Multimedia Tools and Applications (2023) 82:8773–8789
https://doi.org/10.1007/s11042-021-11601-9

1209: RECENT ADVANCES ON SOCIAL MEDIA ANALYTICS AND MULTIMEDIA SYSTEMS: ISSUES AND CHALLENGES

COVID-19 and cyberbullying: deep ensemble model to identify cyberbullying from code-switched languages during the pandemic

Sayanta Paul[1] · Sriparna Saha[1] · Jyoti Prakash Singh[2]

Received: 6 November 2020 / Revised: 27 May 2021 / Accepted: 22 September 2021 /
Published online: 8 January 2022

Abstract

It has been declared by the World Health Organization (WHO) the novel coronavirus a global pandemic due to an exponential spread in COVID-19 in the past months reaching over 100 million cases and resulting in approximately 3 million deaths worldwide. Amid this pandemic, identification of cyberbullying has become a more evolving area of research over posts or comments in social media platforms. In multilingual societies like India, code-switched texts comprise the majority of the Internet. Identifying the online bullying of the code-switched user is bit challenging than monolingual cases. As a first step towards enabling the development of approaches for cyberbullying detection, we developed a new code-switched dataset, collected from Twitter utterances annotated with binary labels. To demonstrate the utility of the proposed dataset, we build different machine learning (Support Vector Machine & Logistic Regression) and deep learning (Multilayer Perceptron, Convolution Neural Network, BiLSTM, BERT) algorithms to detect cyberbullying of English-Hindi (En-Hi) code-switched text. Our proposed model integrates different handcrafted features and is enriched by sequential and semantic patterns generated by different state-of-the-art deep neural network models. Initial experimental results of the proposed deep ensemble model on our code-switched data reveal that our approach yields state-of-the-art results, i.e., 0.93 in terms of macro-averaged F1 score. The dataset and codes of the present study will be made publicly available on the paper's companion repository [https://github.com/95sayanta/COVID-19-and-Cyberbullying].

Keywords Natural language processing · Cyberbullying · Code-switched language · Deep ensemble

List of Abbreviations

En-Hi	English-Hindi
ML	Machine Learning

✉ Sriparna Saha
sriparna@iitp.ac.in

Extended author information available on the last page of the article

DL	Deep Learning
SVM	Support Vector Machine
LR	Logistic Regression
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
BiLSTM	Bi-directional Long Short Term Memory
BERT	Bidirectional Encoder Representations from Transformers
TF-IDF	Term Frequency-Inverse Dense Frequency

1 Introduction

With the viability of internet, online social media platforms have become mediums for common people to share and express their thoughts and feelings freely and publicly, which broadly include several tech-empowered exercises, e.g., photo sharing, blogging, social gaming, social video sharing, business networking, comments & reviews and others. The information available over these social media is a rich resource for sentiment analysis or inferring other increasing uses and abuses. Reflecting the dark side of this viability, there is an exponential growth of harassment and stalking over these online medias which is commonly referred to as cyberbullying [23]. Cyberbullying is broadly categorized into different forms as it appears to be, e.g., racism (facial features, skin colour), sexism (male, female), physical appearance (ugly, fat), intelligence (ass, stupid) and others. This event of cyberbullying is anonymous[1], consequently, quite hard to trace, which often ends up with devastating effects. Therefore identifying cyberbullying is crucial in order to avoid any fatal incident caused by it. Several research activities have been observed on developing different machine learning and deep learning based methods for addressing the issue of cyberbullying in recent years.

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus[2]. A mild to moderate respiratory disorder can be observed if infected with COVID-19 and it does not require any kind of special medical treatments but the person having any earlier medical issues, e.g., cardiovascular disease, diabetes, chronic respiratory disease, and cancer may encounter serious sickness. The virus spreads between people when someone has close contact with an infected person through either of droplets of saliva or discharge from the nose. It has been observed that people who are having any kind of symptoms are being abandoned, dumped completely, disowned and socially deprived as well. There have been quite a few instances where people were also disowned their family members for testing positive with COVID-19. Those people were bullied over social media, instead of spreading awareness, they have been mistreated, refused for medical assistance. Therefore, identifying these instances have become utmost important in order to prevent cyberbullying to occur any further or for taking any appropriate action against the offender. These motivate us to instigate a thorough research over how the phenomenon of cyberbullying takes place amid this pandemic and its effective detection.

[1] https://cyberbullying.org/

[2] https://www.who.int/health-topics/coronavirus#tab=tab_1

Table 1 Code-switched tweet instances of cyberbullying

Tweet utterance	Data sample
Code-switched tweet sample	O godi media ke dalal nahi h. Kya nizamuddin markaz ki laboratory me hi corona virus produced huwa tha. Don't be such fool.

Code-switching refers to fluid alternation between multiple languages in a single post/utterance [16]. In multilingual societies like India, code-switched text is very common [17]. Consequently, processing of code-switched text has received an exponential interest and attention from the NLP research community [20]. Therefore code-switched text draws words and linguistic structures from multiple languages, use of language-specific parallel word embeddings for processing such text could be useful and essential. Table 1 illustrates our problem definition:

In the aforementioned example, there are existence of two different languages, i.e., English and Hindi. Also, the sample talks about a person being accused of produced corona virus as he or she might go to a place called nizamuddin markaz. The focus of our present study is to identify these events of cyberbullying from the code-switched languages, which is, to the best of our knowledge, is the first of its kind. In order to foster this, we collect, develop, annotate and contribute a code-switched corpus to detect cyberbullying, taking the ongoing COVID-19 pandemic into account. The main contributions of this work are as follows:

i. We curate a new annotated code-switched (En-Hi) dataset for facilitating cyberbullying identification in COVID-19 perspective, which is one of the major contributions of our work.
ii. We study and investigate the role of individual language-specific word embeddings, e.g., English & Hindi; also using the concatenation of these embeddings by projecting them into similar dimension, to handle the code-switched texts.
iii. We empirically experimented and introduced several baselines (both ML and DL models) for the curated corpus and showed that deep ensemble model is significantly more effective when compared to their base classifiers.

The rest of the paper is organized as follows: Section 2 represents a brief survey of previous works solving the objective task. Section 3 defines the details of the proposed dataset to accomplish the task. The proposed frameworks have been explained in Sect. 4. Section 5 presents experimental evaluation along with the obtained results. Conclusion of the work is elucidated in Sect. 6.

2 Related works

In this section, we have presented different research activities concerned NLP-based detection of cyberbullying, as well as computational models of code-switching.

Al-Garadi et al. [1] proposed a framework for cyberbullying detection based on a set of unique features derived from Twitter, e.g., network, activity, user, and tweet content. Subsequently, they developed a supervised machine learning model for detecting cyberbullying.

The performance of their proposed framework, in terms of area under the receiver-operating characteristic curve is 0.943 and a F-measure of 0.936, which is quite significant for such kind of feature-driven model. In 2016, Zhao et al. [27] developed a cyberbullying identification model based on representation learning approach. Along with a pre-defined list of insulting words, they have taken both Bag-of-words and latent semantic features under consideration. As a text classifier they employed SVM and achieved a F1-score of 0.78. Authors also used Twitter data for the task. Again, a predictive model of cyberbullying incidents has been proposed by Hosseinmardi et al. [11], where the authors considered text captions along with comments of an Instagram post. Subsequently, they extracted profanity and linguistic contents of the text caption, and also the social graph parameters and temporal content behavior. Using logistic regression classifier, the model had attained a F1-score of 0.85. It has often been observed that there is a trend of using non-standard forms of words while posting or commenting in social media, which is also an influencing reason behind the high misclassification error. Keeping the aforesaid in mind, Zhang et al. [26] proposed a pronunciation based convolutional neural network (PCNN) to address this issue. They have considered both non-standard form of words and misspelled words and used their phoneme codes as input to the deep convolution neural network. Authors have evaluated their novel model over two different social media, e.g., Twitter and FormSpring. This PCNN network achieved 0.56 in terms of F1-score. Following this deep learning based approach, Rosa et al. [19] proposed a hybrid deep network consisting of CNN and LSTM for solving the objective task. Authors had experimented over a FormSpring dataset. Also they had tested using three different text representations, namely, Google-News word embedding, FormSpring word embedd503ing and Twitter word embedding. Experimental results had shown that the hybrid model of CNN-LSTM outperformed other deep learning and machine learning models used in the experiment by scoring 0.84 as F1-score.

The phenomenon of code-switched language is constantly getting attention of NLP research community over the last few years. In any multilingual society, the existence of code-switched language can be observed. In various research domains like detecting sarcasm, offensive and hatespeech detection, sentiment analysis, rumor detection; the existence of code-switched language has drawn the research recognition. In [18], Rao & Devi have shown entity extraction from code-switched data. In their work, they have demonstrated the challenges of processing code-switched data such as ambiguity, occurrence of non-standard words and also, existence of low resource languages for entity extraction. Authors have introduced different machine learning and deep learning bases approaches such as Conditional Random Field(CRF) along with rule based system, LSTMs have been used to extract entities from both Hindi-English and Tamil-English code-switched data. Jaech et al. proposed a character-word model which is hierarchical in nature for identifying language from code-switched data in [13]. The proposed work is able to provide a label, i.e., the language from code-switched text by applying a CNN to a whitespace-delimited word's Unicode character sequence and assigning a word vector. Subsequently, a bidirectional LSTM recurrent neural network (RNN) maps a sequence of such word vectors to a label (a language). The model is able to identify a language from such text at word level with 0.95 and 0.94 in terms of F1 score for English and Spanish language, respectively. Involvement of code-switched text has also observed in case of user intent classification and slot filling. In [14], Krishnan et al. presented a zero-shot learning approach for joint intent classification and slot filling from multilingual code-switched text. To accomplish the objective task, authors have proposed by augmenting monolingual data making use of multilingual code-switched text through random translations to enhance the neutrality of transformer's language. All the experiments have been done over English and

Haitian Creole code-switched dataset, which is human annotated and collected from Twitter. Authors achieved 87.92% and 91.03% of F1 score for the task of intent classification and slot filling, respectively. Sentiment analysis from code-switched text have also been an evolving research area. Shakeel et al. proposed a deep learning based approach for classifying sentiment from code-switched short text [22]. Authors have also developed a Urdu-English code-switched corpus collected from Twitter for the intended task. Stacked-CNN and Stacked-LSTM have been used for classifying sentiment and reported F1 score is 0.65 using ELMo. In [3], Bansal et al. presented a study of humour, sarcasm and hate speech detection from code-switched text. Their approach involves encoding different switching features in order to improve the performance of detecting humour, sarcasm and hate speech. The reported results clearly indicates the significant performance improvement over the baseline models.

3 Data description

To enable the exploration of cyberbullying detection on code-switched data during this current pandemic, we introduce a new dataset consisting of tweet utterances.

3.1 Data collection

We have collected tweets from January 28, 2020, leveraging Twitter's streaming API[3] and Tweepy[4] to follow specific keywords and account that are trending at this current pandemic. While collecting tweets, we have also used Twitter's search API[5] on the same keywords to gather related historical tweets. We have collected over 22k tweets from the inception until April, 2020. As mentioned earlier, our fundamental motivation is two-fold, i.e., detection of cyberbullying instances from code-switched text and also amid this ongoing pandemic, COVID-19. Therefore, we capture conversations associated with coronavirus outbreak using the following hashtags:

1. Coronavirus
2. COVID19
3. Lockdown
4. CoronavirusDelhi
5. CoronavirusOutbreak
6. CoronavirusPandemic

Note that, some hashtag overlapping can be seen as inclusion of a particular hashtag can be a substring of another. We keep all these occurrences for having a good measurement. Also, while collecting the data, there have been instances where monolingual texts, i.e., texts entirely either in English or in Hindi, scrapped too. In those cases, we have excluded such samples as our primary objective is to identify cyberbullying from code-switched text. Table 2 shows our data inclusion or exclusion criteria with example. Also, there are many

[3] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data
[4] https://www.tweepy.org/
[5] https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

Table 2 Data inclusion and exclusion example

Activity	Data sample
Data inclusion (Code-switched text)	hey why you such an asshole? why thank yuh to friends! Sale chutiye ek number ka!
Data exclusion (Monolingual text)	National capital is reporting hospitalisations like other cities, but the proportion of people requiring in ICU bed is much higher

variations that may exist in case of any occurrence of code-switched text, for example, a text may contain Hindi but written in English text, Hindi written in Hindi text which we have included but we excluded such text which are only in Hindi and also written in Hindi or entire sentence is in English, i.e., monolingual.

3.2 Annotation process

The entire data annotation process we opted is using Active Learning [21]. We, first, manually annotate 2k randomly selected tweet utterances into two classes, namely, *bully* and *nonbully*. Then the step-by-step annotation process is as follows:

1. Start with manually annotated small set of tweet instances
2. Train text classifiers on these annotated instances
3. For n-steps:
    (a) Select the next set of the most promising instances of tweets using entropy, where we rank the instances by their prediction entropy and only pick the top instances which have highest entropy values. These tweets are manually annotated. Tweets with entropy below a threshold are labeled by the predicted tags of the model.
    (b) Trained the model on all tweet instances selected so far.
    (c) Evaluate the trained model on the test set.
4. End of the annotation process.

In order to maintain the quality of annotations, we processed this Active Learning approach with the help of different ML and DL frameworks (please refer to Sect. 4). Thus, we obtain our fully annotated code-switched dataset. In order to show that our annotation is effective, we compare this approach with the random annotation strategy. Figure 1 illustrates the efficiency of our adopted Active Learning method.

The very detail of our final code-switched dataset can be seen in Table 3. Also Fig. 2 shows class-wise distribution of tweet utterances.

Presence of Profane words: Use of abusive words has been repeatedly observed to cyberbullying. Initial analysis of our proposed dataset shows that depending on profane words usage can neither lead to high precision nor high recall for identifying cyberbullying. For this, we have calculated $P(B \mid Pr)$ and $P(Pr \mid B)$, where B indicates Bully and Pr indicates

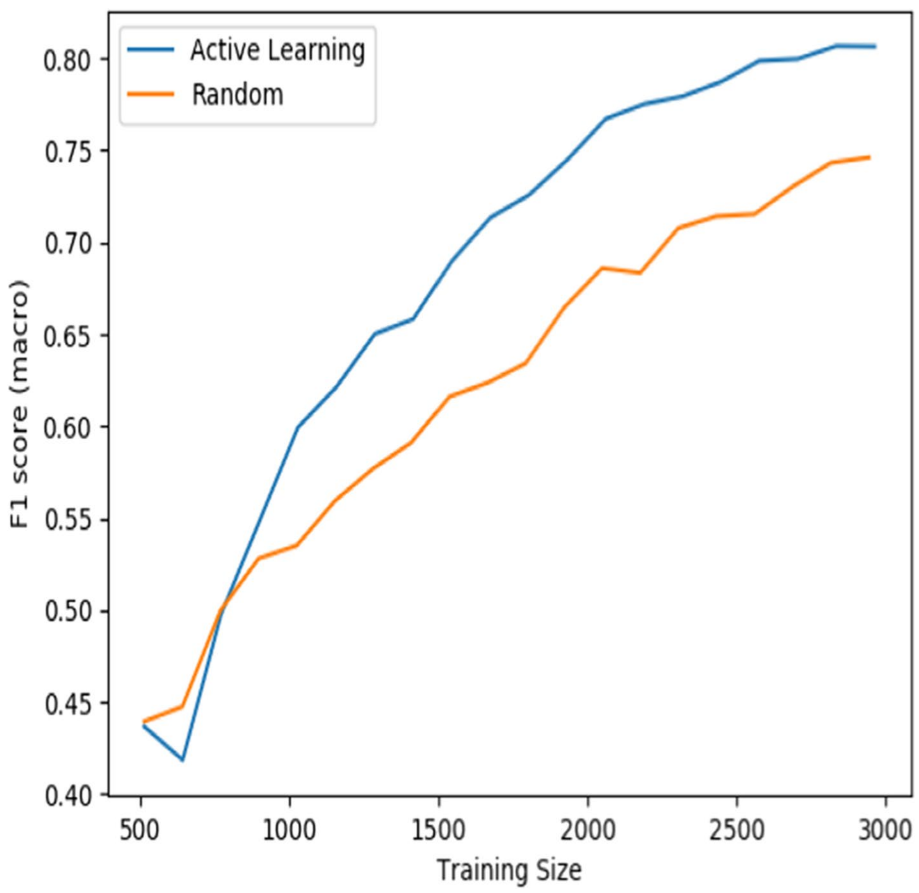


Fig. 1 Active Learning vs. Randomly annotated instances

Profane. This method will have low precision as $P(B \mid Pr)$ is not close to 1. Profane word list based methods will also have a low recall as $P(Pr \mid B)$ is not close to 1.

4 Proposed methodologies

In this section, we have described different machine learning and deep learning architectures developed and accordingly their ensemble frameworks for accomplishing the objective task. For machine learning classifiers, we have used Support Vector Machine (SVM) and Logistic Regression (LR). Also, Multi-Layer Perceptron (MLP), Convolution Neural

Table 3 Dataset Statistics

# bully post	# nonbully post	Total no. of instances
9921	12759	22,680

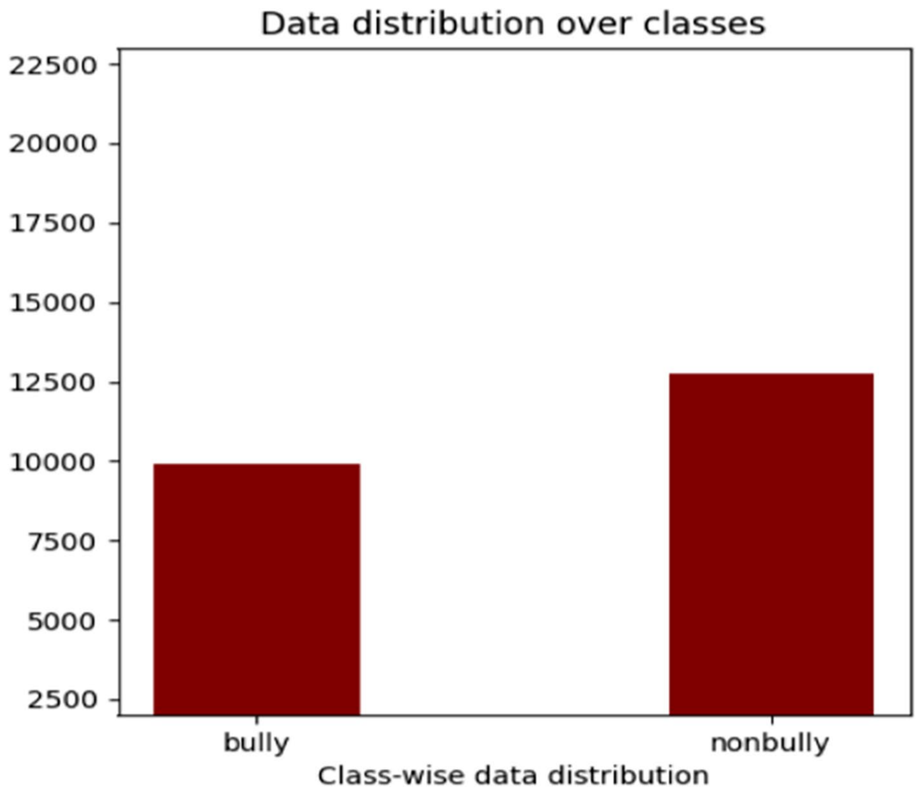


Fig. 2 Dataset Statistics

Network (CNN), BiLSTM with attention layer and BERT have been chosen as deep learning classifiers.

4.1 Word embeddings

In Natural Language Processing, word embedding plays crucial role as it provides a way to utilize a dense and efficient representation in which similar words have similar encodings. Note that, these embeddings are dense vectors of real numbers which are able to reveal hidden relationships between words. In recent years, the concepts of multilingual or bilingual word embeddings have gained much attention as words from two different source languages can be embedded into a same space which can solve certain tasks that involve reasoning across two or multiple languages.

We have experimented with two different language specific fasttext[6] word embeddings, i.e., English and Hindi as our code-switched corpus contains aforementioned two languages which are trained on Common Crawl and Wikipedia using fastText [10]. These models were trained using CBOW with position-weights which are helpful to capture positional

[6] https://fasttext.cc/docs/en/crawl-vectors.html

information, in fixed-size dimension 300, with character n-grams of length 5, a window of size 5 and 10 negatives. In our case, a vector of a word is being predicted based on context words. For example, we want to predict the vector of a particular word w_0 based on its context words $w_{-n}, ..., w_{-1}, w_1, ..., w_n$. A vector representation h of this context is obtained by considering the average of the corresponding word vectors, which can be defined as:

$$h = \sum_{i=-n;i\neq 0}^{n} u_{w_i} \tag{1}$$

However, code-switched text is indistinct in its syntactic, semantic and statistical properties, e.g., grammatical and syntactic constraints might introduce a collocation in word space. In order to achieve a synthetic presentation for En-Hi code-switched text, we therefore concatenate both the embeddings while utilizing word-level alignments, by performing canonical correlation analysis[7] (CCA) on two embeddings and project these into shared vector space where they are maximally correlated, as described in [7].

Let $\Sigma \in \mathbb{R}^{n_1 \times d}$ and $\Omega \in \mathbb{R}^{n_2 \times d}$ be vector space embeddings for two different languages, where each row corresponds to words. Since two embeddings are of different sizes, it may be noted that there might not exist translation for each and every word of Σ in Ω. Let $\Sigma' \subseteq \Sigma$, where each word in Σ' is translated to another word in $\Omega' \subseteq \Omega$. Let x and y be two corresponding vectors from Σ' and Ω', and v, w be the two projection directions, respectively. Then the projected vectors are:

$$x' = xv \tag{2}$$

$$y' = yw \tag{3}$$

and, the correlation between projected vectors can be as follows:

$$\rho(x', y') = \frac{E[x'y']}{\sqrt{E[x'^2]E[y'^2]}} \tag{4}$$

4.2 Architectures

4.2.1 Machine learning architectures

For the purpose of classification and active learning, several machine learning and deep learning based standard classifiers are first employed. Finally the outputs of these classifiers are combined together with the help of ensemble techniques to further improve the classification accuracy which in turn helps in improving the annotation process.

1. *Support Vector Machine*: (SVM) is considered as a competitive machine learning classification algorithm and was proposed in [4]. The objective of this algorithm is to find a hyperplane in an n-dimensional space (n indicates the number of features) that distinctly classifies the data points into pre-define categories. SVM is widely used for text categorization as described in [24]. The linear kernel is recommended for text categorization

[7] http://www.mathworks.com/help/stats/canoncorr.html

as the linear kernel performs efficiently when there are a lot of features [6]. Hence linear SVM has been used in our experiments.

2. *Logistic Regression*: (LR) performs well for binary class classification problem [8]. We have implemented logistic regression using liblinear, a library for large scale linear classification [6].

4.2.2 Deep learning architectures

1. *Multi-Layer Perceptron*: (MLP) is supposed to be the most intuitive neural network. In a typical supervised learning algorithm like classification, each of the input vectors has predefined class-label. The output of the perceptron network provides a class-belongingness probability. A loss function is also defined in order to measure the network performance. The function produces a high value whenever misclassification occurs, otherwise it generates a low value. Overfitting is an associated issue with this kind of network. To avoid this, a dropout layer is also employed.
2. *Convolution Neural Network*: (CNN) originally invented for computer vision, has been shown to achieve significant performance on text classification tasks [2]. CNN categorises text into a particular class based on the following intermediate computations: firstly, 1-dimensional convolving filters are used as n-gram detectors, where each of the filters is specialized in a closely-related family of n-grams. Then, for making a decision, max-pooling extracts the relevant n-grams. Therefore, based on the obtained information, network classifies the text [12].
3. *BiLSTM with attention layer* ($BiLSTM_{attn.}$): Bidirectional LSTM network intensifies the amount of available input information by encoding it in both forward and backward directions [28]. Attention mechanism [25] allows the model to learn what to attend based on the input sequence and what the model has computed yet.
4. *Bidirectional Encoder Representations from Transformers*: (BERT) has been used widely as it produces state-of-the-art results on a wide variety of NLP tasks, which include question answering (SQuAD v1.1), natural language inference (MNLI), text classification and others [5]. Unlike uni-directional models, the Transformer encoder considers the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to conclude that it's non-directional. This unique feature allows the model to learn the context of a word based on all of its surroundings (left and right contexts of the word).

Please refer to Fig. 3 for general architecture that we have used across all deep learning models.

4.3 Ensembling

Ensemble learning combines several machine learning or deep learning techniques into one predictive model in order to decrease the variance, bias, or improve predictions. The fundamental aim of ensemble methods is to combine the predictions of different meta classifiers built using different learning algorithms in order to improve robustness over a single classifier.

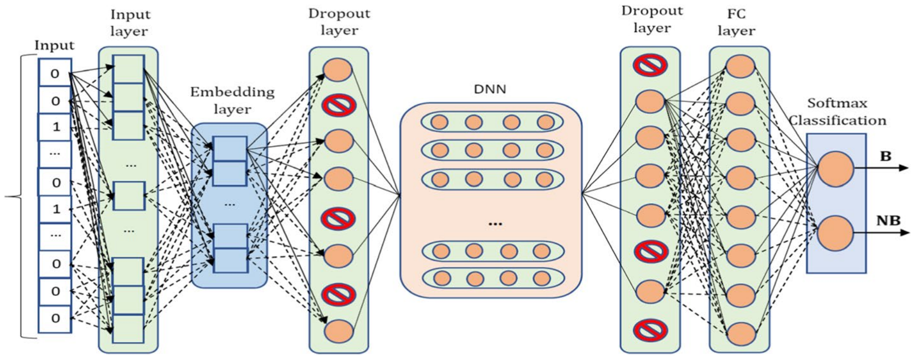


Fig. 3 Deep learning based architecture used in our experiments

4.3.1 Ensemble: machine learning approach

We propose a weighted majority voting based ensemble approach to leverage the efficiency of various state-of-the-art machine learning classifiers. Here, we have developed two feature-driven models, *viz.* Support Vector Machine (SVM) [4] and Logistic Regression (LR) [8]. We have used language specific word embedding models of embedding dimension 300 for each of the language corpora along with some TF-IDF/Count models. The language specific embeddings are trained on Common Crawl and Wikipedia using fastText[8]. These models were trained using CBOW (as described in [15]) with position-weights, in dimension 300, with character n-grams of length 5 as shown in [10]. As our proposed corpus contains English-Hindi (En-Hi) code-switched text, we have deployed English and Hindi embeddings separately and then we frame their ensemble model using weighted majority voting.

In this case, we predict the class label y by computing a weighted majority vote by associating a weight w_j with classifier C_j, precisely, weight implies the macro averaged f1 score (calculated over training data) of the corresponding classifier for the class label, y, (the idea is to assign priority to that classifier which was good in detecting that class). It can be defined as follows:

$$y = argmax_i \sum_{j=1}^{m} w_j \chi_A(C_j(x) = i) \tag{5}$$

where χ_A is the characteristic function $[\ C_j(x) = i \in A]$ and A is the set of unique class labels, which are bully and nonbully in our case.

While SVM or LR tries to learn only one hypothesis from the available training data, this weighted majority based ensemble model tries to develop a set of hypotheses and accordingly combine them for further use. Here, we have linearly combined the class belonging probabilities as produced by the individual classifiers (SVM and LR) along with the fine-tuned set of parameters.

[8] https://fasttext.cc/

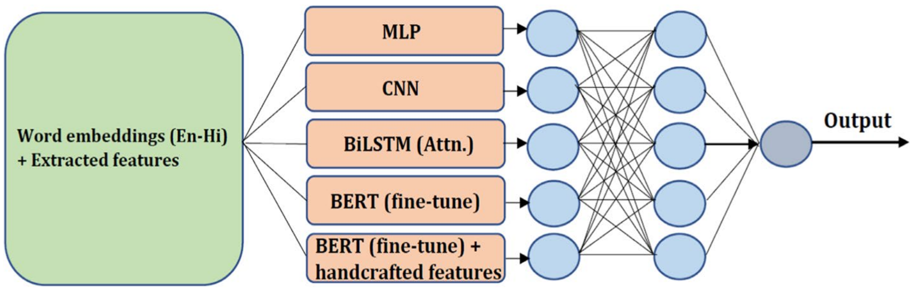

Word embeddings (En-Hi) + Extracted features
MLP
CNN
BiLSTM (Attn.)
BERT (fine-tune)
BERT (fine-tune) + handcrafted features
Output

Fig. 4 Steps of Ensembling Various Network Structures

4.3.2 Ensemble: deep learning approach

One way of enhancing the performances of neural network models is to frame an ensemble of deep neural network models. The architectures of different deep neural networks used in our experiment are described in Fig. 4. The weights for each of the deep networks are initialized at random using Xavier initialization technique [9] as it keeps the variance the same across every layer that helps to make the variance of the output to be equal to the variance of its input. Here, we have initialized weights randomly for the shake of stability of deep networks. Therefore, we can leverage the stability of deep neural networks by ensembling the decisions taken by the same network but with different random initializations, using the following equation:

$$y_{i,c} = \frac{1}{n}\sum_{j=1}^{n} \sigma_c(x_i, \theta_j) \mid \forall_c \in [1, C] \tag{6}$$

where $y_{i,c}$ denotes the output probability of ensemble having input x_i that belongs to a class c. σ_c is the logistic output averaged over n randomly initialized models. Fig. 4 shows, an overall architecture of the proposed deep ensemble approach.

In the initial stage, the rigorously pre-processed data is independently trained using MLP, CNN, BiLSTM with attention layer, BERT and hand-crafted feature driven BERT. Then each of the predictions given by the aforementioned models are summarised by initializing weights. Here, we have used Xavier weight initialization technique [9]. This in turn minimises the overall misclassification error rate by ensuring fine-grained classification. Subsequent sections consisting of the experimental results will demonstrate the improved efficiency of the deep ensemble network over individual deep networks.

5 Experimental evaluation

In this section, we present the results of different machine learning and deep learning models and their ensemble frameworks, evaluated on our proposed code-switched dataset. We have also explained the details of hyperparameter analysis along with the evaluation of our proposed framework. Also, we have provided a through error analysis on misclassified instances. All our experiments were conducted on a hybrid cluster of multiple GPUs comprised of RTX

Table 4 Optimised set of parameters for ML models; here SVM: Support Vector Machine, LR: Logistic Regression

Model	Parameters
SVM	Regularization parameter, C = 0.8; kernel = linear; class_weight = balanced; Tolerance = 1e-3
LR	penalty = l1; class_weight = balanced; solver = liblinear; multi_class = ovr

2080 Ti. It is to be noted that the dataset, codes and obtained results will be made publicly available on associated repository.

5.1 Training details

Machine learning models are implemented using Scikit-Learn 0.22.2. Keras 2.3.1 is used as the backend framework to train deep learning frameworks. Randomly sampled 80% of the data is used for training and 10% each is used for validation and testing. We have presented the performance of both the machine learning and deep learning frameworks on our validation set in the subsequent subsections. After tuning the hyperparameters sufficiently, we have reported the optimised set of hyperparameters for machine learning and deep learning models in Tables 4 and 5, respectively.

5.2 Experimental results

We have shown the mean of macro-averaged F1 scores across five runs of proposed machine learning and deep learning models in Tables 6 and 7, respectively.

In Table 6, SVM_{Hi}, SVM_{En}, LR_{Hi} and LR_{En} correspond to Support Vector Machine(SVM) and Logistic Regression(LR) trained using Hindi and English fasttext word embeddings, respectively. Thereby, the ensemble of these ML models is framed by combined word embedding (as described in Subsection 4.1) and weighted majority voting of the individual classification models (as described in Subsection 4.3). The obtained results suggest that ensemble of these ML models performed reasonably good than the individual classifiers. Although SVM and LR work well where there is clear margin of separation between both the classes and are memory efficient but in our case, the ensemble of both the ML models outperforms due to SVM or LP as an individual classifier lacks to learn the classification hypothesis.

Table 5 Optimised set of parameters for DL models; All the DL models have been trained on 30 epochs

Model	Hyperparameters
MLP	No. of layers = 4; Activation = ReLU; Regularization = Dropout; Loss = Entropy; Batch size = 16; learning rate = 1.0
CNN	No. of layers = 3, Pooling = Avg; Activation = Sigmoid; Optimizer = Adam; loss = MSE; Batch size = 16; learning rate = 0.001
BiLSTM with attention	batch size = 32; Activation = sigmoid; Optimizer = Adam; loss = binary_crossentropy; dropout probability = 0.5
BERT	batch size = 16; learning rate = 2×10^{-5}; dropout probability = 0.5

Table 6 Performance of machine learning models in terms of macro-averaged f1

Model	performance
SVM_{Hi}	0.66
SVM_{En}	0.60
LR_{Hi}	0.61
LR_{En}	0.67
Ensemble	0.69

Table 7 shows that ensembling of randomly initialized deep networks achieves a significant improvement in terms of macro-averaged f1 score, over ML models (both individual and ensemble of ML models). Also, the experimental results demonstrate that an ensemble of neural networks is always better than a single classifier. The fundamental reason behind this significant performance of deep ensemble model can be the ability of minimising the squared error as compared to the individual predictors of the classifier and also with better generalisation.

5.3 Key observations

Figure 5(a) shows the comparison between the prediction quality on the validation set and on the test set. The graph conveys the effectiveness of our proposed deep ensemble model. In order to show our deep ensemble model has been sufficiently trained, we have shown validation accuracy over a number of epochs in Fig. 5(b). From the Table 8, we have seen that BiLSTM with attention layer has outperformed MLP and CNN as it has the cell memory gate based architecture for processing of text bidirectionally.

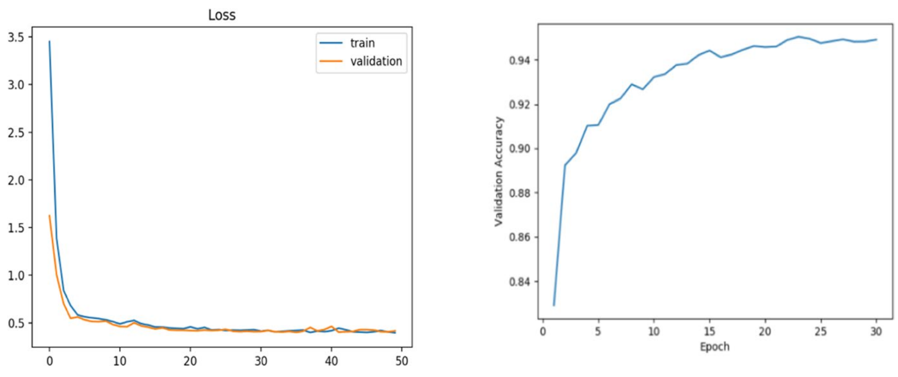
5.3.1 Error analysis

An overall error analysis has been conducted by manually checking the predicted misclassified tweet samples. We observe that our proposed model is not able to classify a post or comment that contains profane words, e.g., “bitch”, “mad”, which are likely to appear in bullying posts. It may be noted that occurrences of such words need not imply that particular post to be bullying. Few possible reasons behind these misclassifications enlisted below:

Table 7 Performance of Deep learning models in terms of macro-averaged f1

Model	performance
MLP	0.76
CNN	0.79
BiLSTM with Attn.	0.79
BERT(fine tuning)	0.81
BERT(fine tuning + handcrafted features)	0.86
Ensemble	**0.93**

The performance of our proposed model is highlighted in bold



(a) Model loss on both train and validation (b) Model accuracy on validation set

Fig. 5 Learning of our proposed deep ensemble model indicates a good fit

- Existence of obscene words in the posts or comments: If a certain post contains any abusive word, our model categorised it as in bully class.
- We notice that our corpus contains non-standard English and Hindi words which our model is not able to interpret properly to build a vector representation.
- The lengths of most of the tweet utterances are relatively small, therefore, our proposed model was not able to capture the proper context, in case there exists any sarcasm.
- The language Hindi has dialectal variations in comparison with English language.

Table 8 contains some misclassified tweet samples and the reasons for the misclassification.

Table 8 Misclassified instances and reasons behind misclassification

Instances	Predicted	Original	Possible reason
Any views on the Nizamuddin Maulana and his viral audio sweetheart???? Shayed nahi h	nonbully	bully	Absence of any profane word. Our model is able to capture the sarcasm of this instance
O godi media ke dalal nahi h. Kya nizamuddin markaz ki laboratory me hi Corona virus produced huwa tha. Don't be such a fool!	bully	nonbully	Due to existence of the words, e.g., 'dalal(a negative word in Hindi)' and 'fool'
hey why you such an asshole? why thank yuh to friends! Sale chutiye ek number ka!	bully	nonbully	Presence of words like 'asshole', 'sale' and 'chutiye'. Here, 'sale' and 'chutiye' are used as profane words in Hindi

6 Conclusions and future works

In this paper, we introduce a new exploratory area, i.e., identifying cyberbullying from code-switched text and also, develop and contribute an annotated code-switched corpus snowballed from Twitter. We have also shown cyberbullying is still taking place amid this pandemic. Here, we have shown that ensembling deep neural networks with random initialization of weights and with the help of parallel language specific word embeddings can achieve state-of-the-art performance for identifying cyberbullying in code-switched text. As the corpus is new, state-of-the-art models as well as their ensembles have been used to provide a base-line to identify cyberbullying from code-switched text.

In the future, we would like to consider multimodal fusion in order to predict the instances of cyberbully in a more fine-grained way. Also, we would like to develop a multi-task learning approach where the framework is able to understand hidden sarcasm in the post along with capturing information regarding cyberbullying.

Acknowledgements This publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya Ph.D. Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (Formerly Media Lab Asia)

References

1. Al-garadi MA, Varathan KD, Ravana SD (2016) Cybercrime detection in online communications: The experimental case of cyberbullying detection in the twitter network. Comput Hum Behav 63:433–443
2. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. In: arXiv preprint arXiv:180301271
3. Bansal S et al (2020) Code-switching patterns can be an effective route to improve performance of downstream NLP applications: A case study of humour, sarcasm and hate speech detection. In: arXiv preprint arXiv:200502295
4. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
5. Devlin J et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. In: arXiv preprint arXiv:181004805
6. Fan RE et al (2008) Liblinear: A library for large linear classification. J Mach Learn Res 9:1871–1874
7. Faruqui M, Dyer C (2014) Improving vector space word representations using multilingual correlation. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp 462–471
8. Genkin A, Lewis DD, Madigan D (2007) Large-scale bayesian logistic regression for text categorization. Technometrics 49(3), 291–304
9. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp 249–256
10. Grave E et al (2018) Learning word vectors for 157 languages. In: arXiv preprint arXiv:180206893
11. Hosseinmardi H et al. (2016) Prediction of cyberbullying incidents in a media-based social network. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, pp 186–192
12. Jacovi A, Shalom OS, Goldberg Y (2018) Understanding convolutional neural networks for text classification. In: arXiv preprint arXiv:180908037
13. Jaech A et al (2016) Hierarchical character-word models for language identification. In: arXiv preprint arXiv:160803030
14. Krishnan J et al (2021) Multilingual code-switching for zero-shot cross-lingual intent prediction and slot filling. In: arXiv preprint arXiv:210307792
15. Mikolov T et al (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp 3111–3119

16. Myers-Scotton C (1997) Duelling languages: Grammatical structure in codeswitching. Oxford University Press
17. Parshad RD, Bhowmick S, Chand V, Kumari N, Sinha N (2016) What is india speaking? exploring the "hinglish" invasion. Physica A: Statistical Mechanics and its Applications 449:375–389
18. Rao PR, Devi SL (2016) CMEE-IL: Code mix entity extraction in Indian languages from social media text@ FIRE 2016-An overview. In: FIRE (Working Notes). pp 289–295
19. Rosa H et al (2018) A "deeper" look at detecting cyberbullying in social networks. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, pp 1–8
20. Rudra K et al (2016) Understanding language preference for expression of opinion and sentiment: What do Hindi-English speakers do on twitter? In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp 1131–1141
21. Settles B, Craven M, Friedland L (2008) Active learning with real annotation costs. In: Proceedings of the NIPS workshop on cost-sensitive learning. Vancouver, CA, pp 1–10
22. Shakeel MH, Karim A (2020) Adapting deep learning for sentiment classification of code-switched informal short text. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing. pp 903–906
23. Smith PK et al (2008) Cyberbullying: Its nature and impact in secondary school pupils. J Child Psychol Psychiatry 49(4):376–385
24. Tong S, Koller D (2001) Support vector machine active learning with applications to text classification. J Mach Learn Res 2:45–66
25. Vaswani A et al (2017) Attention is all you need. In: Advances in neural information processing systems. pp 5998–6008
26. Zhang X et al (2016) Cyberbullying detection with a pronunciation based convolutional neural network. In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp 740–745
27. Zhao R, Zhou A, Mao K (2016) Automatic detection of cyberbullying on social networks based on bullying features. In: Proceedings of the 17th international conference on distributed computing and networking. pp 1–6
28. Zhou P et al (2016) Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. arXiv preprint arXiv:161106639

Authors and Affiliations

Sayanta Paul[1] · Sriparna Saha[1] · Jyoti Prakash Singh[2]

Sayanta Paul
1811cs16@iitp.ac.in

Jyoti Prakash Singh
jps@nitp.ac.in

[1] Indian Institute of Technology Patna, Bihta, India

[2] National Institute of Technology Patna, Patna, India