Toxic Comment Classification using Transformers

Akash G, Himanshu Kumar and Bharathi D
Department of Computer Science
Amrita School of Engineering
Coimbatore, India
akashsuper2000@gmail.com, himanshu6k@gmail.com, d_bharathi@cb.amrita.edu

Abstract

This paper focuses on a comparison study between the traditional deep learning techniques like LSTMs and GRUs with the latest state-of-the-art transformers for the task of classifying a comment based on its toxicity. Specifically, four different deep learning models are built, implemented and trained on a standard dataset for comparing the performance of the models. The models that are explored and compared are Bi-directional LSTMs, GRUs, Bi-directional LSTMs with CNNs and Transformers (with pre-trained RoBERTa weights).

Keywords
Transformers, GRU, LSTM, CNN, RoBERTa

1. Introduction

With the growth of Internet usage, communication has been revolutionized in a way no one had ever imagined. The people suddenly have access to real-time discussion forums where a huge number of users could actively engage simultaneously and have meaningful discussions and debates.

While this type of engagement contributes significantly to the quality of discussions and idea sharing, it also puts the netizens at a risk of being harassed by personal attacks and cultural assaults. This has triggered the industrial and research community in the last few years to arrive at a solution to detect and remove toxic comments from these forums. However, the solutions developed for this problem are still in their infancy and are highly unreliable. Thankfully, advances in hardware, cloud computing and natural language processing allow the development of NLP-based Deep Learning approaches.

It has been a while since the traditional text classification models like dictionary-based models and TF-IDF-based machine learning models have been replaced with complex deep learning models. Bi-LSTM models and GRU models have been used in the paper to help the model in identifying the dependencies between the words in a paragraph. A hybrid Bi-LSTM + CNN model has also been used to extract higher-level features with CNN layers that use convolutional layers and max-pooling for extracting local stationarity features, while the LSTM layers capture the long-term dependencies between the word sequences. For text classification, in particular, the use of pre-trained transformer models has been proposed to effectively understand the context and structure of the text and classify it based on its toxicity. The latest findings have shown how pre-trained embeddings can greatly help the model in understanding the word-word co-occurrence relations. In this paper, the 300-dimensional GloVe embeddings trained from 2.2M word corpus is used that offers a word-word co-occurrence matrix that tabulates how frequently words co-occur with one another in the given corpus.

1.1 Objectives

The objective is to build the aforementioned deep learning models and train them on a common dataset with similar hyperparameters to compare their performance on standard and reliable metrics.

2. Literature Review

Related work has been mainly into methods to detect a toxic comment, challenges faced in the process and proposals for the inclusion of new and innovative architectures for detection. One such proposed architecture for solving NLP tasks, in general, is BERT proposed by Devlin et al. (2019) in their paper. The paper primarily focused on all layers of the BERT architecture, pre-trained procedures, fine-tuning of the model and analyzing the performance of the model based on standard parameters and benchmark scores including GLUE score, MultiNLI accuracy and F1 score. The paper explains how an attention layer can greatly help in solving various challenges in the field of NLP. Yang et al. (2020) in their paper proposes an incremental iteration for the BERT model that is a generalized autoregressive pre-training method that tries to enable learning bidirectional contexts by maximizing the expected likelihood over all possible permutations of the factorization order. It also succeeds in overcoming the limitations of BERT with its proposed autoregressive formulation. The paper also claims better scores on 20 diverse tasks over the traditional pre-trained transformer architecture. The paper proposed by Joshi et al. (2019) proposes a pre-training approach called RoBERTa that was specifically developed to overcome the shortcomings of BERT. The primary differences in the pre-training method were that the model was trained over more data, for more epochs and with a bigger batch size, and the next sentence prediction objective was removed from the process.

Non-transformer architecture approaches include the capsule layer proposed by Srivastava et al. (2018) that is employed to capture the instantiated parameters of the inputs like the order of words and semantic representation thus, allowing the model to learn the contexts better. Another interesting approach discussed in the paper by Georgakopoulos et al. (2018) discusses the uses of CNN to exploit local stationarity of data. The intuition is that the structure and morphology of the words appearing in a sentence are directly dependent on its neighboring words in the same sentence. The paper also compared this approach with other similar NLP techniques.

Machine learning methods proposed for general NLP problems include the use of SVM and Decision Tree models in the paper by Gann et al. (2019) where the authors have discussed these methods in detail for a specific use case. Furthermore, the paper also elucidates benchmarks like Daily Sentiment Index (DSI) Sentiment Key Performance Index (SKPI), polarized sentiment and mood analysis. Elbagir et al. (2019) in their paper has aimed at analyzing the sentiments of tweets based on ordinal regression with machine learning algorithms. Park et al. (2019) proposed the usage of Word Sense Disambiguation techniques with a new unsupervised fuzzy rule-based model for classifying the comments into negative, neutral and positive sentiment classes.

Transfer learning methods and performance has been verbosely discussed in the paper by Raffel et al. (2020) that explains the transfer learning techniques for NLP by introducing a new unified framework and using the 'Colossal Clean Crawled Corpus'. They have achieved outstanding results for various NLP tasks like question answering, text summarization and text classification.

Challenges faced in the process of classifying a comment based on toxicity have been discussed in several papers including the one by Aken et al. (2018) that focuses primarily on the types of challenges faced when approaching the task of toxic comment classification and also proposes a few methods of circumventing it. Mohammad et al. (2018) in their paper questions whether proceeding to build state-of-art models is worth the effort considering the numerous difficulties in the way.

3. Methods

Figure 1 presents a comprehensive overview of the system designed to carry out the study. There are five components to the system; data collection (collecting data from different sources), preprocessing (preparing the data for training), word embedding (using pre-stored word-word embedding matrix), model training (training multiple models on the data) and performance analysis.

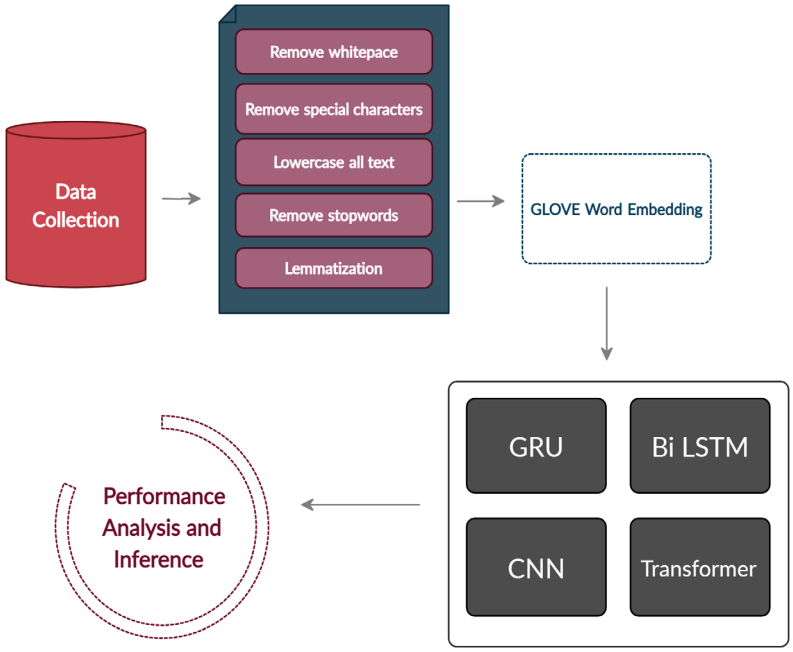

Figure 1: System Overview

3.1 Bi-directional LSTM

Bi-directional or BiLSTM came into existence after working out on different problems of existing models. It all started with the Recurrent Neural Network. RNN was developed to deal with the datasets with sequential inputs; real-valued vectors are passed at the input nodes, each vector in turn. At some random time step, each non-input unit registers its present initiation (result) as a nonlinear function of the weighted amount of the enactments of all units that associate with it. The recurrent hidden states in RNN gives it the ability to predict the next input sequence. The Limitation of this model is that the RNNs memory is limited to only a few words. It cannot handle long-term dependencies.

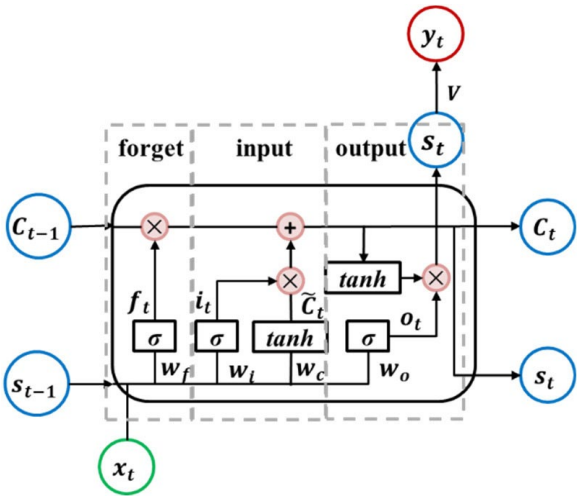

Figure 2: Structure of an LSTM cell

LSTM was specially modeled to deal with the issue of long-term dependencies. The LSTM memory is also known as a gated cell since it uses multiple gates to identify what memory to retain and what to discard. As illustrated in figure 2, the LSTM does so via input, output, and forget gates; the input gate identifies what part of the new cell state to hold

on to, the forget gate decides what portion of the existing memory to forget, and the output gate determines how much of the cell state should be passed on to the next layers.

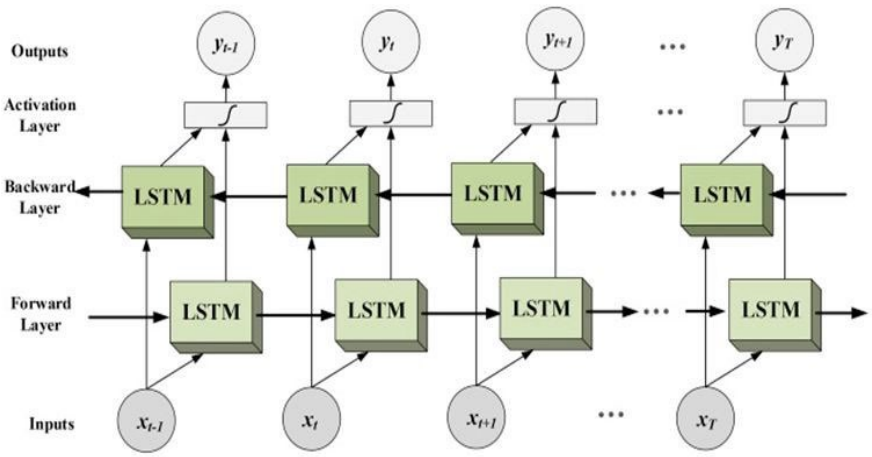

Figure 3: BiLSTM model

As we have seen LSTM computations of weight are done based on the data at the previous timestamp, which makes LSTMs a very productive model for sequential modeling. BiLSTM is one of the derivatives of LSTM that aims to solve the uni-directional dependency. As shown in Figure 3, BiLSTM uses 2 LSTMs simultaneously, first, to train the sequences in the forward direction and the second one to train the same sequences in the backward direction. This helps the model understand the context of the sentence much better.

3.2 GRU

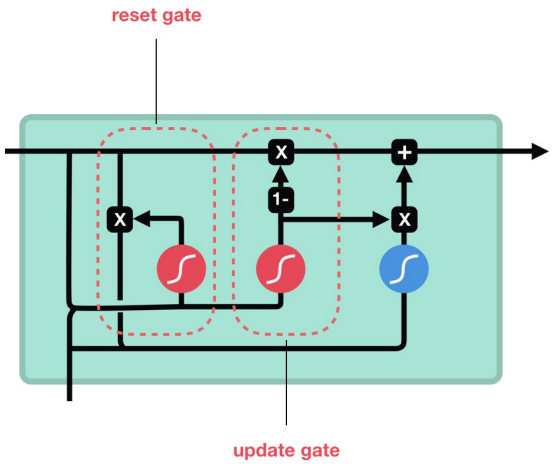

Figure 4: Structure of a GRU cell

GRU is an improvement over the traditional RNN. When the sequence of vectors that carries the information about the input passes through a series of layers, the information vanishes in the process and is lost by the time when it reaches the end or beginning layer, this problem is called vanishing gradients. This makes it challenging for the model to capture long-term dependencies. Similar to LSTMs it too helps in solving the vanishing gradient problem. It does so utilizing a reset and an update gate. This is illustrated in Figure 4 that the reset gate lies between the past and the next applicant activation to fail to remember the past state, and the update gate decides the amount of the candidate

initiation to utilize while refreshing the cell state. GRUs have fewer tensor operations; therefore, they are much faster to train than LSTM.

3.3 BiLSTMs with CNNs

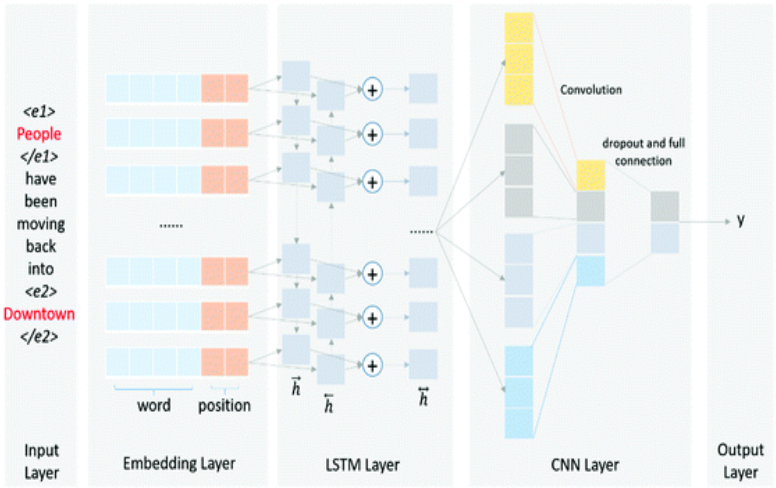

Figure 5: BiLSTMs with CNNs model

Convolutional Neural Network is a deep learning network with several layers of convolutions with nonlinear activation functions like ReLU or TanH. It is an improvement over the error backpropagation network CNN was mainly used for solving different computer vision tasks. Due to the fastness of building and training models and efficiency in terms of representation. As we have seen already BiLSTM models are very helpful when it comes to dealing with the application related to sequential text input. Figure 5 facilitates the understanding of how we used two different types of neural networks to create a better model. The objective of using the hybrid model is so that BiLSTM can help us in extracting the global features related to context and CNN in extracting the local features.

3.4 Transformers

Transformers is a multi-head attention mechanism that learns contextual relations between the words in the given text. Generally, a transformer consists of two separate mechanisms - an encoder that accepts the text input and an optional decoder or a sigmoid/softmax layer that produces a prediction for the task. BERT is a pre-training approach that uses this architecture for modeling. There are still a lot of areas of improvement possible in the original BERT model. After identifying this, FAIR (Facebook AI Research) introduced a robust and optimized version of BERT which is named RoBERTa (Robustly Optimized BERT Pre-training Approach). Figure 6 presents a broad overview of the transformer architecture.

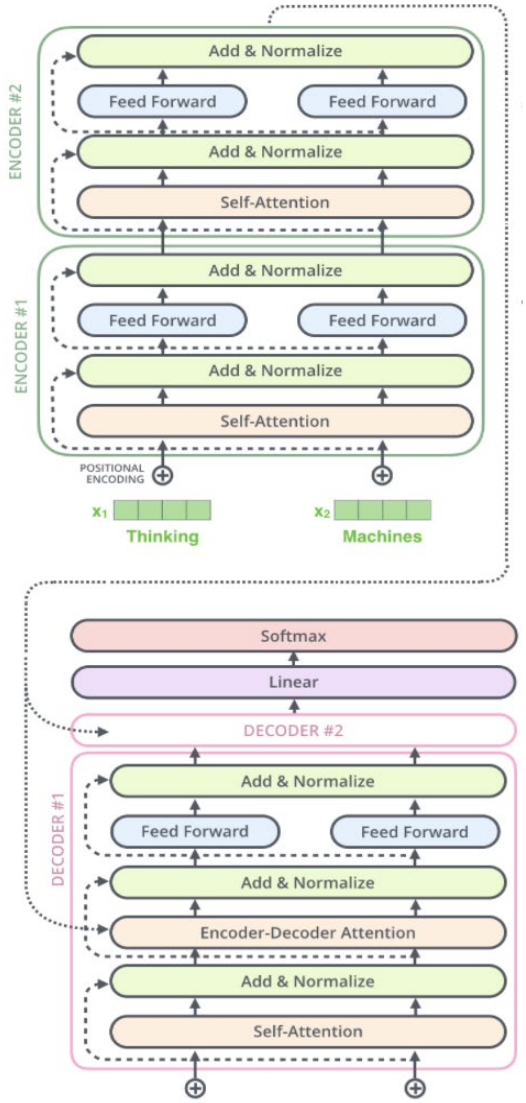

Figure 6: Transformers architecture

RoBERTa builds upon BERT's language masking strategy, where the system learns to predict hidden sections of text within otherwise unannotated language examples. The model uses a byte-level Byte-Pair Encoding (BPE) encoding scheme with a vocabulary containing 50K subwords. RoBERTa, which was originally implemented in PyTorch, modifies key hyperparameters in BERT, including the removal of BERT's Next Sentence Prediction (NSP) objective, and training it with much larger mini-batches and learning rates. This enables the model to improve on the masked language modeling objective compared to BERT and leads to better performance.

4. Data Collection

4.1 Dataset Description

The dataset used to train the models is derived from Kaggle. The dataset is a compilation of comments from various discussion forums including Civil Comments and Wikipedia Talk Pages. The dataset is under CC0, with the underlying comment text being governed by Wikipedia's CC-SA-3.0. The Kaggle competition data consists of data points that are categorized into 6 labels. The objective of this paper is to build a model for binary classification of text to identify whether it is toxic or not.

Table 1: A random sample from the dataset

Comment	Toxic
Stupid law. Why does Quebec always have to cause a fuss over everything? Just leave people alone, for heaven's sake.	1
sorry people, i'm just bored!}}	0
fuck you white trash!!!	1
White racism leaks out through the seams of an establishment race bigot.	1
Who are you to determine fact from fiction? The general public considers the NAACP and ADL to be civil rights organizations. David Duke is regarded as a racist by most and even the US Government has investigated his actions as a domestic terrorist along with federal tax evasion.	0

The six labels that were present in the original dataset were to identify the type of toxicity. To fulfill the objective, the dataset has been converted to contain a single label that classifies the given comment based on its toxicity i.e. label 0 for non-toxic comments and label 1 for toxic comments. A sample for the dataset is presented in Table 1. Initially, the original dataset was skewed towards the non-toxic side. To better generalize the model and to avoid bias towards the non-toxic label, the dataset has been transformed into a balanced one by randomly sampling non-toxic labels. By doing this, the model has been able to generalize and perform with improved accuracy.

There are over 200000 data points available for use from which a subsample of 160000 data points is extracted for use. The dataset is evenly distributed among the labels namely, toxic and non-toxic.

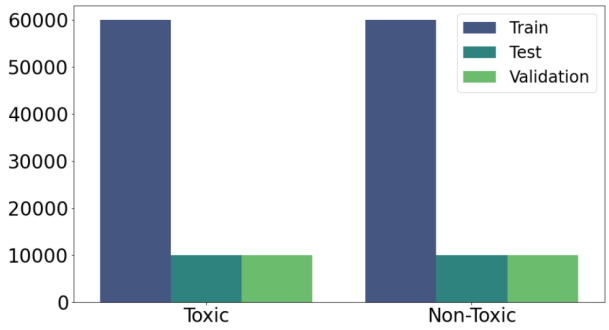

Figure 7: Data distribution

As illustrated in Figure 7, the dataset is divided into three parts with 75% for training and 12.5% each for validation and test sets. Training, validation and test sets downsampled to ensure that they are balanced.

4.2 Data Preprocessing

The main part of preprocessing is to remove redundant white spaces, numbers and special characters, convert all the non-ASCII characters to ASCII characters and expand abbreviations. Figure 8 helps us determine the maximum length of the sequence, resulting in the layer being padded or truncated with the exact length values of the sequence.

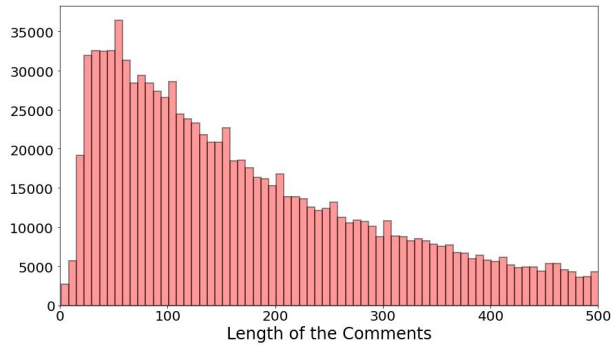

Figure 8: Length vs Number of comments

Further, all the characters are converted to lowercase and lemmatized (replacing the original word with its root word). The result is then passed through GloVe embedding for vectorizing the sentences which are used as the inputs for the models.

5. Results and Discussion

5.1 Evaluation Metrics

In this paper, we have used the ROC-AUC score, F1 Score and Accuracy score to compare the performances of the models. To calculate the ROC-AUC score we need to follow the following steps:

1. Calculator the True Positive Rate (TPR) and False Positive Rate (FPR) for each threshold with the below formula:

   TPR= TP / (TP + FN)

   FPR= 1 - (FP / (TN + FP))

   where TP (True Positive) is an outcome where the model's prediction of the positive class is correct and TN is a similar outcome for a negative class. Similarly, FP (False Positive) is when the model incorrectly predicts the positive class and FN is when the model prediction of the negative class is wrong.

2. Plot a curve between TPR and FPR.

3. Calculate the area under the curve in the resultant graph.

Higher the AUC score, the better the model's performance.

5.2 BiLSTM

The model was trained for 20 epochs monitoring the validation AUC score. The model uses Adam as the optimizer and binary cross-entropy as the loss function for the classification. Due to this callback, the model faces early stopping which ensures that the model has retained the best weights. Since the test data is from the same source as the training and validation data, the performance is significantly higher than usual. The results are presented in Table 2.

Table 2: BiLSTM Results

Model	AUC score	Accuracy
BiLSTM	0.9871	0.9439

5.3 GRU

GRUs are also trained in a similar approach used for training BiLSTMs. For training the model, an embedding layer was used that produces the embedded matrix weights after obtaining vector representations for words using GloVe word embedding. It is then passed through a dense layer with sigmoid as the activation function. After experimenting, it is very clear that there is no clear winner between these two RNN models i.e GRU and LSTM. GRU gives a slightly better result when compared to BiLSTM but the difference is not significant. The results are presented in Table 3.

Table 3: GRU Results

Model	AUC score	Accuracy
GRU	0.9867	0.9426

5.4 BiLSTM with CNN

The hybrid model consists of an initial BiLSTM network layer that receives the word embeddings for each token in the sentence given as input. The idea behind doing this is to not only capture the initial token but also to retain the previous tokens. The output of this layer is then passed on to the convolution layer which extracts the features associated with the spatial locality of data. Finally, the convolution layer's output is passed through a sigmoid layer that predicts the probability for the comment's toxicity. The model is trained using Adam optimizer with the initial learning rate 1e-5. The results are presented in Table 4.

Table 4: BiLSTM with CNN Results

Model	AUC score	Accuracy
BiLSTM with CNN	0.9873	0.9441

5.5 Transformers (with RoBERTa weights)

For training the given dataset using a transformer, pre-trained RoBERTa weights are used. The activation function used is sigmoid. The optimizer used is Adam with an initial learning rate at 1e-5. It is evident that the Transformer model outperforms all the other models. The results are presented in Table 5.

Table 5: Transformer Results

Model	AUC score	Accuracy
RoBERTa	0.9918	0.9515

6. Conclusion

To summarize the work, four deep learning models with different architectures were implemented on the same dataset and results were compiled. The state-of-the-art transformer model out-performed all the other models by a discernible margin. BiLSTMs with CNNs was the second best and closest to the transformer model.

For future work, newer and more innovative architectures can be implemented on an even bigger dataset that contains comments from a diverse range of online forums.

Further, multilingual models can also be implemented for classifying toxic comments in various languages.

Acknowledgements

We would like to acknowledge the Jigsaw and Kaggle for providing us with datasets and HuggingFace for the tokenizers and weights:

- https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification/data
- http://nlp.stanford.edu/data/glove.840B.300d.zip
- https://huggingface.co/jplu/tf-xlm-roberta-large

References

Devlin, J., Chang, M.W., Lee, K., Toutanova K., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv: 1810.04805v2 [cs.CL]* 2019

Zhilin, Y., Dai, Z., Yiming, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V., XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv: 1906.08237v2 [cs.CL]* 2020

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Wei, L., Liu, P.J., Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv: 1910.10683v3 [cs.LG]* 2020

Shihab, E., Jing, Y., Twitter Sentiment Analysis Based on Ordinal Regression. *In: IEEE Access, voL. 7, pp. 163677 - 163685 DOI: 10.1109/ACCESS.2019.2952127* 2019

Gann, W.J.K., Day, J., Zhou, S., Twitter Analytics for Insider Trading Fraud Detection. *Available: https://docplayer.net/14928867-Twitter-analytics-for-insider-trading-fraud-detection.html* 2014

Park, S., Lee, J., Kim K., Semi-supervised distributed representations of documents for sentiment analysis. *ScienceDirect: S0893608019302187* 2019

Georgakopoulos, S.V., Tasoulis, S.K., Vrahatis, A.G., Plagianakos, V.P., Convolutional Neural Networks for Toxic Comment Classification. *arXiv: 1802.09957v1 [cs.CL]* 2018

Aken, B.V., Risch, J., Krestel, R., Loser, A., Challenges for Toxic Comment Classification: An In-Depth Error Analysis. *arXiv: 1809.07572v1 [cs.CL]* 2018

Joshi, M., Liu, Y., Ott, M., Goyal, N., Du, J., Chen, D., Levy, O., Lewis, M., RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv: 1907.11692v1* 2019

Srivastava, S., Khuran, P., Tewari, V., Identifying Aggression and Toxicity in Comments using Capsule Network. *In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, pp. 98–105 Santa Fe, USA*. 2018

Schmidt, A., Wiegand, M., A Survey on Hate Speech Detection using Natural Language Processing. *In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pp. 1–10, Valencia, Spain.* 2017

F. Mohammad, Is preprocessing of text really worth your time for online comment classification? *arXiv:1806.02908* 2018

Biographies

Akash G is an undergraduate student pursuing his Bachelor of Technology in Computer Science at Amrita School of Engineering, Coimbatore, India and will be graduating in the year 2021. He has worked on a variety of research-oriented projects ranging from image recognition to time series predictions to sentiment analysis in the field of artificial intelligence and has co-developed numerous innovative solutions for various organizations. Akash is an accomplished competitive programmer and has achieved top ranks in various international programming competitions and CTF contests. He is also one of the top-ranked Kagglers with several competition medals to his name. The goal

of his research is to help build better and more sophisticated systems to address the challenges faced in the field of computer science.

Himanshu Kumar is a student pursuing Bachelor of Technology in Computer Science at the Amrita School of Engineering, Coimbatore, India and will be graduating in 2021. He has secured the second rank in AI Adept (AI Problem solving competition) at the National level Techfest organized by Amrita School of Engineering. Mr. Kumar has worked on various minor projects focusing on machine learning and deep learning algorithms which includes Detecting the text from image and translating into other languages, Image classification tasks, etc. His current work is mostly focused on various NLP techniques and tasks such as sequence to sequence modeling (language translation) and sequence to vector modeling (text classification).

Bharathi D currently serves as an Assistant Professor at the Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, India. Her areas of research include Machine Learning, Image Processing, and Remote Sensing. She is currently pursuing her Ph.D.