Toxicity Detection Using State of the Art Natural Language Methodologies

Enes Faruk Keskin
Turkish Aeronautical Association University
Department of Computer Engineering
Ankara, Turkey
enesfarukkeskin@gmail.com

Erkut Açıkgöz
Atılım University
Department of Industrial Engineering
Ankara, Turkey
acikgoz.erkut@student.atilim.edu.tr

Gulustan Dogan
University of North Carolina Wilmington
Department of Computer Science
Wilmington, USA
dogang@uncw.edu

2023 8th International Conference on Control and Robotics Engineering (ICCRE) | 979-8-3503-4565-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCRE57112.2023.10155587

***Abstract*—In this paper, the studies carried out to detect objectionable expressions in any text will be explained. Experiments were performed with Sentence transformers, supervised machine learning algorithms, and Bert transformer architecture trained in English, and the results were observed. To prepare the dataset used in the experiments, the natural language processing and machine learning methodologies of the toxic and non-toxic contents in the labeled text data obtained from the Kaggle platform are explained, and then the methods and performances of the models trained using this dataset are summarized in this paper.**

***Index Terms*—language models, bert, transformers, natural language processing, supervised machine learning algorithms, deep learning**

I. Introduction

On many online platforms, users can publicly share objectionable (toxic) text message content without supervision. In addition, any English text may contain toxicity. The study carried out to determine the toxicity level of such messages and texts with machine learning-based approaches will be summarized in this paper.

Toxic texts contain insults, violence, or obscene expressions. The presence of such texts as comments, messages, or any text on websites reduces the site's quality. And this annoys the readers. For this reason, the need to measure the toxicity of any sentence before it is publicly published has arisen.

Natural language processing methods are frequently used in text classification studies. Text classification applications are used for problems such as eliminating spam in the e-mail box or categorizing books in the library. Sentiment analysis is one of the most well-known applications of natural language processing methodologies. This application is aimed to determine with machines whether a given sentence contains a positive or negative emotion. To perform the emotion detection, the model should be trained with a training dataset, and then the model should be evaluated with the test dataset. For this purpose, Bag-of-words, Recurrent Neural Networks, and, as of 2017, transformer-based approaches have been developed. The way these approaches deal with problems can be briefly summarized as follows:

- **Bag-of-words:** It is a method that does not consider the order of the words in the sentence. [1].
- **Recurrent Neural Networks:** With an optional attention mechanism that takes into account the order of the words in the sentence but does not pay attention to the difference in the meaning of the words in different contexts, they can pay attention to the specific parts of the sentence in predictions.
- **Transformer:** Pays attention to the order of words; unlike RNNs, the attention mechanism is always included in the system in a versatile way, learns the relationship between both input words and input-output words, consists of an attention mechanism, and is aware of the meaning differences of words depending on the context, 8- It has a methodology using a deep architecture with 12-24 layers.

In this article, the process of measuring the toxicity level in the text will be summarized by explaining the Bert language models based on transformer architectures, which is the most popular approach in natural language processing. Afterward, the techniques and methodologies used will be explained. Obtained results will be shared and interpreted. The article will end with a discussion section.

II. Methodology

This section gives detailed information about transformers architectures, Bert, and supervised machine learning algorithms used.

A. Transformer Architectures and Bert

Transformer architectures, Transformer architectures took their place in the literature in 2017. This approach uses encoder and decoder architectures for input and output. Both the encoder and decoder architectures are designed as deep architectures, including attention mechanisms (attention). The encoders of the words given as input are given as input to the next layer. In contrast, the interaction of these words with other words is evaluated within the attention mechanisms. In the last layer, the output of the encoder is given as input to the decoder. In the decoder, there are both attention mechanisms with the cells in the input (Figure 1), separate attention mechanisms in the input encoder, and separate attention mechanisms between the encoder and the output encoder and within the output encoder. With this approach, each word is represented by a 768-dimensional vector, as seen in (Figure 2).

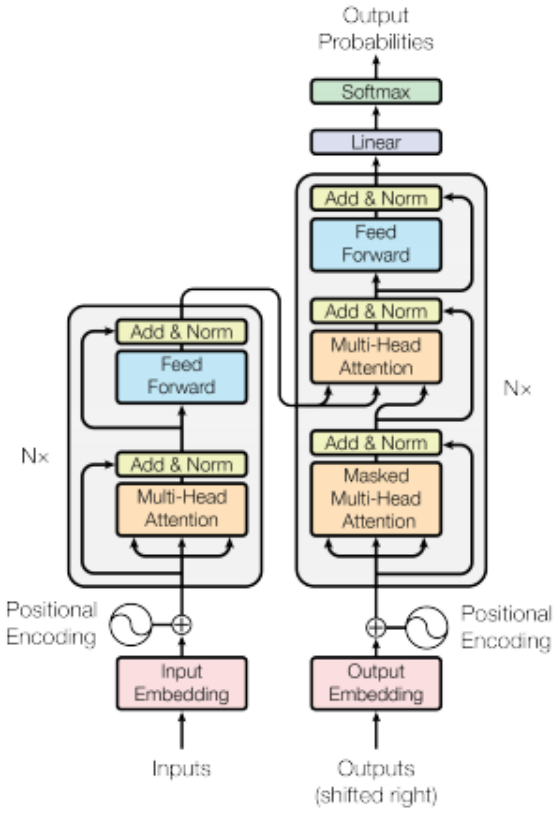

Fig. 1. Transformer model architecture [1]

Bert, It is a method that Google introduced in 2017 [1]. This approach involves training a language model that gives importance to different parts of the sentence. This language model is known as the masked language model. The training process takes place in an unsupervised manner. A random word is masked from the sentences in the training set, and the language model is trained to predict the word correctly. In addition, consecutive sentence pairs and non-consecutive sentence pairs are also given to Bert as input, and Bert can distinguish consecutive word pairs from non-consecutive ones. Classification by transfer learning is possible in a subtask, such as classifying a sentence later.[2].

Huggingface is a New York-based company implementing this type of transformer architecture. Experiments were carried out by applying transfer learning (fine tuning) to this model within the scope of the study [3].

Bert's encoder architecture is available. The words are separated to their roots at the input stage in the encoder architecture and then pass through a deep architecture network. Unlike the word2vec approach, the words are represented depending on the context in which they are located.

Bert - Transfer Learning process, there is no change in the coding part of the Bert architecture. In the sequence classification process, the output of the text encoded with PyTorch is given as an input to a Feedforward Neural Network (FFNN), the weights of this network are trained with labeled data, so it is not possible to train a language model from scratch.

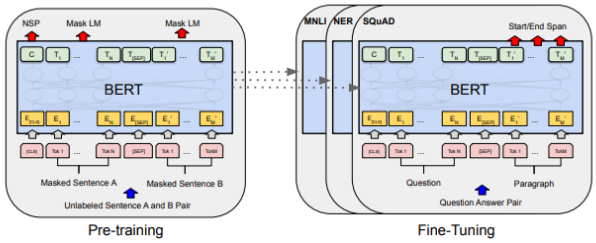

Fig. 2. Bert transfer learning architecture example [2]

B. Supervised Machine Learning Algorithms

This section gives brief information about the supervised machine learning algorithms used in the study.

Logistic Regression, It is an algorithm that shows high performance in binary classification problems when the dependent variable takes two different values. [4]. It is widely used in linear classification problems. This dependent variable; determines the probability of realization of the values that the advertisement's content can take as appropriate or inappropriate. The mathematical representation of the algorithm is shown in Formula 1, p represents the probability of success.

$$logit(y) = \log(\frac{p}{1-p}) \tag{1}$$

Random Forest, It is a supervised learning algorithm that can be used for classification and regression problems, combining these decision trees to make an accurate prediction by creating more than one decision tree. [4].

Xgboost Classifier, It is a hybrid algorithm that uses a gradient boosting framework based on a decision tree. [5]. The development of Xgboost started from decision trees and continued as bagging, random forest, augmentation, and gradient augmentation and took its final form.

K-Nearest Neighbors (KNN), It is a classification algorithm that is based on two values, the distance and the number of neighborhoods. It produces predictions according to the class of the neighboring vectors that are closest to the vector to be estimated. [6]. The given optimum k parameter calculates over the k nearest neighbor vectors. Over-learning can occur when the given k value is too small and too large and produces very general predictions. Ideal estimation results are reached by choosing the optimum number of K neighborhoods.

III. Results

This section details the experiments performed with different methodologies, the results, and the data preprocessing operations performed. In addition, the dataset used in the study and the methods of obtaining vector representations of the dataset is explained.

The experiments were performed on a macOS operating system computer using the M1 processor with 16GB of RAM.

A. Dataset

In this study, different datasets on the Kaggle platform were used. [1],[2].

The dataset used in the study was taken from the Jigsaw Rate Severity of Toxic Comments competition on the Kaggle platform [1]. 30108 more toxic and 30108 less toxic consists of 60216 lines of text data in total. In addition, the dataset for this competition contains text that may be considered profane, vulgar, or offensive. In this study, this data set is used as a test and validation set.

For the training of Bert and Supervised ML algorithms, a different data set from the kaggle platform was used [2]. The dataset used contains less toxic, more toxic, and normal comments. In addition, it has been observed that there are rows with different languages in the data set. By adding another column to this data set, the language of the sentences was recorded in this column.

The most common words in the dataset texts were determined in the exploratory data analysis studies. Graph of the 30 most frequent words Figure 3.

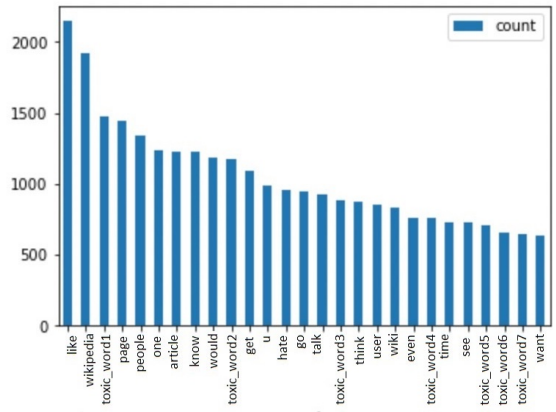

Fig. 3. 30 most frequently used words in sentences

The sentences in the dataset were vectorized with the CountVectorizer, matrices were created with the most used word groups, and the results were observed with different word counts in separate trials. The most toxic words have been identified.

In order to adapt the dataset used in the experiments to the machine learning methodology, the data were divided into different sections. The entire data was divided into three parts: the training set, the validation set, and the test set. The ratios used in this study are:

- 60% training set,
- 20% test set,
- 10% validation set.

Data Preprocessing Steps: Data cleaning in machine learning studies differs from problem to problem. The preprocessing steps performed on the data in the study are given below in order:

- Words are all represented in lowercase letters,
- The language of the content of the text was determined by the library named textpipe,
- Non-English content is not included in the training.,
- Beautifulsoup library was used during text cleaning,
- Eliminated html tags with the Beautifulsoup library.

B. Results

Python programming language was used in the experiments. Scikit-learn library was preferred in supervised machine learning studies, and PyTorch-Transformers library was preferred in Bert studies.

Experiments carried out in machine learning studies should be evaluated with success measurement metrics following the study methodology. Multiple metrics were used to monitor the results in this study. These metrics are accuracy, sensitivity, precision, and F-score.

- **Accuracy:** It represents the ratio of correct answers in the predictions made to all answers. It is the most widely used evaluation method. The calculation of the accuracy is shown in Formula 2.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2}$$

- **Recall:** It is the ratio of correctly detected positive classes in the estimates to all positives. The calculation of the recall is shown in Formula 3.

$$Recall = \frac{TP}{FN + TP} \tag{3}$$

- **Precision:** It represents how many of the positively predicted values are actually positive. The calculation of the precision is shown in Formula 4.

$$Precision = \frac{TP}{FP + TP} \tag{4}$$

- **F-Score:** It is the harmonic mean of the Precision and Sensitivity values. The calculation of the F-Score is shown in Formula 5.

$$F - Score = 2 \times (\frac{Precision \times Recall}{Precision + Recall}) \tag{5}$$

*1) **Supervised Machine Learning Algorithms Results**:* Experiments were conducted using supervised machine learning algorithms using the dataset taken from the Kaggle website. XGboost, random forest classifier, KNN, and Logistic Regression algorithms were used in the experiments. The results of the models on the test data are given in Table 1.

TABLE I
SUPERVISED MACHINE LEARNING ALGORITHMS RESULTS

Algorithms	Accuracy	F-Score	Recall	Precision
Logistic Regression	62%	62%	64%	61%
XGboost	60%	63%	67%	59%
Random Forest Classifier	61%	63%	68%	59%
KNN (k=8)	51%	67%	69%	50%

In the first stage of the study, the texts in the dataset were represented by vectors and given as input to the classification algorithms for training and testing processes. Different

methodologies have been used to vectorize texts, with the most successful result being TF-IDF. TF-IDF is one of the most preferred algorithms, enabling texts to be represented by numbers and obtaining vector representations.

Tf-Idf, represents each different word in the text with a number. Each sentence is a vector of the weights of these numbers. TF is calculated by dividing the number of times the term appears in a document by the total number of words. The IDF is calculated by taking the logarithm of the number of documents in which the term appears divided by the total number of documents. TF-IDF weighting is calculated by multiplying the TF and IDF values. [1].Its mathematical representation is in Formula 6.

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i}) \tag{6}$$

*2) **Bert Results**:* In the study, Bert's "bert-base-english-uncased" model and tokenizer found in huggingface were used [1]. As a result of the exploratory data analysis of the data we have, it has been determined that the number of words in the texts does not exceed 250. Therefore, the input length of the model was chosen as 250. "Adam Optimizer" was selected as the optimizer, and the batch size value was set to 32.

The Bert model trained for 5 epoch was used in the study. Table II contains the total error values (loss) and metric values of the model trained for 5 epoch.

TABLE II
RESULTS OF ERROR VALUES AND MATRIC

Devir	Training Loss	Validation Loss	Training Epoch Took	F-Score	Precision	Recall	Accuracy
1	0.321968	0.201502	0:52:21	0.932962	0.944188	0.922	0.922642
2	0.191665	0.179053	0:52:16	0.939381	0.950520	0.9285	0.930037
3	0.170393	0.174333	0:53:06	0.939918	0.957468	0.923	0.931108
4	0.161951	0.169272	0:53:47	0.941276	0.958852	0.924333	0.9326651
5	0.157089	0.168838	0:53:49	0.942074	0.959074	0.925666	0.9335409

Visualized results of error values and metrics in Table II are shown in Figure 4 and Figure 5, respectively.

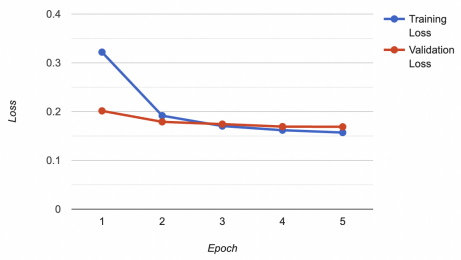

Fig. 4. Training and Validation Error Values

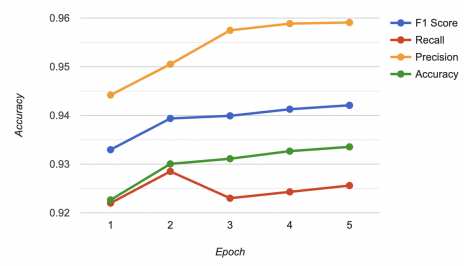

Fig. 5. Metric Values Observed During Training

IV. DISCUSSIONS AND CONCLUSION

This study aims to provide control with deep learning-based methodology instead of manually determining the toxicity level in any text by the controllers. It can be used to shorten the duration of text sharing with inappropriate content by measuring the toxicity level of text shares made by people on any internet platform.

While Tf-Idf is concerned with the frequency of words in sentences, transformer language models also give importance to the context of the word. It takes the context of the word into account to distinguish between vectors of words that have more than one meaning. For this reason, Bert preferred at the stage of vectorizing the sentences in the study. Sentences vectorized with Bert are trained with the Bert For Sequence Classification model. Among the different methods in which the experiments were carried out, the most successful results were obtained with Bert.

It produces a toxicity score between 0 and 1 for each text sent to the trained Bert model. If the toxicity score of the text sent to the Bert model is between 0.7 and 1, this text contains high toxicity. It contains disturbing sentences that can be bad examples. This way, websites can control textual shares shared on their sites. Furthermore, it can automatically control whether the post stays in the ad or not.

REFERENCES

[1] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," International Journal of Machine Learning and Cybernetics, vol. 1, pp. 43–52, 12 2010.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762.pdf

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[4] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[6] C. Y. Peng, K. L. Lee, and G. Ingersoll, “An introduction to logistic regression analysis and reporting,” The Journal of Educational Research, vol. 96, pp. 14 – 3, 2002.

[7] L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 2004.

[8] T. Chen and C. Guestrin, “Xgboost,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug 2016. [Online]. Available: http://dx.doi.org/10.1145/ 2939672.2939785

[9] P. Soucy and G. Mineau, “A simple knn algorithm for text catego- rization,” in Proceedings 2001 IEEE International Conference on Data Mining, 2001, pp. 647–648.

[10] [Online].Available: https://www.kaggle.com/competitions/jigsaw-toxic-severity-rating/overview

[11] [Online].Available: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/overview

[12] G. M. Demirci, R. Keskin, and G. Do gan, “Sentiment analysis in turkish with deep learning,” in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 2215–2221.