

# Attention Based BERT-FastText model for Hate Speech and Offensive Content Identification in English and Hindi Languages

Krishanu Maity<sup>1</sup>, Abhishek Kumar<sup>1</sup> and Sriparna Saha<sup>1</sup>

<sup>1</sup>Indian Institute of Technology, Patna

## Abstract

This paper describes our model submitted for HASOC-2021 as the IIT\_Patna team for hate and offensive content identification in English and Hindi languages. A deep learning model, namely BERT+FastText-GRU, has been developed based on BERT and FastText, followed by GRU with attention. Our proposed model uses a BiGRU-based deep neural network to extract textual features, followed by an Attention layer to focus on the most important phrase of the text. The BERT language model and FastText embedding have been employed to examine the effectiveness of joint embedding representation compared to a single one. We have set up some baselines by varying the RNN architecture (LSTM/GRU) and the word vector representation approach (BERT/FastText). Our model outperforms all the baselines with the highest accuracy values of 76.32% for subtask-1A (EN), 56.73% for subtask-1B (EN), 69.17% for subtask-1A (HI) and 40.45% for subtask-1B (HI).

## Keywords

BERT, FastText, Attention, Hate Speech, Offensive Content

## 1. Introduction

With the progress of technology and the growing popularity of many social media platforms, the number of people active on these platforms has risen dramatically. The majority of the time, these individuals abuse their right to free of speech and violate the forums' permissible usage standards. This has prompted the detection of any offensive or obscene posts, comments, photos, or other content and the prevention of further distribution in order to limit the impact on social media. On social media, user-generated content is not always structured according to the standards. In reality, foul language content has been common on social media in recent years [1, 2]. Some terms have various meanings that may be objectionable to some individuals in some locations. On social media messages that are largely offensive, there is an increasing demand for foul language identification. Social media creates a significant amount of data on a daily basis. As a result, even an expert will find it impossible to manually detect inappropriate language on social media. At this point, effective techniques are required to monitor the content on social media. TRAC 1, 2018 (related to Aggression Identification) [3], GermEval Task 2 [4], SemEval 2019 Task 5 [5], HASOC 2019 [6], HASOC 2020 [7] and OffensEval 2019 Task [8] are some of the related tasks. Recent research has looked into classifying hate speech into


---

Forum for Information Retrieval Evaluation, December 13-17, 2021, India

✉ [krishanu\\_2021cs19@iitp.ac.in](mailto:krishanu_2021cs19@iitp.ac.in) (K. Maity); [abhishek.km23@gmail.com](mailto:abhishek.km23@gmail.com) (A. Kumar); [sriparna@iitp.ac.in](mailto:sriparna@iitp.ac.in) (S. Saha)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings ([CEUR-WS.org](http://CEUR-WS.org))

sub-categories such as abusive, aggressive, or insulting speech. Such classification of social media messages aids law enforcement authorities in social media surveillance. There are three sub-tasks in HASOC-2021 [9]. In *subtask-1A*, we have classified data into hate and offensive labels or not hate and offensive labels, and in *subtask-1B*, we have classified data into Profane, Hate, Offensive and None labels. These classifications are for both English (EN) and Hindi (HI) languages. The goal of *Subtask 2* is identifying Conversational Hate-Speech in Code-Mixed Languages (ICHCL).

In this work, we have developed a deep learning model(BERT+FastText-GRU) based on BERT and FastText followed by GRU with attention to solve subtask-1A and subtask-1B in both Hindi and English languages [10]. Our proposed model outperforms all base lines with the highest f1 score of 75.58%, 56.52%, 68.48% and 37.82% for four tasks, i.e, subtask-1A (EN), subtask-1B (EN), subtask-1A (HI) and subtask-1B (HI), respectively.

## 2. Related Work

Researchers have been studying and reporting their findings and observations linked to online abuse of social media platforms for quite some time now like in [11]. The task of identifying objectionable languages in Arabic was discussed in by Authors in [12]. A machine learning technique for identifying abusive language on Twitter data is used by authors in [13]. With the Foul Greek Tweet Dataset, this study employs a variety of machine learning and deep learning models to identify offensive language. Misuse in the form like cyberbullying have been discussed in [14], trolling have been discussed in [15] and offensive language in [16]. Authors in [17] presented a set of CNN-based deep neural models for categorising tweets into four categories: sexism, racism, either (sexism or racism) and non-hate. Racism, sexism, and a non-hate-speech categorization system are all included in the Twitter Hate Speech document. Authors in [18] have documented the use of word n-grams and emotion lexicons. Various linguistic, lexical, emotion, surface, and other characteristics that may be used to create a classifier for detecting hate speech were discovered in a comprehensive study in [19]. For hate speech identification, a CNN and GRU-based method was presented by authors in [20]. A fascinating study was conducted on forecasting future animosity and its severity based on the existing circumstances by authors in [21]. Despite the fact that the majority of works on offensive language and hate speech have been written in English, there are a few works in other languages as well. Authors in [22] used neural networks and advanced attention mechanisms to work on a huge dataset of Greek Sports Comments and presented different ways to manage user content moderation.

### 2.1. Problem definition

The task in HASOC 2021 is divided into sub tasks based on the kind and target of offences. Detailed train and test set distributions of subtask-1A and subtask-1B in both Hindi and English languages are shown in Table 1.

- **Subtask-1A : Hate and Offensive labels detection** Here we have to categorize between hate and offensive and non hate and offensive tweets. The class labels are HOF and NOT. This is for both English and Hindi languages. In Hindi subtask-1A train data, there are

**Table 1**  
Data set description

| Task Description                  | Training Set | Test Test |
|-----------------------------------|--------------|-----------|
| subtask-1A (EN) & subtask-1B (EN) | 3843         | 1281      |
| subtask-1A (HI) & subtask-1B (HI) | 4594         | 1532      |

total of 4594 instances, of which 3131 instances are of not offensive while 1433 instances are offensive. In English subtask-1A train data, there are total 3843 instances of which 2501 instances are of Hate & offensive while 1342 instances are of Not hate & offensive.

- **Subtask-1B : Offensive labels detection** Here we have to categorize between Profane, Hate, Offensive and none labels on tweets. The class labels are PRFN, HATE, OFFN and NONE. This is also for both English and Hindi languages. In Hindi sub task-1B train data, there are 3161 instances of None, 654 instances of Offensive, 566 instances of Hate and 213 instances of Profane. In English sub task-1B, there are 1342 instances of None, 1196 instances of Profane, 683 instances of Hate and 622 instances of Offensive.

### 3. Methodology for Cyberbullying Detection

This section describes an attention based framework we have developed to identify hatespeech and Offensive Content in English and Hindi Languages. Figure 1 depicts the overall architecture of our proposed *Bert+FastText-GRU* model.

#### 3.1. BERT

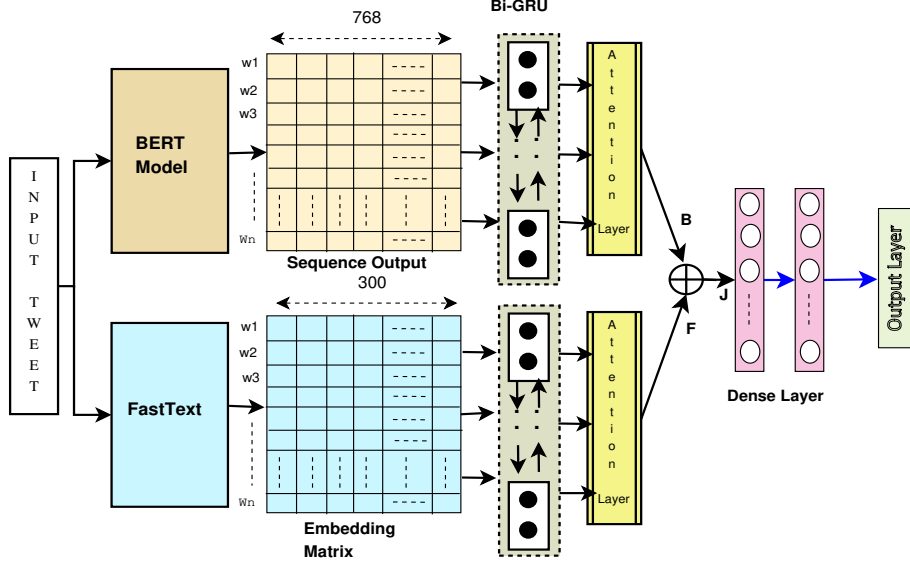
BERT [23] is a Transformer-based [24] language model. Fine-tuning BERT has shown a decent improvement in solving several Natural Language Processing (NLP) tasks like text classification, question answering, machine translation etc. BERT has different variances like BERT base model, Multilingual BERT (M-BERT), medical BERT, etc. For English language based task we have considered BERT base model<sup>1</sup>. We choose the M-BERT<sup>2</sup> for the Hindi language task since it has been trained on 104 languages, including Hindi.

#### 3.2. FastText

The Facebook Research Team developed FastText[25] for efficient word embedding of more than 157 different languages. FastText model was trained using CBOW technique with character n-grams of length 5, a window of size 5, and 10 negatives, and it returns 300 dimensional dense vector corresponding to each token. In a social media text, the inclusion of Out-of-Vocabulary (OOV) words is a severe issue. To evade automatic checking, users in social media often perform

<sup>1</sup>[https://tfhub.dev/google/bert\\_uncased\\_L-12\\_H-768\\_A-12/1](https://tfhub.dev/google/bert_uncased_L-12_H-768_A-12/1)

<sup>2</sup>[https://tfhub.dev/google/bert\\_multi\\_cased\\_L-12\\_H-768\\_A-12/1](https://tfhub.dev/google/bert_multi_cased_L-12_H-768_A-12/1)



**Figure 1:** Bert+FastText-GRU architecture.

intentional obfuscation of words by using short words, abbreviations, and misspelled words. Such words are not represented in the pre-trained word embedding model, resulting in the loss of morphological information. FastText utilizes the character level in represented words into the vector, unlike word2vec [26] and Glove [27], which use word-level representations. We have considered FastText Hindi and English word embeddings for Hindi and English subtasks respectively.

### 3.3. Bi-directional GRU Layer

To learn the contextual information of input tweet from both the directions, the word vectors from both BERT and FastText are passed through two separate Bidirectional GRUs [28] layer. To capture long-term dependencies in the tweet, bi-directional GRU sequentially encodes these feature map into hidden states as,

$$\vec{h}_t^i = \overrightarrow{GRU}(w_t^i, h_{t-1}^i), \overleftarrow{h}_t^i = \overleftarrow{GRU}(w_t^i, h_{t+1}^i) \quad (1)$$

where each word vector  $w_t^i$  of sentence  $i$  is mapped to a forward hidden state  $\vec{h}_t^i$  and backward hidden state  $\overleftarrow{h}_t^i$  by invoking  $\overrightarrow{GRU}$  and  $\overleftarrow{GRU}$ , respectively.  $\vec{h}_t^i$  and  $\overleftarrow{h}_t^i$  are combined to form  $h_t^i$ , which is a single hidden state representation.

$$[h_t^i = \vec{h}_t^i, \overleftarrow{h}_t^i] \quad (2)$$

### 3.4. Attention Layer

The basic principle behind the attention mechanism [29] is to give greater weight to the words that contribute the most to the meaning of the phrase. To produce an attended sentence vector,

we leverage the attention mechanism on the Bi-GRU layer's output. Specifically,

$$u_t^i = \tanh(W_w h_t^i + b_w) \quad (3)$$

$$\sigma_t^i = \frac{\exp(u_t^i T u_w)}{\sum_t \exp(u_t^i T u_w)} \quad (4)$$

$$S_i = \sum_t (\sigma_t^i * h_t^i) \quad (5)$$

Where  $u_t^i$  is the hidden representation of  $h_t^i$  and  $u_w$  is the context vector.  $S_i$  is the output generated by attention layer and attention weight for a particular word is  $\sigma_t^i$ .

### 3.5. Loss Function

As a loss function, we have used categorical cross-entropy  $L(\hat{y}, y)$  to train the parameters of the network.

$$L_{CE}(\hat{y}, y) = -\frac{1}{N} \sum_{j=1}^M \sum_{i=1}^N y_i^j \log(\hat{y}_i^j) \quad (6)$$

Where  $\hat{y}_i^j$  is predicted label and  $y_i^j$  is true label.  $M$  and  $N$  represents the number of classes, and the number of tweets respectively.

### 3.6. Bert+FastText-GRU Framework

Let  $(X_k, t1_k, t2_k)_{k=1}^N$  be a set of  $N$  tweets where  $t1_p \in T1$  (Hate Classes: HOF and NOT) and  $t2_p \in T2$  (Offensive Classes: PRFN, HATE, OFFN and NONE).  $t1_k, t2_k$  represents the hate and offensive labels corresponding to  $X_k^{th}$  tweet respectively. This Bert+FastText-GRU Framework aims to learn a function that maps an unknown instance  $X_k$  to its appropriate hate label  $t1_p$  and offensive label  $t2_p$ .

Let the input sentence  $X = \{x_1, x_2, \dots, x_n\}$  be a sequence of  $n$  input tokens, where  $n$  is the maximum length of a sentence. The input text  $X$  is fed into both the BERT and FastText models. BERT generates two types of outputs: a pooled output of shape  $[batch\ size, 768]$  that represents the whole input sequences, and a sequence output of shape  $[batch\ size, max\ seq\ length, 768]$  for each input token. Let  $W_B \in \mathbb{R}^{n \times D_B}$  be the embedding matrix obtained from the BERT model for input  $X$  where  $D_B = 768$  is the embedding dimension of each token. On the other hand, FastText generates an embedding matrix  $W_V \in \mathbb{R}^{n \times D_V}$ , where  $D_V = 300$ . Outputs from both BERT and FastText are passed through two separate Bi-GRUs (128 hidden units) followed by an attention layer to learn the contextual information and to assign more weightages on the relevant words. The outputs  $B$  and  $F$  returned by the attention layers placed on the top of BERT+GRU and FastText+GRU are concatenated to make a joint representation  $J$  of the input tweet  $X$ . The concatenated feature vector  $J$  is passed through a fully connected layers ( $FC_1(100\ neurons) + FC_2(100\ neurons)$ ) followed by an output layer.

### 3.7. Model Parameters and Settings

We use Tanh activation in GRU cells and ReLU activation in all fully connected layers (100 neurons each). We add a 25% dropout after the attention and fully-connected layers for all subtasks in both languages. With a batch size of 32, we train our models for 10 epochs. We utilize Adam optimizer and set the learning rate to 0.001 to backpropagate the loss across the network.

## 4. Experimental Results and Analysis

This section shows the results of different baseline models and our proposed model. All our experiments were conducted on a hybrid cluster of multiple GPUs comprised of RTX 2080 Ti. All the models are implemented using Scikit-Learn 0.22.2<sup>3</sup> and Keras 2.4.3<sup>4</sup> with TensorFlow2 2.4.1<sup>5</sup> as a backend.

### 4.1. Baselines Setup

Following baselines have been introduced for comparison with our proposed approach.

1. **BERT+GRU (Baseline-1)**: BERT generated word vectors are passed through BiGRU with attention layer. Output from attention layer is then passed to task specific fully connected layers[FC1(100) + FC2(100)] followed by output Softmax layer.
2. **BERT+LSTM (Baseline-2)**: Same as Baseline-1 with one modification: GRU is replaced by LSTM.
3. **FastText+GRU (Baseline-3)**: Same as Baseline-1, the only difference is the embedding approach. Here we have utilized FastText to generate word embedding of the input sentence.
4. **FastText+LSTM (Baseline-4)**: This is identical to Baseline-3, but here GRU is replaced by LSTM.
5. **BERT+FastText-LSTM (Baseline-5)**: This is identical to our proposed model with one modification: GRU is replaced by LSTM.

### 4.2. Results and Discussion

Table 2 presents the results in terms of accuracy, macro F1-score for all the baselines and the proposed model. From the result table, we can conclude that our model outperform all the baselines with a significant margin. Moreover, all the joint embedding based baselines performs better than any single embedding based baseline. Our model attained highest accuracy value of 76.32% 56.73%, 69.17% and 40.45% for four tasks, i.e, subtask-1A (EN), subtask-1B (EN), subtask-1A (HI) and subtask-1B (HI) respectively.

Out of four single embedding based baselines, FastText+GRU (Baseline-3) achieves higher f1 score of 74.92% and 66.41% for subtask-1A (EN) and subtask-1A (HI) respectively. While,

---

<sup>3</sup><https://scikit-learn.org/stable/>

<sup>4</sup><https://keras.io/>

<sup>5</sup><https://www.tensorflow.org/overview/>

**Table 2**

Experimental results of subtask-1A and subtask-1B for English and Hindi languages

| Model              | Task       | English  |              | Hindi    |              |
|--------------------|------------|----------|--------------|----------|--------------|
|                    |            | Accuracy | F1-Score     | Accuracy | F1-Score     |
| BERT+GRU           | Subtask-1A | 71.86    | 71.65        | 64.67    | 64.67        |
|                    | Subtask-1B | 53.18    | 53.01        | 36.87    | 36.52        |
| BERT+LSTM          | Subtask-1A | 71.16    | 71.47        | 64.27    | 64.58        |
|                    | Subtask-1B | 53.09    | 53.45        | 36.11    | 36.35        |
| FastText+GRU       | Subtask-1A | 74.59    | 74.92        | 66.28    | 66.41        |
|                    | Subtask-1B | 54.73    | 54.51        | 37.12    | 35.36        |
| FastText+LSTM      | Subtask-1A | 74.42    | 74.60        | 65.45    | 65.05        |
|                    | Subtask-1B | 54.68    | 54.70        | 36.26    | 35.89        |
| BERT+FastText-LSTM | Subtask-1A | 75.37    | 75.53        | 68.12    | 68.25        |
|                    | Subtask-1B | 55.43    | 55.26        | 40.35    | 37.12        |
| BERT+FastText-GRU  | Subtask-1A | 76.32    | <b>75.78</b> | 69.17    | <b>68.48</b> |
|                    | Subtask-1B | 56.73    | <b>56.52</b> | 40.45    | <b>37.82</b> |

FastText+LSTM (Baseline-4) attains the best f1 score of 54.70% and 35.89% for subtask-1B (EN) and subtask-1B (HI) respectively. FastText+GRU achieves 2.73% and 1.55%, respectively, improvements in accuracy values for subtask-1A (EN) and subtask-1B (EN) over the BERT+GRU. On the other hand, FastText+LSTM (Baseline-4) attains the improvements in accuracy values for subtask-1A (EN) and subtask-1B (EN) over the BERT+LSTM (Baseline-2) as 3.26% and 1.59%, respectively. We have also examined that for all the single embedding based baselines when embedded with FastText performs better than the one embedded with BERT.

Experimental results of this work imply that joint embedding representation enhances the hate and offensive post detection task’s performance compared to the one with single embedding.

## 5. Conclusion

With the expansion of digital sphere and advancement of technology, identifying hate, offensive and profane content from the post is strongly determined by many of the researchers. In this work, we have developed a deep learning model (BERT+FastText-GRU) based on BERT and FastText followed by GRU with attention. This attention mechanism allows us to give more weight to the words that contribute the most to the phrase’s meaning. In the HASOC-2021, our model is scored 32<sup>nd</sup> with a macro F1 score of 75.78% and 29<sup>th</sup> with a macro F1 score of 56.52% for English subtask-1A and subtask-1B respectively out of all entries. In Hindi language, our model ranked 32<sup>nd</sup> with a test accuracy of 69.17% for subtask-1A and 21<sup>st</sup> with a test accuracy of 40.45% for subtask-1B. From the results table, we can observe that the FastText-model outperformed the BERT-model in most experiments. Our team’s participation in the HASOC-2021 competition has been a valuable learning experience, and we look forward to learning from the other top-performing submissions.

## References

- [1] A. H. Razavi, D. Inkpen, S. Uritsky, S. Matwin, Offensive language detection using multi-level classification, in: Canadian Conference on Artificial Intelligence, Springer, 2010, pp. 16–27.
- [2] M. H. Ribeiro, P. H. Calais, Y. A. Santos, V. A. Almeida, W. Meira Jr, "like sheep among wolves": Characterizing hateful users on twitter, arXiv preprint arXiv:1801.00317 (2017).
- [3] R. Kumar, A. K. Ojha, S. Malmasi, M. Zampieri, Evaluating aggression identification in social media, in: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, 2020, pp. 1–5.
- [4] J. M. Struß, M. Siegel, J. Ruppenhofer, M. Wiegand, M. Klenner, et al., Overview of germeval task 2, 2019 shared task on the identification of offensive language (2019).
- [5] V. Basile, C. Bosco, E. Fersini, N. Debra, V. Patti, F. M. R. Pardo, P. Rosso, M. Sanguinetti, et al., Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter, in: 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2019, pp. 54–63.
- [6] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, in: Proceedings of the 11th forum for information retrieval evaluation, 2019, pp. 14–17.
- [7] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german, in: Forum for Information Retrieval Evaluation, 2020, pp. 29–32.
- [8] T. Ranasinghe, M. Zampieri, H. Hettiarachchi, Brums at HASOC 2019: Deep learning models for multilingual hate speech and offensive language identification., in: FIRE (Working Notes), 2019, pp. 199–207.
- [9] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, ACM, 2021.
- [10] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, A. K. Jaiswal, Overview of the HASOC subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021. URL: <http://ceur-ws.org/>.
- [11] W. Warner, J. Hirschberg, Detecting hate speech on the world wide web, in: Proceedings of the second workshop on language in social media, 2012, pp. 19–26.
- [12] A. Alakrot, L. Murray, N. S. Nikolov, Towards accurate detection of offensive language in online communication in arabic, Procedia computer science 142 (2018) 315–320.
- [13] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach, arXiv preprint arXiv:1809.08651 (2018).
- [14] J.-M. Xu, K.-S. Jun, X. Zhu, A. Bellmore, Learning from bullying traces in social media, in:



- Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies, 2012, pp. 656–666.
- [15] I. Kwok, Y. Wang, Locate the hate: Detecting tweets against blacks, in: Twenty-seventh AAAI conference on artificial intelligence, 2013.
  - [16] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, J. Leskovec, Anyone can become a troll: Causes of trolling behavior in online discussions, in: Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing, 2017, pp. 1217–1230.
  - [17] B. Gambäck, U. K. Sikdar, Using convolutional neural networks to classify hate-speech, in: Proceedings of the first workshop on abusive language online, 2017, pp. 85–90.
  - [18] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 11, 2017.
  - [19] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: Proceedings of the fifth international workshop on natural language processing for social media, 2017, pp. 1–10.
  - [20] Z. Zhang, D. Robinson, J. Tepper, Detecting hate speech on twitter using a convolution-gru based deep neural network, in: European semantic web conference, Springer, 2018, pp. 745–760.
  - [21] P. Liu, J. Guberman, L. Hemphill, A. Culotta, Forecasting the presence and intensity of hostility on instagram using linguistic and social features, in: Twelfth international aaii conference on web and social media, 2018.
  - [22] J. Pavlopoulos, P. Malakasiotis, I. Androutsopoulos, Deeper attention to abusive user content moderation, in: Proceedings of the 2017 conference on empirical methods in natural language processing, 2017, pp. 1125–1135.
  - [23] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
  - [24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
  - [25] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning word vectors for 157 languages, arXiv preprint arXiv:1802.06893 (2018).
  - [26] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, PMLR, 2014, pp. 1188–1196.
  - [27] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
  - [28] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259 (2014).
  - [29] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, 2016, pp. 1480–1489.