layer for both the CNN and LSTM. This is because they did not want outside influences to potentially impact the results. They used the Wikipedia Talk Page dataset. LSTM is shown to outperfrom SVM in terms of F1 score.

Gao et al. [4] used a bi-LSTM with attention model for detecting online hate speech detection in a Fox News user comments dataset. They used pretrained word2vec embeddings and fed the embeddings as a sequnece to the bi-LSTM with attention model. They showed that the addition of the attention mechanism as well as having a bi-directional LSTM compared a regular LSTM increased the AUC score by 5.7%. The attention mechanism showed usefulness in helping to incorporate context into the prediction when deadline with long text. It was also shown to have the ability to identify small regions of hate speech within comments as well as implicit hate speech. Yao et al. [10] used a RNN architecture in its Attention-based Bi-LSTM neural networks introduced for sentiment classification of short texts. The LSTM overcomes the problem of the exploding gradient in RNN's while at the same time learning sentence representations of any length and dependency. They evaluated its performance on the Stanford Sentiment Treebank dataset and a movie review dataset.

While reviewing the latest advances in the field of Natural Language Processing, we reviewed the BERT Model. BERT has shown significant improvement over existing architectures on sequential classification tasks, obtaining state-of-the-art results on the GLUE benchmark [2]. We also reviewed DistilBERT which is a distilled version of BERT. It was shown to reduce the size of the BERT model by 40%, speed it up by 60% and while still having 97% of BERT's language understanding capabilities [8].

3 DATASET DESCRIPTION AND ANALYSIS

3.1 Data preparation

The data-set comes from the "Insincere Questions Classification" challenge[1] on Kaggle hosted by Quora. The company provided a standard Comma-separated value file for participants to derive a model of predicting the sincerity of questions. The file simply contained over 1.3 million questions labeled insincere or sincere and required minimal cleaning because all the questions in the dataset are labeled questions.

We utilized a pretrained set of GLoVe embeddings from Stanford University. We used the "Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download" GLoVe embedding [2]. The embeddings were mostly comprehensive in including language elements. In order to see how well the GLoVe embedding will fit our vocabulary, we did a coverage test. Without processing the data-set at first and using a word tokenizer, only 63.75% of the data-set vocabulary is covered by the embedding which is 99.04% of all the text. The embedding seemed to be missing embeddings for contractions, uncommon punctuation, and mathematical symbols. As a result we modified the question text in our data set by changing the unknown punctuation, contractions such as "ain't" to "is not", "aren't" to "are not" so on and so forth. But the embeddings are case sensitive and contain also the usual punctuation so we decided to not change them to lowercase and not take out the punctuation in the dataset as we wanted to keep as much information as possible. The resulting cleaned data-set has a new coverage of 74.74% of the vocabulary for 99.58% of all text.

3.2 Raw Data Statistics

There are 1,306,122 Quora questions in the dataset. The dataset has a ground truth labeling consisting of 1,225,312 Quora questions with a label of 0 (not insincere) and 80,810 Quora questions with a label of 1 (insincere) as shown in the Figure 1. Five distinct features of the data set include the vocabulary size, statistics of the number of sentences in each Quora question, statistics of the number of tokens per sentence, statistics of the number of tokens in each Quora question by class, and the distribution of the number of Quora questions by class. The vocabulary size is 508,823 tokens.

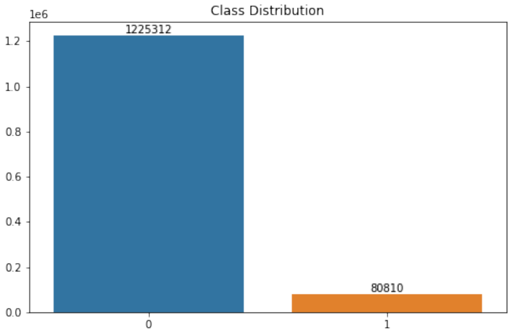

Figure 1: Class distribution

The maximum number of sentences in a Quora question is 11 and the minimum number of sentences in a Quora question is 1. The average number of sentences in a Quora question is 1.1385. The median number of sentences in a Quora question is 1. The maximum number of tokens per sentence in a Quora question is 207 and the minimum number of tokens per sentence in a Quora question is 1. The average number of tokens per sentence in a Quora question is 12.6894. The median number of tokens per sentence in a Quora question is 11.

As can be observed in Table 1, the mean, median, and max number of tokens in the Quora questions is larger for the Insincere question class than it is for the Sincere question class. As can be observed in Figure 2, the center of the Sincere Question Length distribution makes up roughly 12% of the

[1] https://www.kaggle.com/c/quora-insincere-questions-classification
[2] https://nlp.stanford.edu/projects/glove/

Class	Mean	Median	Max	Min
Sincere	14.1079	12.0	182.0	3.0
Insincere	19.5855	17.0	412.0	1.0

Table 1: Number of Tokens by Class

Sincere data points compared to the center of the Insincere Question Length distribution making up roughly 4.5% of the Insincere data points. The max number of tokens for both classes are clearly outliers illustrated in the figure.

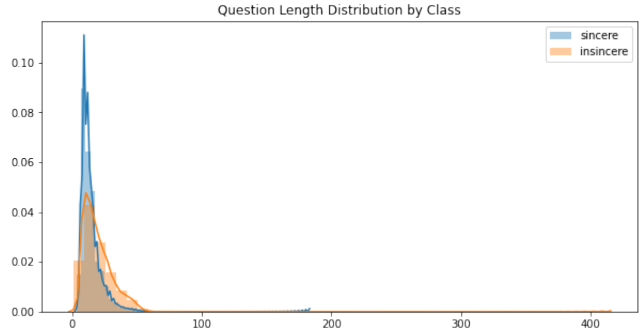

Figure 2: Question length by class

3.3 Data Analysis

Further analysis was performed on the data set through inspecting the frequency of N-grams within questions. Figures 3 through 5 feature graphs displaying the 20 most frequent sincere and insincere N-grams of sizes 1 through 3. It was demonstrated that the non-singular n-grams analysis tended to provide more informative results on nature of vocabulary found in sincere and insincere questions. The top 20 n-grams from insincere questions seem to contain words that are proper nouns and words relating to race, gender, and sex. These phrases clearly share similarities of politicized and controversial themes. On the other hand, we can see the scope of the n-grams for sincere questions to be more broad and more complex to classify/categorize.

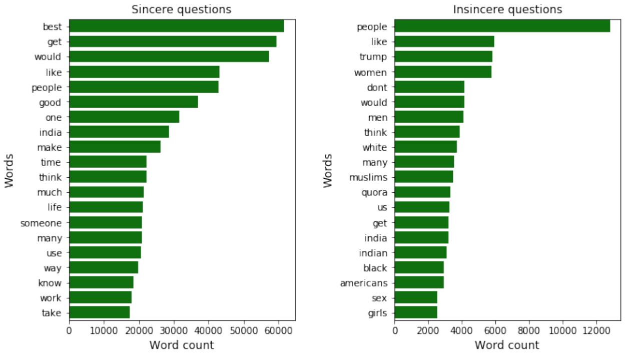

Figure 3: n-gram of size 1

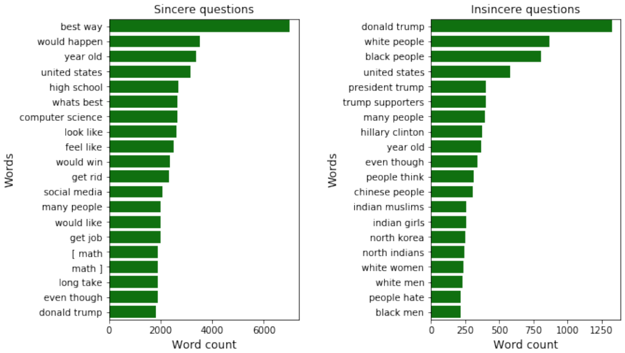

Figure 4: n-gram of size 2

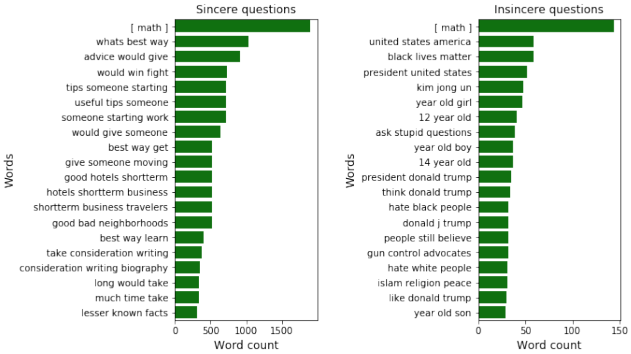

Figure 5: n-gram of size 3

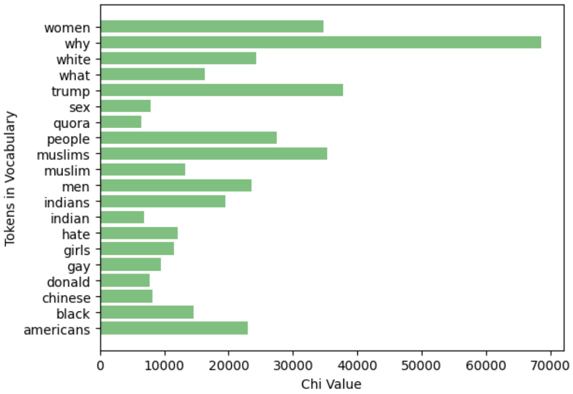

Figure 6: Chi Squared Statistic for Tokens

For computing the Chi-Squared statistics for the tokens, we first constructed the Document-term frequency matrix and then computed the Chi-Squared statistics of it returning the tokens with the top 20 Chi- Squared values. The higher the Chi-Squared values the greater the likelihood that the class (sincere or insincere question) and the feature (token) are dependent on each other. It is shown in Figure 6 that tokens relating to gender, ethnicity, or religion contain high Ch-Squared values indicating that there could be a dependency between if a question contains tokens related to those categories and a question being insincere. Another insight is that 690 sincere questions contain a math equation while only 45 insincere questions do with a 0.9388 sincere ratio. This could also be interpreted as a good feature for indicating if a question is sincere or insincere.

4 EXPERIMENT SETTINGS AND BASELINES

4.1 Experiment Settings

The dataset was split 75% for the training set and 25 % for the test set. The experiments were performed on Colab using GPU's and 38GB of RAM. For the Support Vector Machine baseline experiment 10-fold cross validation was used. Cross Validation was performed by calling the GridSearchCV method in the "`sklearn.model_selection`" library. For Bi-LSTM, a Validation Set with a size of 10% of the training set was used. The evaluation metrics used for analyzing the performance of the models is F1 Score, Precision, and Recall. F1 Score is the weighted average of Precision and Recall. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to the all observations in actual class. In our problem the positive class is the insincere questions.

4.2 Baseline Descriptions

The state of the art method for toxic content detection is found in Gao et al. [4] which shows a Bi-LSTM with attention model being effective for online hate speech detection. It was shown to have the ability to identify small regions of hate speech within comments as well as implicit hate speech [4]. We will design our first baseline with this Bi-LSTM architure. Also it can observed in Yao et al. [10] the use of a RNN architecture composed of an Attention-based Bi-LSTM neural network introduced for sentiment classification of short texts. The model consists of an encoder layer using LSTM, attention-based mechanism layer, and a softmax layer[1]. This model in Yao et al.[10] is used for sentiment classification of short text which is a similar problem to ours.

The input of the encoder layer are the word embeddings using Glove. The encoder is a bidirectional LSTM that contains two independent LSTM's, one going from the beginning of the sentence to the end and the other one in the opposite direction. For each token, the resulting hidden state of the Bi-LSTM model is the concatenation of the hidden states of the two networks. The two LSTM neural network parameters in the Bi-LSTM networks are independent of each other, and they share the same word embeddings (input) of the tokens in the sentence. We are using the bidirectional LSTM because the whole sentence is known as the input and we are not trying to predict the next token given past the tokens.

The attention layer is used to compute a context vector in a sentence. The output of the encoder layer is a hidden state h_t for each token. The hidden state is used to compute a weighted mean of the state C (context vector) using a learned hidden representation u_w of the attention layer. Finally the softmax layer is used to get the correct classification by feeding the context vector from the attention layer. Here the softmax layer is a logit function because we only have two classes.

The base code was taken from a Kaggle notebook[3]. We used the Binary Cross Entropy loss between target and output logits to train our model, the optimizer is Adam optimizer with a learning rate of 0.0001 for the first 20 epoch and then 0.000001 for the last 10 epochs. After training of the model we performed a search for the classification threshold on the validation set. We found that with a classification threshold of 0.28, it gives the best F1-score.

The second baseline we used is Support Vector Machine with a Linear Kernel. The implementation of SVM we used was the svm.LinearSVC method from the sklearn library[4]. The default "`squared_hinge`" hyperparameter was used for specifying the loss function to be used by SVM. The "`class_weight`" hyperparameter was set to "balanced" in order to adjust the decision hyperplane to compensate for the class imbalance. The "balanced" setting allows for the weights to be inversely proportional to class frequencies. The C regularization hyperparameter was tuned during 10-fold cross validation by grid searching through these values for C: 0.03162278, 0.08483429, 0.22758459, 0.61054023, 1, 1.63789371, 4.39397056, 11.78768635, 31.6227766.

We ran KMeans Clustering in order to cluster the word vectors generated from Glove. KMeans partitions data into K clusters where each data point is assigned a cluster based on its nearest mean or centroid. For the representation of each Quora question, we used a Bag-of-Centroids approach in order to use semantically related clusters rather than individual words like Bag-of-Words. For each cluster we found the amount of times the tokens in each Quora question belonged to the specific cluster. K number of clusters was set to 750 in order to best represent the wide scope of semantic meaning embedded in the dataset. It can be observed in [13] that SVM has a higher F1 score than LSTM when detecting "Threat" and "Identity hate" classes. SVM also outperforms LSTM and CNN in terms of precision. It's ability to remain robust with few training examples can be observed in the provided learning curve where LSTM only starts to outperform SVM when more than 25% of the dataset is used in training [13].

[3] https://www.kaggle.com/robertke94/pytorch-bi-lstm-attention

[4] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

5 PROPOSED METHOD

Since the GloVe embeddings that Bi-LSTM is using for the classification are produced by a context-free model only generating one word embedding for every word that is in the vocabulary, it is limiting the amount of information that can be encoded in the embeddings. BERT on the other hand overcomes this shortcoming by generating a representation of every word based on other words that are found in each sentence. It at the same time overcomes shortcomings of other contextual models that are unidirectional or shallow bidirectional such as ELMo by being deeply bi-directional incorporating both left and right contexts of every sentence in its word representations [5]. Another shortcoming of the Bi-LSTM with GloVe embeddings baseline is that Bi-LSTM's dependence on sequential information flow leads to a decrease in the ability to capture long-range dependencies resulting in a decrease in encoding performance for sentences with a longer length [11]. BERT however uses a masked language model pre-training objective being to predict the original vocabulary ID of the word that is masked by its context allowing for a representation resulting from a combined left and right context. This allows for the pretraining of a deep bidirectional transformer overcoming the limitations of the Bi-LSTM [2].

Wang et al. [6] included a comparison of BERT and Bi-LSTM on the context based microblog sentiment classification problem. When comparing with no included context, BERT outperforms Bi-LSTM with a 0.092 increase in Macro F1 score. So in addition to proposing the use of BERT to approach this problem, we will also propose the exploration of the results brought on by the usage of DistilBERT, a distilled version of BERT. Due to its size reduction of 40% and 60% speed up it may be very useful in circumstances of toxic question or toxic content classification where training time and model size are constrained [8].

We will also analyze the effect of keeping the text cased as well as lowercasing all text to see the effects on the model performances. All of the BERT Models we will use are of the type BertForSequenceClassification [3]. This means that it consists of a BERT Model transformer with a sequence classification head on top. We will use a BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters model configuration as well as a BERT-Base, Cased: 12-layer, 768-hidden, 12-heads , 110M parameters model configuration [5]. We will run these models for 2 epochs. For DistilBERT we will use distilbert-base-cased: 6 layers, 768 dimension and 12 heads, totaling 65M parameters as well as distilbert-base-uncased: 6 layers, 768 dimension and 12 heads, totaling 66M parameters [9]. Due to DistilBERT's 60% speed up, we will experiment with running it for 6 epochs. For all BERT model configurations, a Validation Set with a size of 10% of the training set will be used.

6 EXPERIMENTS

Both Bert and DistilBERT models were imported from the Huggingface Library [3].They were fine-tuned on Google Colab using the same GPU and RAM settings as for the baseline models. We used a max token length of 512. The same training set and validation set was used as for Bi-LSTM. We used the Weight decay adam optimizer with learning rate of $2 * 10^{-5}$. The results of the two baseline models and our proposed method for the problem are in Table 2 and Table 3.

Table 2: Baseline and Proposed Method Results on Validation Set

Model	Cased Or Uncased	Precision	Recall	F1 Score
SVM	Cased	0.3134	0.7217	0.4373
Bi-LSTM	Uncased	0.6282	**0.7524**	0.6847
	Cased	0.6269	0.7178	0.6693
DistilBERT	Uncased	**0.7325**	0.6976	**0.7146**
	Cased	0.7264	0.6143	0.6657
BERT	Uncased	0.6849	0.6905	0.6877
	Cased	0.7314	0.6068	0.6633

Table 3: Baseline and Proposed Method Results on Test Set

Model	Cased Or Uncased	Precision	Recall	F1 Score
SVM	Cased	0.3130	**0.7470**	0.4412
Bi-LSTM	Uncased	0.6152	0.7467	0.6746
	Cased	0.6235	0.7313	0.6731
DistilBERT	Uncased	0.7173	0.6745	**0.6953**
	Cased	0.7214	0.6092	0.6606
BERT	Uncased	0.6897	0.6782	0.6839
	Cased	**0.7272**	0.6176	0.6679

Bi-LSTM achieved a higher precision and F1 score than SVM but SVM scores a little better than Bi-LSTM on recall. Because our comparison of SVM and Bi-LSTM is based on the models' performance after 90% of the training set is used, Bi-LSTM outperforms SVM. SVM only is shown to outperform Bi-LSTM in Zhang et al. work [13] when no more than 25% of the dataset is used in training. Bi-LSTM with attention has a technical advantage for the problem at hand by being able to identify small regions of text within the Quora question associated with toxic language or hate speech and implicit toxic language as well [4]. This can explain it's out-performance of SVM.

The BERT models outperform Bi-LSTM and SVM in terms of F1 score except for Bi-LSTM outperforming DistilBERT Cased. Bi-LSTM outperforms DistilBERT Cased for validation and testing. In comparision of all of the types of BERT model configurations when running in validation and testing, the uncased models always outperform the cased models in terms of F1 score. This probably due to the type of problem not being a Named Entity Recognition or Part-of-Speech tagging problem where case is important. The addition of some words being cased could just be adding noise to the data. When comparing uncased BERT models, DistilBert outperforms BERT in validation and testing. This could be due to the 4 additional epochs that DistilBERT is trained for.

Even though SVM is utilizing a representation of bag-of-centroids and these centroids are of semantically related clusters, it is still ignoring grammar and word order resulting in an abstraction only based on the number of times tokens in a Quora question are in semantically related clusters. BERT's outperformance of Bi-LSTM in most situations can be attributed to Bi-LSTM being a context-free model only generating one word embedding for every word that is in the vocabulary [5]. While BERT is a deeply bi-directional incorporating both left and right contexts of every sentence in its word representations [5]. This more advanced representation can have major repercussions on performance when moving on to classification. Although Bi-LSTM with attention had the ability to identify small regions of toxic content within questions as well as implicit toxic content, its weakness is still that it depends upon sequential information flow [11]. BERT is still a more advanced model due to its masked language model with a representation resulting from a combined left and right context allowing it to outperform Bi-LSTM with attention [2].

7 CONCLUSION

Possible shortcomings of our work could be that BERT was not trained for as many epochs as necessary. We saw that DistilBERT outperformed BERT often which could be very much attributed to the 4 additional epochs that DistilBERT is trained for. A more advanced analysis of learning with longer training time could be performed to see at what number of epochs does the model performance plateau. Another possible limitation of our work was that the BERT-Base Uncased model may be too small of a model compared to the larger BERT-Large model [2]. Another limitation is that our current dataset has an extreme class imbalance: 1,225,312 Quora questions with a label of 0 (not insincere) and 80,810 Quora questions with a label of 1 (insincere) not allowing BERT to be fine-tuned on enough insincere questions.

BERT-large outperformed BERT-Base in terms of accuracy for all four datasets used [2]. If the BERT-Large Uncased model was used instead, the 24-layer, 1024-hidden, 16-heads, 340M parameters configuration could be used to better model the complexity of our problem allowing for a possible increase in performance [5]. To try to alleviate the effects of the class imbalance during training we could try to obtain more toxic content data from data sources other than Quora. This could provide BERT with a better representation of questions that contain insincere or toxic content during training. Recent research by Google has resulted in a optimized version of BERT called ALBERT. By factorization of the embedding parametrization and parameter sharing across layers, ALBERT achieved a 89% parameter reduction [7]. This allows for the ALBERT model to scale up again. AlBERT-xxlarge had 30% less parameters than BERT-large but had a 4.2 increase in performance on the SQuAD2.0 test and 8.5 increase on the RACE test [7].

8 CONTRIBUTION

All team members have contributed a similar amount of effort.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473 (2014).

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[3] Hugging Face. 2020. Transformers Library. https://huggingface.co/transformers/model_doc/bert.html#bertforsequenceclassification

[4] Lei Gao and Ruihong Huang. 2017. Detecting Online Hate Speech Using Context Aware Models. In *RANLP*.

[5] Google. 2020. BERT Google Research Github Repository. https://github.com/google-research/bert

[6] Jinshan Wang Hengliang Luo Jiahuan Lei, Qing Zhang. 2019. BERT Based Hierarchical Sequence Classification for Context-Aware Microblog Sentiment Analysis. In *Neural Information Processing*.

[7] Google Research. 2019. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. https://ai.googleblog.com/2019/12/albert-lite-bert-for-self-supervised.html

[8] Julien Chaumond Thomas Wolf Victor Sanh, Lysandre Debut. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

[9] Julien Chaumond Thomas Wolf Victor Sanh, Lysandre Debut. 2020. DistilBERT Github Repository. https://github.com/huggingface/transformers/tree/master/examples/distillation

[10] Xianglu Yao. 2018. Attention-based BiLSTM Neural Networks for Sentiment Classification of Short Texts. In *CloudCom 2018*.

[11] Linfeng Song Yue Zhang, Qi Liu. 2018. Sentence-State LSTM for Text Representation.

[12] N. Zainuddin and A. Selamat. 2014. Sentiment analysis using Support Vector Machine. In *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*. 333–337. https://doi.org/10.1109/I4CT.2014.6914200

[13] Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner. 2019. *Detecting Toxic Content Online and the Effect of Training Data on Classification Performance.* Technical Report. EasyChair.