BERT-Capsule Model for Cyberbullying Detection in Code-Mixed Indian Languages

Krishanu Maity and Sriparna Saha

Department of Computer Science and Engineering
Indian Institute of Technology, Patna, India
{krishanu_2021cs19,sriparna}@iitp.ac.in

Abstract. In this work, we have created a benchmark corpus for cyberbullying detection against children and women in Hindi-English code-mixed language. Both these languages are the medium of communication for a large majority of India, and mixing of languages is widespread in day-to-day communication. We have developed a model based on BERT, CNN along with GRU and capsule networks. Different conventional machine learning models (SVM, LR, NB, RF) and deep neural network based models (CNN, LSTM) are also evaluated on the developed dataset as baselines. Our model(BERT+CNN+GRU+Capsule) outperforms the baselines with overall accuracy, precision, recall and F1-measure values of 79.28%, 78.67%, 81.99% and 80.30%, respectively.

Keywords: Cyberbullying · Code-Mixed(Hindi+English) · MuRIL BERT · Capsule Networks

1 Introduction

Cyberbullying is defined through malicious tweets, texts or other social media posts via various digital technologies as the serious, intentional and repeated actions of a person's cruelty towards others [13]. Cyberbullying outcomes can differ from transient fear to suicide. So, automatically detecting cyberbullying at its initial stage is a crucial step to prevent its outcomes. State of the art research primarily concentrates on cyberbullying detection for the English language. Indigenous languages have not been given much attention due to the lack of proper datasets. Code-mixing(CM) is the process of fluid alternation between two or more languages in a conversation [9]. It is a natural process of embedding linguistic units such as sentences, words or morphs of one language into the speech of another [8].

Data released by the National Crime Records Bureau showed that the cases of cyberbullying against women or children have increased by 36% from 2017 to 2018 in India[1]. In India the majority of text conversations in social media platform are in the form of Hindi, English and Hinglish. Hinglish is nothing but

[1] https://ncrb.gov.in/en/crime-india-2018-0

the representation of Hindi words in Roman script. We have created a Hindi-English code-mixed annotated (Bully/Non-bully) dataset for cyberbullying detection specially related to children and women.

We have developed a model based on BERT [5], CNN, GRU and Capsule network. During our study, we have used MuRIL BERT[2](Multilingual Representations for Indian Languages), pre-trained on 17 Indian languages and their transliterated counterparts. In recent years, capsule network [11] has gained much attention not only in the computer vision domain but also in NLP domain due to its ability to learn hierarchical relationships between consecutive layers by using an iterative dynamic routing strategy. The main contributions of this work are as follows:

1. We create a new Hindi-English code-mixed annotated (Bully/Non-bully) dataset for cyberbullying detection specially related to children and women.
2. We have developed a model based on BERT, CNN, GRU and capsule network for detecting cyberbully from code-mixed tweets.
3. We have considered traditional machine learning models (Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF)) and deep neural network based models (CNN, LSTM) as baselines and our model outperforms all the baselines with a significant margin.

2 Related Works

With the advancement of NLP, a large number of research has been conducted on cyberbullying detection on English language as compared to other languages. Dinakar et al. [6] introduced a machine learning based cyberbullying detection model trained on YouTube comments corpus (4500 instances) based on sexuality, racism and intelligence contents. Reynolds et al. [10] used the data obtained from the Formspring.me website, a formatted question-and-answer website for cyberbullying detection. In 2017, Badjatiya et al. [1] experimented with a dataset of 16K annotated tweets with three labels *sexist, racist, and Nan*. In 2020, Balakrishnan et al. [2] proposed a model for cyberbullying detection based on Twitter users' psychological features and machine learning techniques. Bohra et al. [3] created a Hindi-English code-mixed dataset consisting of 4575 tweets annotated with hate speech and normal speech. Gupta et al. [7] proposed a deep gated recurrent unit (GRU) architecture for entity extraction in code-mixed Indian languages.

From literature review, we have observed that there is no existing corpus for detecting cyberbullying against children and women in Hindi-English code-mixed language.

[2] https://tfhub.dev/google/MuRIL/1

3 Code-Mixed Cyberbully-Annotated Corpora Development

3.1 Data Collection

With the help of Twitter Search API[3], we have collected tweets from Twitter. We have scraped approximately 90K raw tweets between July 2020 to November 2020 based on specific hashtags and keywords related to women's attacks like MeToo, r*ndi, JusticeForSushantSinghRajput, nepotism, IndiaAgainstAbuse, AliaBhatt, bitch etc.

3.2 Data Annotation

After preprocessing of raw tweets, we perform manual annotation of the dataset. Two human annotators having linguistic background and proficiency in both Hindi and English, carried out the data annotation task. For annotation, we follow the guidelines used in Hee et al. [14]. Some examples of the annotated tweets are shown in Table 1. To check the quality of annotation carried out by two annotators, we have calculated the inter-annotator agreement (IAA) using Cohen's Kappa coefficient. Kappa score is 0.85, which proves that data is of acceptable quality. After data preprocessing, we have kept 5062 number of tweets in our corpus. Out of 5062 tweets in our corpus, 2456 were labeled as nonbully and the remaining 2606 tweets were labeled as bully.

Table 1. Samples from annotated dataset

Tweets	Class
T1: Kuch bengali se baat kiya kaar phir Main bhe guwahati gaya tha ak baar beautiful place ha **Translation**:I went to Guwahati after discussing with few Bengali people, it's a beautiful place.	Non-Bully
T2: Aurat mard brbr hai yh modern concept nikl do khud k dmg sy **Translation**: Woman men are all equal, let this modern concept leave from mind itself.	Bully
T3: han g bhai address likh lo, jider tumari maa aur behn soyee huee hai the **Translation**:Yes brother please write the address, wherever your mother and sister were sleeping.	Bully
T4: tum itne simple ho isliye sob tumko chuthiya banate he. **Translation**:You are so simple, that's why everyone makes you fool.	Non-Bully

4 Methodology for Cyberbullying Detection

Our model (BERT+CNN+GRU+Capsule), drawn in Figure 1, is a variant of the BERT-Caps [12]. We have also examined some baseline models based on the traditional machine learning algorithms (SVM, LR, NB, RF) and compared them with our model.

4.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) [5] is a Transformer-based [15] language model developed by the Google AI research

[3] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets

team. Let the input sentence $X = \{x_1, x_2, ...x_n\}$ be the sequence of n input tokens where n represents the maximum sentence length. We feed the input sentence X to BERT model. It returns two types of outputs, i.e., the pooled output of shape $[batch\ size, 768]$, which represents the entire input sequences and a sequence output of shape $[batch\ size, max\ seq\ length, 768]$ with representations from each input token. Let $W_B \in \mathbb{R}^{n \times D}$ be the embedding matrix obtained from the BERT model for input X where $D = 768$ is the embedding dimension of each token.

4.2 N-gram Convolutional Layer

The output from the BERT model $W_B^{n \times D}$ is then passed through convolution layers to extract the N-gram feature map. Let $F_a \in \mathbb{R}^{K_1 \times D}$ be the learnable filter where K_1 is the N-gram size. Filter F_a performs an element-wise dot product over each possible word-window, $w_{i:i+k_1-1}$ to get feature map, $\mathbf{c^a} \in \mathbb{R}^{n-K_1+1}$. A feature map c_i^a is generated after convolution by $c_i^a = f(w_{i:i+k_1-1} * F_a + b)$, where f is a non linear activation function with bias b. After applying t number of different filters of the same N-gram size, one can generate t feature maps, which can be rearranged as $\mathbf{C} = [\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}, ...\mathbf{c_t}] \in \mathbb{R}^{n-K_1+1 \times t}$.

4.3 Bi-directional GRU Layer

To learn semantic dependency-based features, we passed t-channel feature vector $\mathbf{C}$ through a bi-directional Gated Recurrent Units (GRUs) [4]. Bi-directional GRU sequentially encodes these feature maps into hidden states to capture long-term dependencies in the tweet as, $\overrightarrow{h}_t = \overrightarrow{GRU}(c_t,\ h_{t-1})$, $\overleftarrow{h}_t = \overleftarrow{GRU}(c_t,\ h_{t+1})$, where each convoluted feature map c_t is mapped to a forward hidden state $\overrightarrow{h}_t$ and backward hidden state $\overleftarrow{h}_t$ by invoking $\overrightarrow{GRU}$ and $\overleftarrow{GRU}$, respectively. Finally $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are concatenated to get a single hidden state representation h_t, $\left[h_t = \overrightarrow{h}_t, \overleftarrow{h}_t\right]$. The final hidden state matrix is obtained as,

$\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \mathbf{h_3}, ...\mathbf{h_t}] \in \mathbb{R}^{t \times 2d}$, where d is the dimension of hidden state.

4.4 Primary Capsule Layer

Primary capsules hold a group of neurons to represent each element in the feature maps as opposed to a scalar, in order to preserve the instantiated parameters such as the local order of words and semantic representations of words. Let $p_i \in \mathbb{R}^d$ denote the instantiated parameters of a capsule, where d is the dimension of the capsule. By sliding each kernel K_i, over the GRU generated hidden state matrix H, we have a sequence of capsules, p_i. A channel P_i in the primary capsule layer is the list of capsules p_i, described as $P_i = g(H * K_i + b)$ where g is a squashing function with bias b. For all R such channels, the generated capsule feature map can be compiled as $P = [P_1, P_2, P_3, ...P_R]$.

4.5 Dynamic Routing Between Capsules

The fundamental idea of dynamic routing is to build a non-linear map in an iterative way, assuring that the lower label capsule has a strong connection to

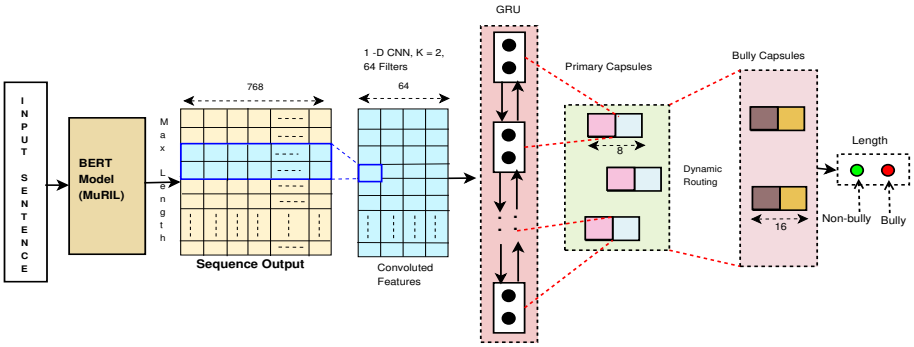

Fig. 1. BERT+CNN+GRU+Capsule architecture.

an appropriate capsule in the next layer. This algorithm increases or decreases the connection strength for each potential higher label capsule and by this way, it not only detects whether a feature is present in any position of the text or not, but also keeps the spatial information about the feature. Let u_i be a capsule in layer l. A capsule v_j in layer $l+1$ is calculated as:

$$v_j = g(\sum_i S_{ij}\hat{u}_{j|i}) \;\; and \; \hat{u}_{j|i} = W_{ij}u_i \tag{1}$$

Where a predicted vector $\hat{u}_{j|i} \in \mathbb{R}^d$ is calculated from the capsule u_i, W_{ij} is a weight matrix, g is a non-linear squashing function which restricted the length of the capsule in the range of $[0, 1]$ and S_{ij} is a coupling coefficient iteratively updated by the dynamic routing algorithm [11].

4.6 Bully Capsule Layer with Loss

The bully capsule layer is the final capsule layer consisting of two class capsules, one for the bully class and another for the non-bully class. Each capsule has 16-dimensional (d = 16) instantiated parameters, and its length (norm) describes the probability of the input sample belonging to this class label. In order to magnify the difference between the lengths of two class capsules and for better generalization, we have considered separate margin loss [16] as,

$$L_e = T_e \; max(0, m^+ - \parallel v_e \parallel)^2 \; + \lambda \; (1 - T_e) \; max(0, \parallel v_e \parallel - m^-)^2 \tag{2}$$

where v_e represents the capsule for category e. In our problem, e is either bully or non-bully. Top and bottom margins are represented by $m^+ = 0.9$ and $m^- = 0.1$, respectively. λ is used for down-weighting of the classes which are not present.

5 Experimental Results and Analysis

Out of 5562 instances in our proposed dataset, we have randomly selected 75% of data for training, 15% for validation, and the remaining 15% for testing. We have used Scikit-Learn 0.22.2 to implement machine learning algorithms. Keras 2.3.1 with TensorFlow as a backend is used to implement deep learning-based models. We have conducted all the experiments ten times and reported the average results.

5.1 Comparison with the Baselines

We have introduced the following baselines for comparison with our model.

1. **BERT Embedding+SVM (Baseline-1)**: The pooled output of MuRIL BERT with dimension 768 is fed to SVM classifier for predictions. Hyperparameters of SVM: regularization parameter C=0.8; kernel=linear; class weight=balanced; tolerance=1e-3.
2. **BERT Embedding+LR (Baseline-2)**: The pooled output of MuRIL BERT with dimension 768 is given to LR model as an input. Hyperparameters of LR: penalty=l1; class weight=balanced; solver=liblinear.
3. **BERT Embedding+NB (Baseline-3)**: The pooled output of MuRIL BERT with dimension 768 is fed to NB classifier for predictions.
4. **BERT Embedding+RF(Baseline-4)**: The pooled output of MuRIL BERT with dimension 768 is given to LR model as an input. Hyperparameters of LR: criterion = "gini", max features ="auto".
5. **BERT+LSTM (Baseline-5)**: A sequence of words with 768 embedding vectors generated from BERT model is sent to the LSTM layer with 64 hidden states. Outputs of the LSTM layer are then passed through a soft-max layer for prediction. Hyperparameters used are: batch size=32; optimizer=Adam; loss=categorical cross-entropy; dropout probability=0.5
6. **BERT+CNN (Baseline-6)**: The sequence output from the BERT model is passed through 1-D convolution layers. We have considered 64 filters with filter sizes 1 and 2. After performing the average pooling operation, we have concatenated the feature maps and passed them through fully connected layers with 60 neurons followed by a soft-max layer.
7. **BERT+CNN+Capsule (Baseline-7)**: In this baseline, BERT's output is passed through a 1D CNN layer with filter sizes 1, 2 and the number of filters for each size = 64.
8. **BERT+LSTM+Capsule (Baseline-8)**: Sequence output of BERT model is sent to the Bidirectional LSTM layer with 64 hidden states. Hidden state matrix generated from LSTM is then passed through the capsule network for prediction.
9. **BERT+GRU+Capsule (Baseline-9)**: This is identical to Baseline-8, the only exception is hare LSTM is replaced by GRU.

Table 2 presents the results attained by all the baselines and the proposed model in terms of accuracy, precision, recall, and F1-score. Methods from both

Table 2. Evaluation results of cyberbully detection attained by the baselines and the proposed approach

Model	Accuracy	Precision	Recall	F1 Score
BERT Embedding + SVM (Baseline-1)	73.93	71.79	78.61	75.04
BERT Embedding + LR (Baseline-2)	72.26	70.74	75.6	73.11
BERT Embedding + NB (Baseline-3)	69.29	68.02	72.47	70.18
BERT Embedding + RF (Baseline-4)	71.86	71.23	73.06	72.14
BERT + LSTM (Baseline-5)	76.18	74.20	79.89	76.94
BERT + CNN (Baseline-6)	77.28	77.87	77.12	77.45
BERT + CNN + Capsule (Baseline-7)	77.70	75.75	77.43	76.58
BERT + LSTM + Capsule (Baseline-8)	78.18	78.24	80.75	78.48
BERT + GRU + Capsule (Baseline-9)	78.33	76.19	78.22	77.19
BERT+CNN+GRU+Capsule	**79.28**	**78.67**	**81.99**	**80.30**

machine learning (baseline - 1, 2 , 3, 4) and deep learning (baseline - 5, 6 , 7, 8, 9) have been taken into account in our baselines. It can be concluded from the table that our proposed model produced better results than all other baselines by a significant margin. Compared to the best baseline, i.e., baseline-9, our model showed almost 1% improvement in accuracy. We can conclude that BERT Embedding+SVM (Baseline-1) achieves higher accuracy (73.93%) than other machine learning-based baselines. We have also examined that baseline-7 and baseline-8 outperform baseline-6 and baseline-5 with accuracy values of 0.42% and 2%, respectively. This improvement in accuracy suggests that the inclusion of a capsule network greatly enhances the performance. If we look closely at baseline-7 and 8, we can see that the only discrepancy between these two baselines is separate recurrent network usages, i.e., LSTM vs. GRU. From the result table, we can analyze that baseline-8 marginally outperforms baseline-7. All the reported results are statistically significant as we have performed statistical t-test at 5% significance level.

6 Conclusion and Future Work

In this paper, we have developed a benchmark corpus for cyberbullying identification against children and women in code-mixed Indian languages. From Twitter, we have crawled Hindi-English code-mixed tweets and, after pre-processing, we have manually annotated 5062 number of tweets. Hindi and English are selected because these languages are the most preferred mode of communication in India. We have developed a model based on four deep learning models: BERT, CNN, GRU, and Capsule networks. We have examined that the inclusion of capsule networks with other deep learning models (CNN, LSTM or GRU) significantly enhances the classifier's performance. Experimental results showed that our model BERT+CNN+GRU+Capsule produced better results than all other baselines by a significant margin. In future, we would like to develop a multitasking framework for cyberbullying detection, where sentiment and emotion detections can act as auxiliary tasks.

References

1. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 759–760 (2017)
2. Balakrishnan, V., Khan, S., Arabnia, H.R.: Improving cyberbullying detection using twitter users' psychological features and machine learning. Computers & Security **90**, 101710 (2020)
3. Bohra, A., Vijay, D., Singh, V., Akhtar, S.S., Shrivastava, M.: A dataset of hindi-english code-mixed social media text for hate speech detection. In: Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media. pp. 36–41 (2018)
4. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Dinakar, K., Reichart, R., Lieberman, H.: Modeling the detection of textual cyberbullying. In: Proceedings of the International Conference on Weblog and Social Media 2011. Citeseer (2011)
7. Gupta, D., Ekbal, A., Bhattacharyya, P.: A deep neural network based approach for entity extraction in code-mixed indian social media text. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
8. Muysken, P., Muysken, P.C., et al.: Bilingual speech: A typology of code-mixing. Cambridge University Press (2000)
9. Myers-Scotton, C.: Duelling languages: Grammatical structure in codeswitching. Oxford University Press (1997)
10. Reynolds, K., Kontostathis, A., Edwards, L.: Using machine learning to detect cyberbullying. In: 2011 10th International Conference on Machine learning and applications and workshops. vol. 2, pp. 241–244. IEEE (2011)
11. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. arXiv preprint arXiv:1710.09829 (2017)
12. Saha, T., Jayashree, S.R., Saha, S., Bhattacharyya, P.: Bert-caps: A transformer-based capsule network for tweet act classification. IEEE Transactions on Computational Social Systems **7**(5), 1168–1179 (2020)
13. Smith, P.K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., Tippett, N.: Cyberbullying: Its nature and impact in secondary school pupils. Journal of child psychology and psychiatry **49**(4), 376–385 (2008)
14. Van Hee, C., Verhoeven, B., Lefever, E., De Pauw, G., Daelemans, W., Hoste, V.: Guidelines for the fine-grained analysis of cyberbullying. Tech. rep., version 1.0. Technical Report LT3 15-01, LT3, Language and Translation … (2015)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
16. Xiao, L., Zhang, H., Chen, W., Wang, Y., Jin, Y.: Mcapsnet: Capsule network for text with multi-task learning. In: Proceedings of the 2018 conference on empirical methods in natural language processing. pp. 4565–4574 (2018)