# Using transformer architecture and natural language processing for detecting offensive content and cyberbullying

VOLODYMYR HALCHENKO, University of Twente, The Netherlands

The prevalence of offensive content and cyberbullying on the Internet has become an increasingly widespread issue. They can inflict emotional harm, instigate social isolation, and exacerbate mental health problems. Since content moderation is a labor-intensive task, machine learning might be helpful here. This research paper presents a comprehensive investigation of how Bidirectional Encoder Representations from Transformers (BERT) performs on the task of detecting offensive content and cyberbullying. The examination is done on how BERT suggests removing offensive content from a message while preserving the idea that a sender wants to express. Findings show that BERT requires fine-tuning to achieve high performance in detecting offensive content and cyberbullying. After fine-tuning, BERT gives useful suggestions on how to remove offensive content from messages while keeping the main idea of a person if offensive phrases are present in the context of a bigger main idea.

Additional Key Words and Phrases: transformers, BERT, ALBERT, BiLSTM, support vector machine, offensive content, cyberbullying, content moderation, natural language processing, artificial intelligence

## 1 INTRODUCTION

In the era of technological advancement, the Internet has emerged as a crucial platform for communication, commerce, education, and entertainment. While it offers numerous benefits, it is also replete with various problems, one of the most notable being the prevalence of offensive content. According to [28], offensive content is described as "any reported or publicized content like articles, films, photographs or websites that is probably going to be disturbing, abusive or offensive to a few people or a larger group". Detecting hate speech is of utmost importance due to the undeniable harm it causes [21, 22, 24]. The issue of hate speech is increasingly complex and multifaceted [2, 9, 18].

Cyberbullying is described as the use of electronic communication to harass or intimidate people. It has become a widespread problem among teenagers. Cyberbullying has a significant negative impact on a student's academic performance [25] and self-esteem [19]. While parents, educators, and governments are responsible to address all types of bullying, online platforms also play a significant role in this process [38]. That is why all platforms must moderate, no matter in which way, offensive or illegal content [36].

Cyberbullying is not easy to identify and moderate. Tone register, dynamics between involved actors, roles, context are often crucial to assess the true intent behind a message [6]. Abusive words can be used in a playful or friendly manner, which can lead to false positives [38]. To address these challenges, it is crucial to raise awareness about the issue and encourage the development of advanced technological solutions and policies which detect and mitigate cyberbullying more effectively.

Moderation is a difficult task due to its resource-intensive nature [36]. Human moderators cannot check everything because of the vastness of content that is published on online platforms [38]. Moderators can also be subject to psychological harm due to the heavy nature of the content they are exposed to regularly during their job [32]. In response to these difficulties, algorithmic moderation systems are deployed by major online platforms to regulate user-generated content [31]. There is a steady increase in research efforts to find effective ways to leverage machine-learning techniques intended to help automate the process of moderation [31, 38]. Considering this, the goals of this research can be defined as follows:

**Goal 1.** Enhance content moderation. Develop an AI-based solution to assist human moderators in identifying and filtering offensive content and cyberbullying across various digital platforms.

**Goal 2.** Investigate the role of linguistic factors in the perception of offensive content and cyberbullying, which can contribute to the development of more accurate and context-aware detection models.

To achieve these goals, the following research questions will be answered:

**Research Question 1.** How does BERT perform in detecting offensive content and cyberbullying?

**Research Question 2.** How does BERT isolate and remove offensive content from a message while preserving the main idea that a sender wants to express?

## 2 RELATED WORK

Extensive research has been conducted on how diverse models detect offensive content and cyberbullying, however, a clear and definitive solution has yet to be identified.

The k-nearest neighbour approach while identifying cyberbullying achieves 78.5% correctly labelled positives on the dataset created by gathering information from Formspring.me site [23]. There are experiments with models which achieve better results only under certain circumstances. The experiment shows that it may be the case the hybrid CNN-LSTM model can achieve high accuracy (90.4% for both macro-$F_1$ score and weighted-$F_1$ score on HASOC 2020 test dataset) in case the class distribution in the dataset is balanced [12]. The decision tree also can detect cyberbullying, however, there is a need to find the "best" size of the tree: a large, complex tree may be overfitted to the data, and a small and uncomplicated tree may suggest an imbalanced training dataset and, as a result, the model cannot be clearly identified [23]. A decision tree model, created by the C4.5 algorithm from the

*TScIT 39, July 7, 2023, Enschede, The Netherlands*

attributes provided, is able to identify messages containing cyberbullying (78.5% correctly labelled positives on the dataset created by gathering information from Formspring.me site) [23].

Since revealing cyberbullying requires a model which can learn long-distance semantic relations in the text, recurrent neural networks (RNNs) are not appropriate for this task. The transition function in RNNs has the problem of exploding and vanishing gradients: components of the gradient vector can grow or decay exponentially over long sequences of data during training [33, 42]. This issue is resolved in the long short-term memory (LSTM) architecture which enforces constant error flow [34]. However, a limitation of the basic LSTM architecture is in the strict sequential information propagation [20]. Tree-LSTM, a generalization of LSTMs to tree-structured network topologies, outperforms LSTMs in predicting the semantic relatedness of two sentences and sentiment classification [20]. There is still missing research on how Tree-LSTM models perform on the task of detecting offensive content and cyberbullying.

There is a very promising development of Bidirectional Encoder Representations from Transformers (BERT), which is a powerful encoder based on transformer architecture developed by Google in 2018 [15]. It is trained using a combination of masked language modelling and next sentence prediction tasks [15]. BERT can consider the meaning of both the preceding and following words in a sentence [15]. This makes it particularly effective at detecting cyberbullying and finding the true intent behind a message. The modification of BERT, RoBERTa [41], shows very good results on hate speech detection which have been carried out by [11, 29]. However, because of the very large batches required to train [41], RoBERTa is more computationally intensive to train than BERT, and it is unclear whether computationally difficult training is necessary for such type of task. There is also a lack of research on how transformed-based architecture behaves in the task of detection of long-distance semantic dependencies in the text and context-dependent meaning of the sentences.

## 3 DATASETS

The datasets used for training and evaluating BERT are taken from Toxic Comment Classification Challenge — the collection of publicly available comments from Wikipedia's talk page edits [7]. These datasets are designed for the development and evaluation of machine-learning models in identifying and classifying different types of offensive language. The datasets contain six categories of offensive content, where each category is defined by Perspective API [30] (the product of a collaborative research effort by Jigsaw and Google's Counter Abuse Technology team) as follows:

a) **Toxic**. "A rude, disrespectful, or unreasonable comment that is likely to make people leave a discussion."
b) **Severe toxic**. "A very hateful, aggressive, disrespectful comment or otherwise very likely to make a user leave a discussion or give up on sharing their perspective. This attribute is much less sensitive to more mild forms of toxicity, such as comments that include positive uses of curse words."
c) **Obscene**. "Obscene or vulgar language such as cursing."
d) **Threat**. "Describes an intention to inflict pain, injury, or violence against an individual or group." In this paper, threat is referred to as cyberbullying.
e) **Insult**. "Insulting, inflammatory, or negative comment towards a person or a group of people."
f) **Identity hate** (also referred to as identity attack). "Negative or hateful comments targeting someone because of their identity."

The distribution of the data is examined. The number of clean comments and offensive comments is shown in Figure 1 and the number of comments per offensive class is depicted in Figure 2.

The severe imbalance in the dataset is observed: the training dataset consists of 143346 clean messages and 16225 offensive messages. Such class imbalance can lead to the model which is biased towards the majority class and has an insufficient performance on the minority class. That is why for training the model only a random sample of 22500 clean messages is selected.

The severe imbalance in classes is also observed within offensive messages. Because the number of samples in the classes "Threat", "Identity hate", "Severe toxic" is very low, the assumption can be made that BERT will not reach good performance in detecting these types of offensive content: the limited number of samples will not give the opportunity to learn long-term semantic dependencies which are crucial to recognizing cyberbullying, hostility towards individuals based on their identity, the severe extent of toxicity.

The number of messages with a certain token count in the training dataset is depicted in Figure 3. For this task, considering the context and nature of messages, the best length for the encoding should be chosen to the maximum acceptable by BERT, which is 512 tokens. However, due to constraints in time and computational resources, the token length is reduced to 108. This choice is based on statistical data analysis, specifically the $75^{th}$ percentile of token length in the training dataset. The reduction in testing data to a sample of 10000 examples from comments which are equal to or less than 108 tokens aligns with the computational constraints to maintain consistency between the training and testing phases.

The data should be understood first to know how to pre-process it. After doing an analysis and considering the context, the following procedure has been carried out:

1. **Removing tab characters (\t, \v), newline characters (\n), carriage returns (\r).** Since Wikipedia talk page messages are usually complex with multiple sections or paragraphs, carriage returns, newline and tab characters have been removed with the goal of slightly increasing the contextual relationship between sentences for the model.
2. **Removing links**: Wikipedia talk page messages usually contain a significant number of links which refer to various resources. Since these links do not characterize the text, they have been removed.
3. **Removing special symbols and non-English letters.** Since the focus is to detect offensive content and cyberbullying, mathematical expressions and non-English letters only confuse the model. That is why they have been removed.
4. **Removing extra spaces:** some messages have spaces at the beginning and at the end. These spaces have been stripped. Within messages, multiple spaces are merged into a single space.
5. **Removing empty messages:** the first four pre-processing procedures removed all content in some messages. These messages have been deleted from the training and testing datasets.

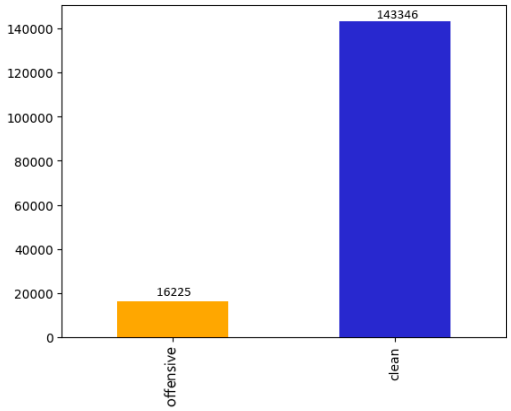


Fig. 1. The number of offensive and clean comments in the training dataset.

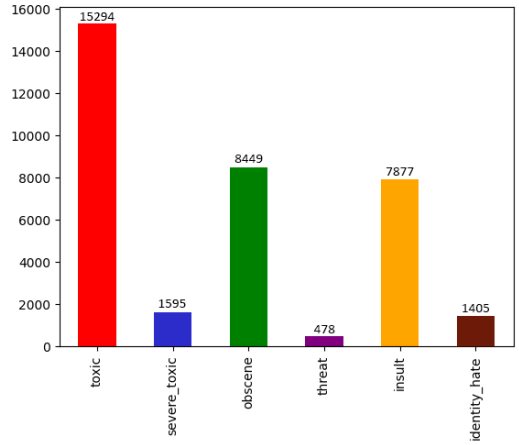


Fig. 2. The number of comments per offensive class in the training dataset.

6. **Removing data rows which are not used for scoring**. Some messages in the testing dataset come with all labels marked as -1. This value indicates that a data row is not used for scoring [7]. These data rows have been removed from the testing dataset.

To ensure fairness and eliminate any potential randomization or bias in the results, the training and testing of all models are performed on the same samples of data. This approach allows for a direct comparison of the performance between different models or configurations because it removes variability introduced by different samples.

## 4 EVALUATION

Evaluation of the performance of neural networks requires the consideration of multiple metrics. While a single metric can provide a limited perspective, a diverse set of metrics gives a comprehensive understanding of the network's capabilities and limitations. The metrics are calculated based on the following values:

*Definition 1.1.* **True Positive** (*TP*) — an instance where a model correctly predicts that an element belongs to a certain class.

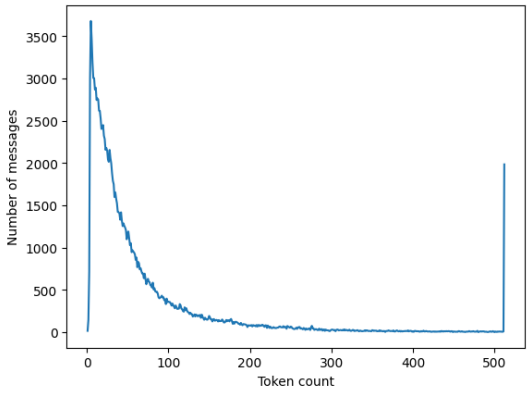


Fig. 3. The number of messages with a certain token count in the training dataset.

*Definition 1.2.* **True Negative** (*TN*) — an instance where a model correctly predicts that an element does not belong to a certain class.

*Definition 1.3.* **False Positive** (*FP*) — an instance where a model incorrectly predicts that an element belongs to a certain class.

*Definition 1.4.* **False Negative** (*FN*) — an instance where a model incorrectly predicts that an element does not belong to a certain class.

In this paper, the following evaluation metrics are used:

1. **Precision**. Precision is the fraction of relevant instances among the retrieved instances: within everything that **has been predicted as belonging to this class**, it measures the percentage of correct predictions. Precision demonstrates the ability to avoid over-predicting a certain class.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

2. **Recall**. Recall is the number of instances which the model correctly identified as relevant out of the total relevant instances: within everything that **does belong to this class**, it measures the percentage of correct predictions. Recall demonstrates the ability of a model to detect a certain class within a dataset.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

3. $\boldsymbol{F_1}$**-score**. $F_1$-score combines precision and recall into a single metric by taking the harmonic mean of precision and recall.

$$F_1 - \text{score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

It is important to note that the above metrics are used to indicate **per-class** performance. It means that these metrics for each class are calculated separately. However, instead of having multiple per-class scores, it is better to combine them and obtain numbers which describe the overall performance of a model. That is why the following generalization metrics are used:

4. **Macro average**. Macro-average is computed using the unweighted mean of all per-class $F_1$-scores.
5. **Weighted average**. Weighted average is calculated by taking the mean of all per-class $F_1$-scores while multiplying each $F_1$-score by the frequency of each class occurrence in the dataset.
6. **Accuracy in binary classification.** Accuracy is a metric for classification models that measures the number of predictions that are correct as a percentage of the total number of predictions that are made:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

7. **Micro average for multi-label classification.** Micro averaging outputs a global average $F_1$-score by considering the sums of True Positives (TP), False Negatives (FN), and False Positives (FP).

## 5 METHODOLOGIES

The objective of this research involves model training. The training is done in Python using Transformers library [37], Pandas library for data preparation, HuggingFace datasets for data extraction during the training and testing phases.

There are two methodologies on how to use a machine learning model in a specific task: ***feature extraction*** and ***fine-tuning*** [27]. In ***feature extraction***, model weights are "frozen", meaning that they do not update for a specific task [27]. In ***fine-tuning***, the pre-trained model parameters are updated on the task-specific corpus to adapt a model to a specific objective [27]. Both approaches have advantages and disadvantages: feature extraction is computationally cheap, however, a model, trained on a different task, may not optimally capture the most relevant features to a specific objective. Fine-tuning is computationally expensive, but it promotes a deeper understanding and better adaptation to the nuances of a particular task in most cases. Interestingly, [40] found an exception where BERT performance did not improve after fine-tuning.

For all tests, BERT base case-sensitive model (bert-base-cased) is used. BERT is a bare transformer outputting raw hidden states without any specific head on top, and it needs modifications to be used in a classification task. [15] discuss the approach of fine-tuning all model parameters because the self-attention mechanism allows BERT to be adapted to various downstream tasks. This represents a shift from previous methods which involve substantial model combinations or modifications for a specific task such as in [35]. That is why there is no need to make significant model modifications for the task of detecting offensive content and cyberbullying: it is enough to add a linear classifier layer without any pretraining on top of the model for all classification tasks. The training environment is described in Table 1.

It is important to explain the technique which is used in a model to make a decision. A common and straightforward approach in binary classification is to pick the class with the highest predicted probability. However, the decision boundary can be set based on the certain probability of the prediction of the certain class: for example, the decision is made that the message is offensive if the model outputs at least a 70% probability that the message is offensive. That is why the following metric is introduced:

Table 1. Training environment used in fine-tuning BERT

| Parameter | Value |
|---|---|
| Epochs | 2 |
| Batch size | 12 |
| Loss function | Binary cross entropy |
| Optimizer | Adam with weight decay [14] |
| Learning rate | $2 \cdot 10^{-5}$ |
| Learning rate scheduler warmup steps | 10% of the training data sample |

*Definition 2.1.* **Threshold** — a value controlling the decision boundary for class assignments.

Threshold adds an extra layer of precision: it allows to control the trade-off between precision and recall. For example, lowering the threshold might increase recall, capturing more true positives at the risk of including more false positives. In the following binary classification reports, the threshold is always attached to the probability that the message is offensive. The threshold corresponding to the classification report is specified.

## 6 BERT PERFORMANCE IN DETECTING OFFENSIVE CONTENT AND CYBERBULLYING

### 6.1 Classification with feature extraction

BERT has already been extensively trained on a large language corpus [15], and it has learned to capture the nuances and structures of language quite effectively. That is why usually there is no need to fine-tune the whole pre-trained model for specific tasks such as sentiment classification. Instead, only a classifier layer on top of the pre-existing BERT model is trained. The goal is to let the classifier layer learn the specifics of the task while leveraging the general language understanding that BERT has already acquired. The results for binary classification, where the model is tested on the task of predicting whether the message is offensive or not without specifying in which way the message is offensive, are presented in Table 2, and the results for multi-label classification are presented in Table 3.

Tables 2 and 3 show that feature extraction did not yield sufficient performance in both binary classification and multi-label classification. The assumption regarding the low performance of BERT in classes "Threat", "Identity hate", "Severe toxic" is confirmed: in feature extraction, the model cannot recognize these classes at all. [27] address the low performance to the high dissimilarity of the tasks on which the model is initially trained and the task on which it is operating. This might happen because the task of detecting offensive content and cyberbullying is highly dissimilar to the tasks on which BERT is initially trained. Another reason for poor performance could be specific linguistic features of the Wikipedia talk page comments that the base BERT model, pre-trained on a general language corpus, might not capture well enough.

Given these circumstances, it turns out that an extensive fine-tuning process is beneficial, one that involves not only the classifier layer but also the underlying layers of the pre-trained model. In this

Table 2. Classification report of BERT with feature extraction in binary classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Clean | 0.943 | 0.835 | 0.886 |
| Offensive | 0.319 | 0.605 | 0.418 |
| Macro average | 0.631 | 0.720 | 0.652 |
| Weighted average | 0.872 | 0.809 | 0.833 |
| **Common metrics** | | | |
| Accuracy | 0.809 | | |
| Threshold | 0.409 | | |

Table 3. Classification report of BERT with feature extraction in multi-label classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Toxic | 0.111 | 1.000 | 0.199 |
| Severe toxic | 0.000 | 0.000 | 0.000 |
| Obscene | 0.230 | 0.698 | 0.346 |
| Threat | 0.000 | 0.000 | 0.000 |
| Insult | 0.273 | 0.400 | 0.324 |
| Identity hate | 0.000 | 0.000 | 0.000 |
| Micro average | 0.141 | 0.699 | 0.234 |
| Macro average | 0.102 | 0.350 | 0.145 |
| Weighted average | 0.171 | 0.699 | 0.251 |
| **Common metrics** | | | |
| Threshold | 0.232 | | |

Table 4. Classification report of fine-tuned BERT in binary classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Clean | 0.973 | 0.939 | 0.956 |
| Offensive | 0.625 | 0.798 | 0.701 |
| Macro average | 0.799 | 0.869 | 0.828 |
| Weighted average | 0.934 | 0.923 | 0.927 |
| **Common metrics** | | | |
| Accuracy | 0.923 | | |
| Threshold | 0.99977 | | |

Table 5. Classification report of fine-tuned BERT in multi-label classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Toxic | 0.463 | 0.953 | 0.623 |
| Severe toxic | 0.520 | 0.265 | 0.351 |
| Obscene | 0.659 | 0.766 | 0.709 |
| Threat | 0.619 | 0.351 | 0.448 |
| Insult | 0.739 | 0.669 | 0.702 |
| Identity hate | 0.731 | 0.408 | 0.524 |
| Micro average | 0.551 | 0.790 | 0.650 |
| Macro average | 0.622 | 0.569 | 0.560 |
| Weighted average | 0.595 | 0.790 | 0.652 |
| **Common metrics** | | | |
| Threshold | 0.621 | | |

approach, the whole model can adjust its understanding to the peculiarities of the data, which in turn could potentially lead to a more robust and improved performance on the specific task.

## 6.2 Classification with fine-tuning

The fine-tuning and testing of the model are done in the same environment as described in Table 1. The results for binary classification are presented in Table 4 and the results for multi-label classification are presented in Table 5.

Tables 4 and 5 show a significant improvement in both binary classification and multi-label classification. The behaviour is subject to the following interpretation.

BERT is built using transformer architecture, and it uses the attention mechanism [15]. The attention mechanism operates on the database $D$ which consists of $m$ tuples of keys $k_i$ and values $v_i$. It accepts a query vector $\mathrm{q}$, and computes the output as a linear combination of values in the database via attention pooling, where the attention weight assigned to each value is determined by a compatibility function that measures the similarity between the query and the respective key [4, 5] and the number of similar observations which are available [5]:

$$\mathrm{Attention}(\mathrm{q}, D) = \sum_{i=1}^{m} a(\mathrm{q}, k_i) \cdot v_i \qquad (5)$$

The term “attention” is used because this operation focuses specifically on terms that have significant weights [5].

A sequence of tokens can be passed to the attention mechanism in such a way that each token has its own query, keys, and values at each step. This allows the token, using its query vector, to attend to each other token based on their key vectors when determining the representation of a token at the subsequent layer. With the complete set of query-key compatibility scores, we can construct a representation for each token by creating the appropriate weighted sum over the other tokens [5]. Because every token is paying attention to other tokens, this architecture is described as a **self-attention mechanism** [4, 5, 44].

BERT base is a bidirectional transformer encoder only which has 12 self-attention layers [15]. The attention pattern of transformer encoders allows all tokens to attend to one another freely [5]. BERT implements this peculiarity in bidirectional self-attention [15]: it gives all tokens the ability to pay attention to one another, and a token depends on input tokens before and after it in the sequence.

The explanation of the attention mechanism is needed to interpret why fine-tuning has significantly improved the performance of the model on this task. Text can be examined from diverse angles. The attention mechanism, used in BERT, can capture diverse text features; however, it may not always prioritize specific features necessary for a given task. Since the data used for the pre-training of BERT [15] does not involve the detection of offensive content and cyberbullying, the assumption is that BERT during feature extraction does not know how to focus on this task. During fine-tuning, the bidirectional encoder adjusts its weights in such a way that it starts paying attention to the features of the text which indicate whether the message is offensive or not. This is done by drastically changing the attention mode of the last layers [3, 26, 39, 40] and modifying the feature extraction mode of intermediate and last layers [39]; the attention mode of the first layers is not changed significantly [3, 26, 39, 40]. Higher layers do not change arbitrarily; instead, they remain similar to the layers in the untuned model [40]. In the attention mechanism, BERT preserves the original spatial structure of the data points while

adjusting the space to suit the particular task [40]. If task labels are not linearly separable, BERT groups data points with the same label into a few clusters (ideally one) during fine-tuning [40]. This makes it easier to linearly separate labels with fine-tuned representations compared to untuned ones. If the task labels in a representation are already linearly separable, BERT pushes clusters of points that represent different labels away from each other, creating large separation regions between labels. Rather than simply scaling the points to make a separation space, BERT moves clusters in various directions and with different extents measured by Euclidean distance [40]. Simultaneously, BERT prevents the complete loss of learned knowledge in the lower layers, a phenomenon known as catastrophic forgetting [39].

If the task of detecting offensive content and cyberbullying, it is safe to hypothesize that BERT first groups data points with the same offensive type in clusters representing different types of offensive content, and then pushes these clusters away from each other which significantly improves its performance. The preservation of the original spatial structure of the data points does not significantly change the general meaning of the words and phrases; instead, BERT adapts the attention to the given task such that each token attends other tokens in order to generate a representation of the sentence that is sensitive to the presence of the offensive content. However, the possible issue is that the true types of toxicity may be grouped into clusters which are located close to each other; for example, "Toxic" and "Severe toxic" types have many similarities: the distinction between these types is in the intent behind using curse words [30] which may not be easy to linearly separate in the attention mechanism.

### 6.3 ALBERT: a lightweight version of BERT

While BERT has set impressive performance benchmarks across numerous tasks, its considerable size can pose significant computational challenges in practice in case fine-tuning is required, like in detecting offensive content and cyberbullying. In scenarios where training resources are limited or efficiency is paramount, ALBERT [43] stands out as an ideal candidate. ALBERT is a lightweight version of BERT which trains faster and scales much better than BERT due to two parameter-reduction techniques: **factorized embedding parameterization** and **cross-layer parameter sharing** [43]. That is why ALBERT is tested using the same training parameters presented in Table 1, and its performance with BERT is compared. The results of the binary classification are shown in Table 6 and the results of the multi-label classification are shown in Table 7.

Table 6 and Table 7 show that ALBERT performs slightly worse than BERT in binary classification: $F_1$ -score of the offensive content, which is the subject of interest, is 1.8% lower. In multi-label classification, the performance per-class is different; however, looking at the classification report in general, it is safe to assume that BERT and ALBERT achieve almost the same performance. Considering the difference in time and resources spent for fine-tuning BERT and ALBERT, the outcomes are very good.

### 6.4 Comparison to Bidirectional LSTM

Various neural network models have been widely used in the field of natural language processing. Long Short-Term Memory (LSTM) [34] and Bidirectional Long Short-Term Memory (BiLSTM) [1] are promising candidates for the task of text classification [10, 16, 35]. They capture the sentiment in the text, making them particularly efficient for emotion classification. The ability of BiLSTM to capture the meaning from both sides of a token and remember long-term dependencies in the text makes them a robust choice for this task.

Table 6. Classification report of fine-tuned ALBERT in binary classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Clean | 0.979 | 0.919 | 0.948 |
| Offensive | 0.573 | 0.846 | 0.683 |
| Macro average | 0.776 | 0.883 | 0.816 |
| Weighted average | 0.933 | 0.911 | 0.918 |
| **Common metrics** | | | |
| Accuracy | 0.911 | | |
| Threshold | 0.99959 | | |

Table 7. Classification report of fine-tuned ALBERT in multi-label classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Toxic | 0.461 | 0.950 | 0.620 |
| Severe toxic | 0.667 | 0.245 | 0.358 |
| Obscene | 0.678 | 0.748 | 0.711 |
| Threat | 0.750 | 0.486 | 0.590 |
| Insult | 0.761 | 0.613 | 0.679 |
| Identity hate | 0.800 | 0.400 | 0.533 |
| Micro average | 0.555 | 0.772 | 0.646 |
| Macro average | 0.686 | 0.574 | 0.582 |
| Weighted average | 0.612 | 0.772 | 0.648 |
| **Common metrics** | | | |
| Threshold | 0.614 | | |

A comparative study is conducted to ascertain whether BERT outperforms BiLSTM in the task of detecting offensive content and cyberbullying. The pre-trained GloVe embedding [17] with a vector size of 50 is used as a starting point for fine-tuning. The architecture and the environment are described in Table 8. The results of binary classification are presented in Table 9, and the results of multi-label classification are presented in Table 10.

Table 8. BiLSTM architecture and environment used in fine-tuning

| **Parameter** | **Value** |
|---|---|
| BiLSTM layers | 1 |
| BiLSTM units per layer | 100 |
| Epochs | 8 |
| Batch size | 12 |
| Loss function | Binary cross entropy |
| Optimizer | Adam with weight decay [14] |
| Learning rate | $2 \cdot 10^{-5}$ |

Tables 4, 5, 6, 7, 9, 10 show that both BERT and ALBERT have successfully outperformed BiLSTM in the detection of offensive content and identifying cyberbullying. The superior performance of BERT and ALBERT does not diminish the capabilities of BiLSTM; it rather highlights the rapid advancements in the field of deep

Table 9. Classification report of BiLSTM in binary classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Clean | 0.934 | 0.774 | 0.846 |
| Offensive | 0.244 | 0.570 | 0.341 |
| Macro average | 0.589 | 0.672 | 0.594 |
| Weighted average | 0.855 | 0.751 | 0.789 |
| **Common metrics** | | | |
| Accuracy | 0.751 | | |
| Threshold | 0.470 | | |

Table 10. Classification report of BiLSTM in multi-label classification

| | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Toxic | 0.160 | 0.552 | 0.248 |
| Severe toxic | 0.007 | 0.224 | 0.014 |
| Obscene | 0.107 | 0.353 | 0.164 |
| Threat | 0.004 | 0.162 | 0.008 |
| Insult | 0.100 | 0.341 | 0.155 |
| Identity hate | 0.024 | 0.300 | 0.044 |
| Micro average | 0.088 | 0.427 | 0.146 |
| Macro average | 0.067 | 0.322 | 0.106 |
| Weighted average | 0.121 | 0.427 | 0.187 |
| **Common metrics** | | | |
| Threshold | 0.409 | | |

learning, showcasing the exciting prospects for future research and applications.

## 7 BERT PERFORMANCE ON SUGGESTING ON HOW TO REMOVE OFFENSIVE CONTENT FROM A MESSAGE

BERT was pre-trained using the masked language modelling approach [15] in which parts of sentences were hidden with a special token [MASK] and the model was asked to predict hidden words. This distinctive pre-training method allows using BERT for such tasks as completing sentences within a given context. In the task of removing offensive content from the message, BERT has the potential to replace toxic expressions with polite alternatives using this procedure.

BERT requires all offensive content to be explicitly hidden. To do this, the pre-trained support vector machine for detecting profanity and toxicity [8] is used for token classification with a threshold of 0.500. The words and phrases which are considered by the support vector machine as offensive are replaced with a single [MASK] token. Then two approaches are used for sentence correction.

1. **Context-unaware mask filling**: this approach uses feature extraction of BERT on Wikipedia talk page messages. The model without fine-tuning is immediately asked to predict the hidden parts of the sentences.
2. **Context-aware mask filling**: this approach involves fine-tuning BERT on the dataset in the training environment described in Table 11. The goal is to familiarize the model with the linguistic features of Wikipedia talk pages so it can learn the peculiarities of the style used in Wikipedia discussions. **The model is fine-tuned only on non-offensive comments since it shall not be taught any harmful language.** Then it is asked to predict the hidden parts of the sentences.

Table 11. Training environment used in fine-tuning BERT for masked language modelling

| **Parameter** | **Value** |
|---|---|
| Epochs | 1 |
| Optimizer | Adam with weight decay [14] |
| Learning rate | $10^{-5}$ |
| Learning rate scheduler warmup steps | 10% of the training data sample |

Both models, base and fine-tuned, are tested on the same sample of data. The data entries, in which the output of the models is different, are saved in a separate dataset. Then 700 randomly chosen comments are manually examined. The observations lead to the following:

1. **Removing offensive language from the context.** The fine-tuned model removes offensive language from the context much better than the base model. This may be interpreted by the data preparation stage: since the model is fine-tuned only on clean messages, it does not know about toxicity, hate speech, insults in Wikipedia discussions, and it can only effectively insert polite phrases, which it does in practice.
2. **Removing offensive language if the whole message is offensive.** Both BERT models, base and fine-tuned, do not remove offensive words and phrases if the whole message is offensive. This may be interpreted by the principle of the attention mechanism: if the idea of the message is to threaten, offend or humiliate, tokens capture the harmful idea while freely attending to each other, and, therefore, BERT inserts a toxic word. If the offensive content is in the context of another idea, BERT captures the main idea and replaces offensive words and phrases with polite alternatives.
3. **Inserting non-English letters or symbols.** The base model quite often inserts a symbol while the fine-tuned model rarely ever does it. This may be interpreted by the data preparation stage: since special symbols and non-English letters were removed from the training dataset, the model during fine-tuning learned that in this context special symbols are not relevant.
4. **Inserting punctuation marks.** Unlike a fine-tuned model, the base model often inserts a dot in the middle of a sentence which splits it into two sentences which do not make any sense, or it replaces a word with an exclamation mark, question mark etc. which breaks the meaning of the sentence. The fine-tuned model does it much less often. This may be interpreted by the procedure of fine-tuning: the model during additional training learned that in this context the aim is to fill in text with words or phrases, but not with punctuation marks.
5. **Possible distortion of message meaning.** Both BERT models, base and fine-tuned, sometimes may insert such words or phrases after which the meaning of a secondary idea is lost or distorted. This is interpreted by the attention mechanism: since it focuses on the main idea, it does not capture the meaning of a secondary idea which has a toxic phrase. Therefore, BERT preserves only the main thought of the sentence, but not the secondary thoughts surrounding it.

6. **Performance on messages written in uppercase letters.** Both BERT models, base and fine-tuned, show poor performance on messages written in uppercase letters. This may be interpreted by the case sensitivity of the models: the same sentence, written normally and in uppercase, is treated differently by bert-base-cased. Since there are few messages and phrases written in uppercase, and they usually do not have a semantic load, both models do not learn much from these sentences.
7. **Word duplication.** Both BERT models, base and fine-tuned, might insert the word which is located to the left or to the right of the [MASK] token. However, the fine-tuned model does it less often. This may be interpreted by the style of Wikipedia messages: since most of the comments have similar semantic meaning, such as a dispute regarding the cancellation of a controversial edit, it is easier for the fine-tuned model to find an appropriate replacement.

The observations suggest that the fine-tuned model outperforms the base model with feature extraction. It has a better understanding of Wikipedia talk page comments, leading to its improved performance in removing offensive content and replacing it with polite phrases.

## 8 LIMITATIONS

Several opportunities for improvement were identified because of computational constraints. The token size is the first improvement for further research. Larger token sizes better capture the meaning of long sentences, but they require more computational power. Due to constraints in time and computational resources, the token length was reduced to 108, which corresponds to the 75$^{th}$ percentile of the token count in the training dataset, and the models could not perform well on the rest of the 25% comments. This problem is especially relevant to the examples where offensive content or cyberbullying is located at the end of a long message: the tokenizer simply cannot capture this part of the message. Since Wikipedia messages are usually long, the best length for this task should be chosen to the maximum acceptable by BERT, which is 512 tokens.

The number of training epochs is another crucial aspect to consider. Due to constraints in time and computational resources, the epochs are reduced to the fixed number of 2 while training BERT and ALBERT. For scientific research, it is desirable to continue fine-tuning a model until there is no improvement in its performance. However, training for too many epochs can lead to overfitting of the model to the training data, which results in excellent performance on the training dataset and poor performance on the testing dataset [5, 13]. In other words, BERT, instead of learning how to identify offensive content and cyberbullying, learns by heart the training data. That is why the early stopping shall be used to ensure that the training stops when there is no improvement.

## 9 CONCLUSIONS

The findings of this research demonstrate the behaviour of Bidirectional Encoder Representations from Transformers (BERT) in detecting offensive content and identifying cyberbullying. The investigation is done on how BERT behaves in suggesting how to remove offensive content from a message while preserving the idea that a sender wants to express. The results presented in this paper demonstrate that BERT achieves strong performance in all these tasks using fine-tuning approach.

It is established that BERT achieves high performance in detecting offensive content and cyberbullying both in binary classification and in multi-label classification after fine-tuning, and feature extraction proves to be a less effective approach in these tasks. This can be attributed to the attention mechanism of BERT which might not focus on the features of the comments which are crucial for detecting offensive content and cyberbullying. BERT requires fine-tuning to adapt the attention of its intermediate and last layers to focus on these features. In case there is a constraint in computational resources or time, ALBERT — a lite version of BERT — can be used for the detection of offensive content and cyberbullying. BERT has stepped ahead, outperforming its predecessor BiLSTM in detecting offensive content and cyberbullying, and opened a promising path in natural language processing.

BERT gives high-quality suggestions on how to remove offensive content from messages while preserving the main idea that a sender wants to express if offensive elements are present in the context of the main idea. However, when the primary intent of a message is to threaten, offend, or humiliate, BERT does not remove offensive content and cyberbullying. This is interpreted mostly by the focus of the attention mechanism: if offensive content is in the context of the bigger main idea, BERT captures the main idea and replaces offensive words and phrases with polite alternatives. However, in instances where the entire message is intended to harm, insult, or discriminate, the focus of the attention mechanism concentrates on the offensive content, leading to a failure in its removal. The study reveals that BERT, fine-tuned on clean messages only, removes offensive content more effectively than the basic BERT. This happens because fine-tuning directs BERT's attention to specific topics of discussion, thereby making it easier for a model to find polite words. The complete absence of offensive content during fine-tuning modifies the layers of the model, reducing their attention to toxic phrases and enhancing their capability to remove harmful speech in instances where the base model concentrates on offensive content and does not remove it.

These findings shed light on both the capabilities and the limitations of BERT in the realm of offensive content detection and mitigation. Future work can focus on applying more complex approaches of fine-tuning and examining how clusters with different offensive labels move inside BERT attention layers to further enhance BERT performance in this field.

## ACKNOWLEDGMENTS

The author thanks Nacir Bouali and Faizan Ahmed for supervising this research project.

## REFERENCES

[1] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5-6 (July–August 2005), 602-610. https://doi.org/10.1016/j.neunet.2005.06.042

[2] Alexander Brown. 2018. What is so special about online (as compared to offline) hate speech? *Ethnicities* 18, 3 (June 2018), 297-326. https://doi.org/10.1177/1468796817709846

[3] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What Happens To BERT Embeddings During Fine-tuning? arXiv:2004.14448. Retrieved June 19, 2023 from https://arxiv.org/abs/2004.14448

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762. Retrieved May 29, 2023 from https://arxiv.org/abs/1706.03762

[5] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2023. Dive into Deep Learning. Retrieved May 29, 2023 from https://d2l.ai/

[6] Chris Emmery, Ben Verhoeven, Guy De Pauw, Gilles Jacobs, Cynthia Van Hee, Els Lefever, Bart Desmet, Vēronique Hoste, and Walter Daelemans. 2020. Current limitations in cyberbullying detection: On evaluation criteria, reproducibility, and data scarcity. *Language Resources and Evaluation* 55 (2021), 597-633. https://doi.org/10.1007/s10579-020-09509-1

[7] Conversation AI. 2017. Toxic Comment Classification Challenge. (December 19, 2017). Retrieved April 26, 2023 from https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge

[8] Dimitrios Mistriotis and Menelaos Kotoglou. 2023. Alt-profanity-check. Retrieved May 20, 2023 from https://github.com/dimitrismistriotis/alt-profanity-check

[9] Eric Barendt. 2019. What Is the Harm of Hate Speech? *Ethical Theory and Moral Practice* 22 (2019), 539–553. https://doi.org/10.1007/s10677-019-10002-0

[10] Gang Liu and Jiabao Guo. 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* 337 (14 April 2019), 325-338. https://doi.org/10.1016/j.neucom.2019.01.078

[11] György Kovács, Pedro Alonso, and Rajkumar Saini. 2021. Challenges of Hate Speech Detection in Social Media. *SN Computer Science* 2, 95 (2021). https://doi.org/10.1007/s42979-021-00457-3

[12] György Kovács, Pedro Alonso, Rajkumar Saini, and Marcus Liwicki. 2022. Leveraging external resources for offensive content detection in social media. *AI Communications* 35, 2 (2022), 87-109. http://dx.doi.org/10.3233/AIC-210138

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. Massachusetts Institute of Technology Press, USA.

[14] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101. Retrieved May 10, 2023 from https://arxiv.org/abs/1711.05101

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805v2. Retrieved April 27, 2023 from https://arxiv.org/abs/1810.04805

[16] Jakub Nowak, Ahmet Taspinar, and Rafał Scherer. 2017. LSTM Recurrent Neural Networks for Short Text and Sentiment Classification. In *Proceedings of the 16th International Conference on Artificial Intelligence and Soft Computing (ICAISC 2017)*, June 11-15, 2017, Zakopane, Poland. Springer International Publishing, 553-562. https://doi.org/10.1007/978-3-319-59060-8_50

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. (August 2014). Retrieved June 8, 2023 from https://nlp.stanford.edu/projects/glove/

[18] Jeremy Waldron. 2012. The Harm In Hate Speech. Harvard University Press, USA.

[19] Justin W. Patchin and Sameer Hinduja. 2010. Cyberbullying and Self-Esteem. *Journal of School Health* 80, 12 (December 2010), 614-621. https://dx.doi.org/10.1111/j.1746-1561.2010.00548.x

[20] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. arXiv:1503.00075. Retrieved April 28, 2023 from https://arxiv.org/abs/1503.00075

[21] Karsten Muller and Carlo Schwarz. 2020. Fanning the Flames of Hate: Social Media and Hate Crime. *Journal of the European Economic Association* 19, 4 (August 2021), 2131-2167. https://doi.org/10.1093/jeea/jvaa045

[22] Katharine Gelber and Luke McNamara. 2015. Evidencing the harms of hate speech. *Social Identities* 22, 3 (2016), 324-341. https://doi.org/10.1080/13504630.2015.1128810

[23] Kelly Reynolds, April Kontostathis, and Lynne Edwards. 2011. Using Machine Learning to Detect Cyberbullying. In *2011 10th International Conference on Machine Learning and Applications and Workshops (ICMLA)*, December 18-21, 2011, Honolulu, HI, USA. IEEE, 241-244. https://doi.org/10.1109/ICMLA.2011.152

[24] Koustuv Saha, Eshwar Chandrasekharan, and Munmun De Choudhury. 2019. Prevalence and Psychological Effects of Hateful Speech in Online College Communities. In *Proceedings of the 10th ACM Conference on Web Science (WebSci'19)*, June 30 – July 3, 2019, Boston, MA, USA. Association for Computing Machinery, New York, NY, 255-264. https://doi.org/10.1145/3292522.3326032

[25] Maham Muzamil and Gulzar Shah. 2016. Cyberbullying and self-perceptions of students associated with their academic performance. *International Journal of Education and Development using Information and Communication Technology* 12, 3 (January 2016), 79-92.

[26] Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. On the Interplay Between Fine-tuning and Sentence-Level Probing for Linguistic Knowledge in Pre-Trained Transformers. arXiv:2010.02616. Retrieved June 30, 2023 from https://arxiv.org/abs/2010.02616

[27] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. arXiv:1903.05987. Retrieved June 8, 2023 from https://arxiv.org/abs/1903.05987

[28] Naman Deep Srivastava, Sakshi, and Yashvardhan Sharma. 2020. Combating Online Hate: A Comparative Study on Identification of Hate Speech and Offensive Content in Social Media Text. In *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, December 3-5, 2020, Thiruvananthapuram, India. IEEE, 47-52. http://dx.doi.org/10.1109/RAICS51191.2020.9332469

[29] Pedro Alonso, Rajkumar Saini, and György Kovács. 2020. Hate Speech Detection Using Transformer Ensembles on the HASOC Dataset. In *Speech and Computer, 22nd International Conference (SPECOM 2020)*, October 7-9, 2020, St. Petersburg, Russia. Springer Nature, Cham, Switzerland, 13–21. https://doi.org/10.1007/978-3-030-60276-5_2

[30] Perspective Developers. 2023. Perspective API. Attributes & Languages. Retrieved May 3, 2023, from https://developers.perspectiveapi.com/s/about-the-api-attributes-and-languages?language=en_US

[31] Robert Gorwa, Reuben Binns, and Christian Katzenbach. 2020. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society* 7, 1 (January-June 2020), 1-15. https://doi.org/10.1177/2053951719897945

[32] Sarah T. Roberts. 2019. *Behind the Screen. Content Moderation in the Shadows of Social Media.* Yale University Press, New Haven, USA.

[33] Sepp Hochreiter. 1998. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6, 2 (1998), 107–116. https://doi.org/10.1142/S0218488598000094

[34] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (15 November 1997), 1735-1780. https://dx.doi.org/10.1162/neco.1997.9.8.1735

[35] Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. 2017. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications* 72 (15 April 2017), 221-230. https://doi.org/10.1016/j.eswa.2016.10.065

[36] Tarleton Gillespie. 2018. *Custodians of the Internet. Platforms, Content Moderation, and the Hidden Decisions That Shape Social Media.* Yale University Press, New Haven, USA.

[37] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771. Retrieved June 2, 2023 from https://arxiv.org/abs/1910.03771

[38] Tijana Milosevic, Kathleen Van Royen, and Brian Davis. 2022. Artificial Intelligence to Address Cyberbullying, Harassment and Abuse: New Directions in the Midst of Complexity. *International Journal of Bullying Prevention* 4 (2022), 1–5. http://dx.doi.org/10.1007/s42380-022-00117-x

[39] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Investigating Learning Dynamics of BERT Fine-Tuning. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, December 2020, Suzhou, China. Association for Computational Linguistics, 87-92.

[40] Yichu Zhou and Vivek Srikumar. 2022. A Closer Look at How Fine-tuning Changes BERT. arXiv:2106.14282. Retrieved June 1, 2023 from https://arxiv.org/abs/2106.14282

[41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692. Retrieved April 27, 2023 from https://arxiv.org/abs/1907.11692

[42] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* 5, 2 (March 1994), 157-166. https://doi.org/10.1109/72.279181

[43] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv:1909.11942. Retrieved May 30, 2023 from https://arxiv.org/abs/1909.11942

[44] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. arXiv:1703.03130. Retrieved June 19, 2023 from https://arxiv.org/abs/1703.03130