Multilingual Cyber Abuse Detection using Advanced Transformer Architecture

Aditya Malte
Dept. of Computer Engineering,
Pune Institute of Computer Technology,
Pune, India.
aditya.malte@gmail.com

Pratik Ratadiya
Dept. of Computer Engineering,
Pune Institute of Computer Technology,
Pune, India.
prratadiya@gmail.com

***Abstract*—The rise in the number of active online users has subsequently increased the number of cyber abuse incidents being reported as well. Such events pose a harm to the privacy and liberty of users in the digital space. Conventionally, manual moderation and reporting mechanisms have been used to ensure that no such text is present online. However, there have been some flaws in this method including dependency on humans, increased delays and reduced data privacy. Previous approaches to automate this process have involved using supervised machine learning and traditional recurrent sequence models which tend to perform poorly on non-English text. Given the rising diversity of users being a part of the cyberspace, a flexible solution able to accommodate multilingual text is the need of the hour. Furthermore, text in colloquial languages often hold pertinent context and emotion that is lost after translation. In this paper, we propose a generative deep-learning based approach which involves the use of bidirectional transformer-based BERT architecture for cyber abuse detection across English, Hindi and code-mixed Hindi English(Hinglish) text. The proposed architecture can achieve state-of-the-art results on the code-mixed Hindi dataset in the TRAC-1 standard aggression identification task while being able to achieve very good results on the English task leaderboard as well. The results achieved are without using any ensemble-based methods or multiple models and thus prove to be a better alternative to the existing approaches. Deep learning based models which perform well on multilingual text will be able to handle a broader range of inputs and thus can prove to be crucial in cracking down on such social evils.**

***Index Terms*—Cyber abuse, Generative networks, Deep learning, Transformers, Transfer Learning, natural language processing, text classification**

I. Introduction

The use of social media has actively been increasing in the 21st century. As of March 2019, there are approximately 4.38 billion internet users across the world. With the outreach of the internet constantly improving, the diversity of users has been on a rise as well. The cyber-world continues to be more global with multiple languages being used by users while interacting online. In linguistically diverse countries like India, South Africa, Indonesia, etc., the gap between users using native language and those using English has been significant. In 2016, there were 234 million Indian language users compared to only 175 million English users in the Indian online market. [1]. This gap is expected to only widen in the coming years.

With this expansion of digital space, the scourge of cyber abuse has also assumed terrifying proportions with a large number of incidents of bullying, threatening and hate on social media being reported every day worldwide. Lack of serious supervision has been a major factor in exacerbating this digital peril. Difference in opinions on various issues related to politics, sports, entertainment etc. often lead to users throwing abuses and invading the personal space of others. Such incidents of cyber abuse pose a serious threat to the privacy of users as well. Thus the detection of such abuses accurately in quick time is the need of the hour.

In the past, there have been numerous attempts to detect cyber abuse in an automated way using supervised machine learning and deep learning algorithms like SVM, Decision trees, LSTMs etc. [2], [3], [5]. However much of the work has been done on the English language only and such models tend to fail for other languages. Limited resources, lack of standard word embeddings and training data have hindered the progress of developing a multilingual cyber abuse detection system. Previous models have also faced difficulties in deriving a context-based understanding for text written in other languages. The general trend observed in approaches which have achieved state of the art results in this field has been the use of ensembling and augmentation methods which nonetheless face issues while handling non-english text. [8], [9]

In this paper, we propose a solution for the detection of cyber abuse in multiple languages-English and Hindi. We suggest the use of an advanced variant of the vanilla Transformer architecture called BERT that also incorporates bidirectional attention for this task [13], [15]. The proposed architecture is found to be superior in handling multilingual dependencies and provides optimum results on a standard dataset without the need of any ensembling methods. The proposed architecture can achieve state-of-the-art results for inputs in Hindi language on the TRAC-1 shared task on aggression identification as well [7]. It thus proves to be superior in terms of handling multilingual inputs denoting cyber abuse.

The rest of the paper is structured as follows: Section 2 talks about the previous approaches which have been used to tackle the problem; while the description of dataset and preprocessing techniques is provided in section 3. The proposed approach is elucidated in detail in section 4. Our analysis of the work conducted is presented in section 5. We present our results in section 6 and conclude the paper in section 7.

978-1-7281-1895-6/19/$31.00

II. Background and related work

Verbal aggression or abuse detection is an important area of research in Natural Language Processing. There have been numerous attempts to detect related behaviors like cyber-bullying in the past especially after the rise in the use of machine learning algorithms as well as the availability of computational power and data. Reynolds et. al [2] was one of the first to propose a machine learning-based system which made use of C4.5 decision trees and instance-based learner for cyberbullying detection. Dinakar et al [3] used various features like Tf-Idf, PoS tagging and label specific features for detecting cyber abuse. A Tf-Idf and N-gram based SVM and Logistic regression model was used by Chavan et al for detection of cyber-aggressive comments [4]. Badjatya et al [5] proposed the use of multiple deep learning-based approaches like CNNs, LSTMs, and Fasttext for the detection of hate speech in tweets. Founta et al proposed a unified deep learning-based model which combined two different networks for abuse detection [6].

The Shared Task on Aggression Identification conducted at COLING 2018 was a one of its kind challenge where participants were expected to detect cyber abuse by classifying text based on aggression level [7]. The task was organized across multiple languages-English and Hindi. We use the same dataset for our work and it is described in further sections. The leaders for the English FB task-Aroyehun and Gelbukh make use of LSTM based architecture along with other preprocessing techniques [8]. Raiyani et al [9] preferred a dense neural network model to achieve the leading result on Twitter(Other social media) English dataset. Samghabadi et al made use of a Logistic Regression based classifier to achieve the best results for the Hindi facebook dataset [10]. A CNN with Fasttext based approach was used to get the leading result for twitter Hindi dataset by Modha et al [11].

A common observation about all of these approaches was that the proposed models failed to perform well on both the datasets simultaneously. Also, there seems to be ample scope for improvement in the performance for the Hindi dataset task. Furthermore, it can be observed that most of the previous approaches use traditional methods and algorithms and fail to take advantage of improvements provided by modern architectures. We intend to improve the results while ameliorating the issues faced in previous approaches.

III. Dataset description and preprocessing

We use the dataset which was released as a part of the Shared Task on Aggression Identification organized at the Trolling, Aggression and Cyberbullying(TRAC-1) workshop at COLING 2018 [7]. The training dataset consists of 12000 randomly sampled Facebook comments in English and Hindi each. Every sample is annotated into one of the three categories-Overtly Aggressive(OAG), Covertly Aggressive(CAG) or Non-Aggressive(NAG) with decreasing levels of abuse in order. The testing dataset consists of 916 English comments and 970 Hindi comments. Along with these, an additional surprise test set consisting of 1,257 English tweets and 1,194 Hindi tweets was also released in the challenge, which we use as well to demonstrate the generalization ability of our trained models.

The distribution of training data across the three categories is as shown in table 1. Some samples in the training data consisted of code-mixed Hindi-English along with irrelevant text and thus needed to be preprocessed before training.

Category	English	Hindi
NAG	5052	2275
CAG	4240	4869
OAG	2708	4856

TABLE I: Distribution of training data

A. Preprocessing for English dataset

Hyperlinks are removed from the text as they do not indicate any aggression. Further, hashtags or other special characters used to tag individuals are stripped off. We remove repeated punctuation and whitespaces from the text. In a broader scope, numbers do not indicate any abuse and are thus removed from the sample. The text also includes emojis which we demojize into text indicating the meaning of the respective emoji. Preprocessing for a new sample text is shown in Table 2.

Example		Text
1	Before	I wish to #kill you ☹ @Yuvraj !!
	After	I wish to kill you :anger: Yuvraj !

TABLE II: Preprocessing example for a new English text

B. Preprocessing for Hindi dataset

The text for the Hindi dataset is also provided in Roman script so we carry out all preprocessing as done for the English dataset. However many of the Hindi words written in this script would not be present in our vocabulary thus making the task difficult. Hence, transliteration of all samples is done using the Google Transliteration API into Devnagari script. It makes the data more model friendly as shown from our results later. Transliteration is also a novelty of our approach as it improves the generalization ability of the model and its flexibility to accept English, Hindi as well as 'Hinglish'(written in both English and Hindi) text.

For a new test sample, it could be checked for whether some x% words in it are present in the English vocabulary of our model or not. If it does not satisfy this condition then the sample could be deemed as in Hindi.

IV. Proposed approach

The pre-processing steps ensure that the noisy data is effectively converted to clean text that can be fed to the model. While previous approaches have mainly focused on traditional methods such as character n-grams, Naive-Bayes, TF-IDF, vanilla LSTM networks and other such algorithms,

they do not improve in performance after a certain extent and are also computationally expensive. Besides, most of the approaches hardly made use of semi-supervised methods or transfer learning. As an improvement over these methods, We propose the use of the advanced self-attention mechanism and the *Deep Bidirectional Transformer* architecture.

A. Self-Attention and Transformer Architecture

Self attention is one variant of the attention mechanism where different positions of a sequence are related to calculate the representation of the same sequence. For predicting one word, an attention vector is computed based on its correlation with other words present in the sequence.

The Transformer Architecture proposed by Vaswani et al [13] has provided major performance improvements in terms of results and efficiency. The Transformer consists of an encoder-decoder architecture as shown in fig 1. The encoder is further comprised of a Multi-Head Attention layer, residual connections, normalization layer, and a generic feed-forward layer. The decoder is almost similar to the encoder with an added "masked" multi-head attention layer.

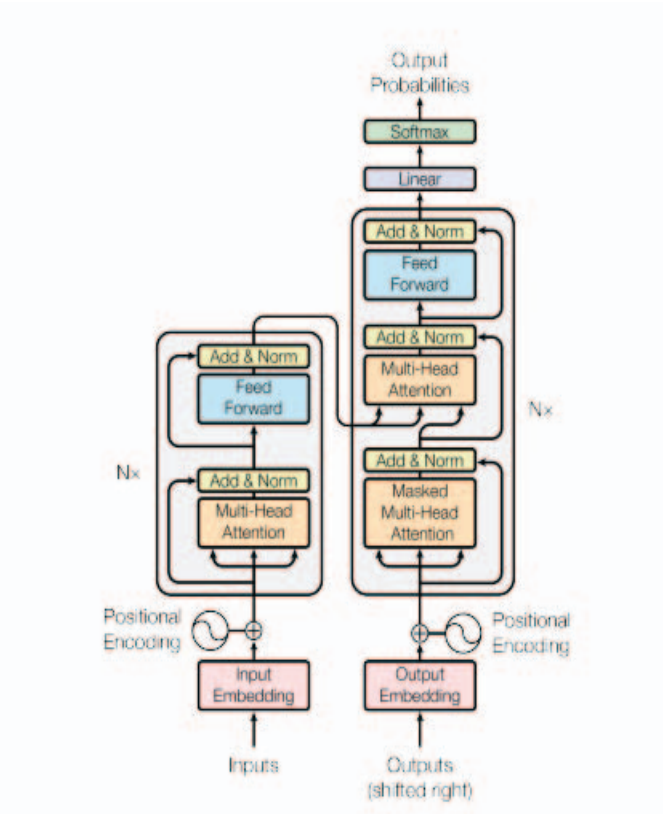

Fig. 1: Overview of Transformer architecture indicating encoder-decoder connections [13]

The encoder takes the input embedding and concatenates it with the positional encoding which provides the position and order related information. The positional encoding is derived as shown in equation 1 and 2.

$$PE_{(pos,2i)} = \sin(pos/10000^{(2i/d_{model})}) \quad (1)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{(2i/d_{model})}) \quad (2)$$

Where *pos* is the position of the word in the sequence and *i* is the dimension. This positional encoding is added to the input embedding. It is then followed by a residual connection *R*, which is calculated as follows:

$$R(x) = LayerNorm(x + MultiHeadedAttention(x)) \quad (3)$$

Where x is the value of the input embedding added with positional encoding.

Scaled dot product attention, a kind of self attention mechanism is also a crucial part of the transformer architecture [13]. Equation 4 represents the formula for calculating the scaled dot product based Attention weights.

$$Attention\,(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (4)$$

where

- V, is the value vector
- Query, $Q = EW_q$
- Key, $K = EW_k$
- Value, $V = EW_v$

Where W_q, W_k and W_v are weight matrices. A dot product of the query vector Q and key vector K is performed with a scaling factor $1/\sqrt{dk}$ to get the final value.

It can be seen that the complex recurrence mechanism is completely substituted by a sole attention based architecture in the transformer.

B. Bidirectional Transformer Architecture

Bidirectional Transformer Architecture, a variation of the vanilla Transformer architecture proves to be even better in providing context-based understanding of sequences [15]. We propose the use of Bidirectional Encoder Representations from Transformer(BERT) architecture. The generative model is trained in a semi-supervised approach composing of two phases:

- Pre-Training Phase:
  During the pre-training phase the model is trained both for the Masked LM task as well as Next Sentence Prediction task.

  Masked Language Model This phase of the Bidirectional Transformer model proceeds as shown in Algorithm 1.

  The use of bidirectional context allows the model to provide better performance as a language model compared to previous approaches that used a single unidirectional model or the combination of two unidirectional(backward and forwards) language models.

  The model is pre-trained as a Masked Language Model(Masked LM) on the Wikipedia dataset of the

Algorithm 1 Masked Language Modeling

Input: WordPeice Tokenized Sentences S
Output: Trained model

for all $s \in S$ **do**
  $S_{\text{masked}} \leftarrow$maskingFunction($s$)
  $output \leftarrow$forward_pass(S_{masked})
  backpropogate(CROSS_ENTROPY_LOSS())
end for
function maskingFunction($tokens$)
for all $token \in tokens$ **do**
  $c \leftarrow$random() {Probability = P}
  if choice = 1 **then**
    $token \leftarrow$[MASK] {P=0.8}
  else if choice = 2 **then**
    $token \leftarrow$random_word {P=0.1}
  else if choice = 3 **then**
    $token \leftarrow$token {P=0.1}
  end if
  $masked\_text$.append($token$)
end for
return $masked\_text$

respective language along with the BookCorpus dataset in case of English [15]. The pre-training phase allows the model to learn inherent dependencies among words in the given sequence. Thus the syntactic and semantic relationships encoded within the sequence are learnt.

Next Sentence Prediction The model is also trained on the Next Sentence Prediction task. Essentially, given two sentences(S_a and S_b), the model has to predict whether the first sentence S_a is succeeded by the sentence S_b. The same is implemented using the steps described in Algorithm 2.

Algorithm 2 Next Sentence Prediction

Input: T_n: tuples of form (S_a, S_b, $label$) ;
Output: Trained model

for all $tuple \in T_n$ **do**
  $S_a \leftarrow$random_mask($tuple$[1], percent=15)
  $S_b \leftarrow$random_mask($tuple$[2], percent=15)
  $input \leftarrow$[CLS] + S_a + [SEP] + S_b
  $label \leftarrow tuple$[3]
  $output \leftarrow$forward_pass($input$)
  backpropogate(CROSS_ENTROPY_LOSS())
end for

Training the model for this task aids the model during the fine-tuning phase of sentence pair tasks such as semantic similarity.

- Fine-Tuning Phase: This phase is used to train the model for a specific task. In essence, this is akin to using *Transfer Learning* for neural networks in Computer Vision. Transfer learning implies using a model pre-trained for another task and trying to fit it for the current task. Fine-tuning is performed by adding linear layers to the output corresponding to $[CLS]$ token(which is inserted at the beginning of the sequence) for single sentence classification tasks.

We add a single linear layer followed by a softmax operation on the same. Our output layer consists of three nodes corresponding to the three aggression labels. We use the publicly available multilingual-cased pre-trained model weights for the Hindi dataset and $BERT_{BASE}$ pre-trained model weights for the English dataset for the fine-tuning task[1].

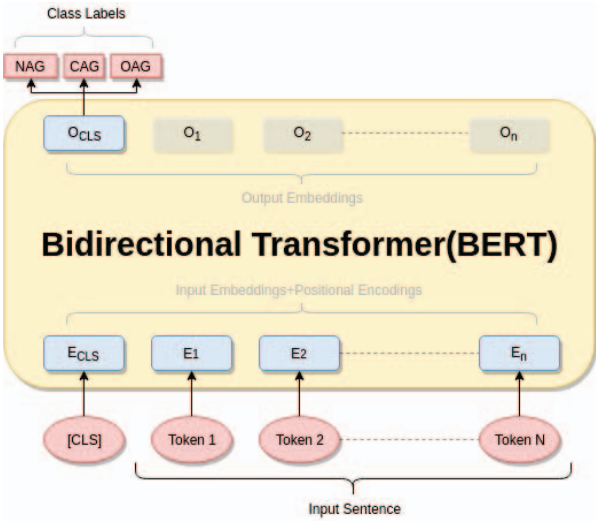

Fig. 2: Proposed Architecture using Bidirectional Transformer

C. Slanted Triangular Learning Rate

We use Slanted Triangular Learning Rate [14] to improve the convergence of the model. The learning rate is first linearly increased for the first 'n' iterations and then linearly decreased till the end. The formula for SLTR is given as follows:

$$cut = [T \cdot cut_{frac}] \tag{5}$$

$$p = \begin{cases} t/cut, & t < cut \\ 1 - \frac{t - cut}{cut \cdot (1/cut_{frac} - 1)}, & otherwise \end{cases} \tag{6}$$

$$\eta_t = \eta_{max} \cdot \frac{1 + p \cdot (ratio - 1)}{ratio} \tag{7}$$

Here, cut_{frac} is the fraction of iterations where the learning rate is linearly increased. The same is kept at 0.1, which provides good performance.

The intuition behind using SLTR is that it allow the model to quickly converge to a hyperspace that is near to the optimal

[1]https://github.com/google-research/bert

solution. The linear decay then allows for finer updates in the weights such that error is minimized.

D. Dropout

Dropout has been a novel solution to mediate the effect of overfitting in neural networks [12]. It works by randomly(with a probability) setting the activations of nodes in the neural network to zero. By doing so we reduce the ability of individual neurons to develop co-dependencies thereby increasing the predictive power of individual neurons. Besides, by creating unique sub-networks during each training iteration, we implicitly allow the network to act as an ensemble using just a single network which increases its performance. During our experimentation, we found a value of dropout probability equal to 0.1 to be ideal. Further increase of the dropout led to the model being unable to converge on an optimum solution thus leading it to underfit on the training data.

The use of *Dropout* and *Slanted Triangular Learning Rates* ensures that the model converges to optimum and strongly generalized solutions-an advantage over the existing approaches.

V. Analysis

The best performance on the validation set was achieved when batch size was set to 32 for the English dataset and 64 for the Hindi dataset. Increasing this hyperparameter further led to lower performance. Further, the validation loss was consistently less than or equal to the training loss, indicating no overfitting by the model. We found that the model generally converged and provided high validation performance when the number of training steps were set to 700 for the Hindi dataset and 800 for the English dataset. Lower values generally led to underfitting of the model while higher values led to lower performance on the validation set. Transliteration as a preprocessing step of the code-mixed Hindi dataset consistently led to better convergence of the model. This can be directly attributed to the fact that the BERT-Multilingual model was not pre-trained on code-mixed data.

VI. Results

The performance of the model is evaluated using the weighted macro F1 score metric. Weighted F1 score calculates the F1 score for each label which is then averaged based on the weights provided by support(no. of positive instances of each label). F1 score is defined as:

$$F1\ score = \frac{2 * p * r}{p + r}$$

where p stands for precision which is the ratio of the number of correct positive results to the number of all positive results obtained by the classifier. r stands for recall which is number of correct positive results divided by total number of samples which should had been classified as positive.

We compare our results with the 30 participant teams for the challenge. The results obtained were as follows:

We have thus been able to achieve state-of-the-art results for the Hindi facebook test dataset while achieving a satisfactory third position on the English test dataset as well.

Rank	System	F1(Weighted)
1	BERT(multilingual cased)	**0.6596**
2	N-grams(Samghabadi et al)	0.6451
3	GRU-Ensemble(Krestel et al)	0.6311
–	Random baseline	0.3571

TABLE III: Results on Hindi FB test set

Rank	System	F1(Weighted)
1	LSTM+augmented data(Aroyehun et al)	0.6425
2	Multimodel ensemble(Modha et al)	0.6315
3	BERT(base uncased)	**0.6244**
–	Random baseline	0.3535

TABLE IV: Results on English FB test set

The confusion matrix is visualized in figures 3 and 4. To further demonstrate the flexibility of our model, we test its performance on the surprise test dataset of Twitter comments without tuning the model at all. The results obtained are as tabulated in table 5.

Approach	Weighted F1: Hindi	Weighted F1: English
BERT	0.4521	0.5520
Random baseline	0.3206	0.3477

TABLE V: Results on surprise test set: Twitter

System	F1(Weighted)
BERT(multilingual cased) raw text	0.641
BERT(multilingual cased) transliterated text	**0.6596**

TABLE VI: Effect of transliteration as preprocessing on Hindi FB test set

Our approach is thus able to produce satisfactory results which give us the 5th and 6th positions out of 31 teams on the Hindi and English dataset respectively. It should be noted here that this surprise test set consisted of tweets which contain a lot of slang and grammatically incorrect language. Despite this, our model which is trained on relatively cleaner data can produce good results. We have thus been able to use an architecture which not only generalizes across different nature of data(clean text or slang) but also different languages(Hindi and English) as shown from our results. It is also remarkable that adding transliteration as a preprocessing significantly improves the performance on the Hindi test set as can be seen in table 6.

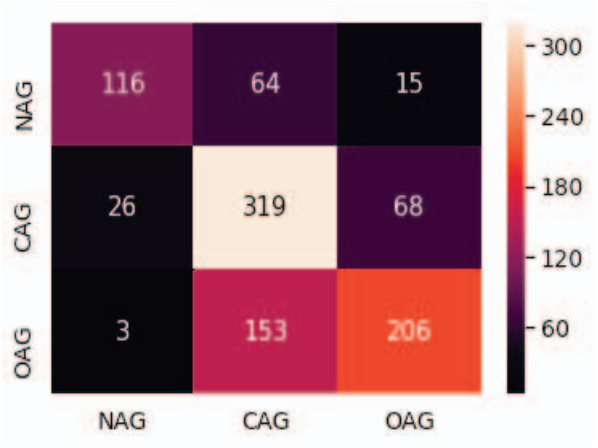

Fig. 3: Confusion matrix for the Hindi FB test dataset

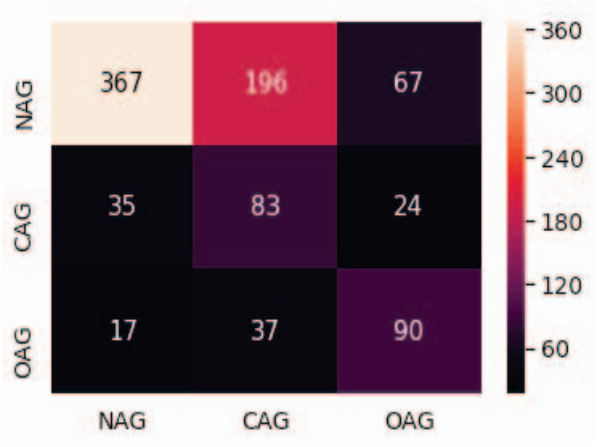

Fig. 4: Confusion matrix for the English FB test dataset

VII. Conclusion

Thus by making use of bidirectional transformer architecture, we have been able to achieve benchmark results across multiple languages for cyber abuse detection. The ability of the model to produce quality results on both clean as well as noisy data is motivating. Unlike previous approaches that have relied on recurrence and ensembling based methods to solve this problem, we made use of an architecture which is both computationally effective as well as more flexible. The model can be further trained upon external data to improve the results. Analysis of the BERT-Large model could be of interest in the future. A more effective way to handle slang text in non-English languages can also be considered to improve the representations of such words. With the world coming more closer in this new digital age, machine learning models must be also able to handle diverse data effectively. Maintaining privacy and liberty of all users across cyberspace is essential and the development of such generative deep learning models is an early step in this direction.

References

[1] Tech Startups, Take Note: More Indians Access The Internet In Their Native Language Than In English, "https://www.forbes.com/sites/baxiabhishek/2018/03/29/more-indians-access-the-internet-in-their-native-language-than-in-english/27aaee7b4a03"

[2] Kelly Reynolds, April Kontostathis, Lynne Edwards, "Using Machine Learning to Detect Cyberbullying", 10th International Conference on Machine Learning and Applications and Workshops, Dec 2011.

[3] Dinakar, K., Reichart, R. and Lieberman H, "Modeling the detection of textual cyberbullying", In Proceedings of the International Conference on Weblog and Social Media 2011.

[4] Vikas Chavan, Shylaja SS, "Machine learning approach for detection of cyber-aggressive comments by peers on social media network", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015.

[5] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, Vasudeva Varma, "Deep Learning for Hate Speech Detection in Tweets", Proceedings of ACM WWW'17 Companion, Perth, Western Australia, Apr 2017

[6] Antigoni-Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, Ilias Leontiadis, "A Unified Deep Learning Architecture for Abuse Detection", arXiv:1802.00385 [cs.CL]

[7] Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri, "Benchmarking Aggression Identification in Social Media", In Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC), Santa Fe, USA, 2018.

[8] Segun Taofeek, Aroyehun and Alexander Gelbukh. "Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling", In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC 1), Santa Fe, USA, 2018.

[9] Kashyap Raiyani, Teresa Goncalves, Paulo Quaresma, and Vitor Beires Nogueira. "Fully connected neural network with advance preprocessor to identify aggression over facebook and twitter", In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC 1), Santa Fe, USA, 2018.

[10] Niloofar Safi Samghabadi, Deepthi Mave, Sudipta Kar, and Thamar Solorio, "Ritual-uh at Trac 2018 shared task: Aggression identification", In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC 1), Santa Fe, USA, 2018.

[11] Sandip Modha, Prasenjit Majumder, and Thomas Mandl, "Filtering aggression from multilingual social media feed", In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC 1), Santa Fe, USA, 2018.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Journal of Machine Learning Research, 1929 1958, 2014.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention is All you Need", Advances in Neural Information Processing Systems(NIPS) 30, 2017.

[14] Jeremy Howard, Sebastian Ruder, "Universal Language Model Fine-tuning for Text Classification", arXiv:1801.06146 [cs.CL]

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv:1810.04805 [cs.CL]