Hostility Detection in Online Hindi-English Code-Mixed Conversations

Aditi Bagora
Indian Institute of Technology Hyderabad
Hyderabad, Telangana, India
cs21mtech14007@iith.ac.in

Kamal Shreshtha
Indian Institute of Technology Hyderabad
Hyderabad, Telangana, India
cs21mtech16001@iith.ac.in

Kaushal Kumar Maurya
Indian Institute of Technology Hyderabad
Hyderabad, Telangana, India
cs18resch11003@iith.ac.in

Maunendra Sankar Desarkar
Indian Institute of Technology Hyderabad
Hyderabad, Telangana, India
maunendra@cse.iith.ac.in

ABSTRACT

With the rise in accessibility and popularity of various social media platforms, people have started expressing and communicating their ideas, opinions, and interests online. While these platforms are active sources of entertainment and idea-sharing, they also attract hostile and offensive content equally. Identification of hostile posts is an essential and challenging task. In particular, Hindi-English Code-Mixed online posts of conversational nature (which have a hierarchy of posts, comments, and replies) have escalated the challenges. There are two major challenges: (1) the complex structure of Code-Mixed text and (2) filtering the relevant previous context for a given utterance. To overcome these challenges, in this paper, we propose a novel hierarchical neural network architecture to identify hostile posts/comments/replies in online Hindi-English Code-Mixed conversations. We leverage large multilingual pre-trained (mLPT) models like mBERT, XLMR, and MuRIL. The mLPT models provide a rich representation of code-mix text and hierarchical modeling leads to a natural abstraction and selection of the relevant context. The propose model consistently outperformed all the baselines and emerged as a state-of-the-art performing model. We conducted multiple analyses and ablation studies to prove the robustness of the proposed model.

CCS CONCEPTS

• **Computing methodologies** → *Artificial intelligence*; *Natural language processing*.

KEYWORDS

Neural networks, hostility detection, Code-Mixed data

ACM Reference Format:
Aditi Bagora, Kamal Shreshtha, Kaushal Kumar Maurya, and Maunendra Sankar Desarkar. 2022. Hostility Detection in Online Hindi-English Code-Mixed Conversations. In *Proceedings of 14th ACM Web Science Conference 2022 (WebSci '22).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3501247.3531579

WebSci '22, June 26–29, 2022, Barcelona, Spain

ACM ISBN 978-1-4503-9191-7/22/06…$15.00
https://doi.org/10.1145/3501247.3531579

1 INTRODUCTION

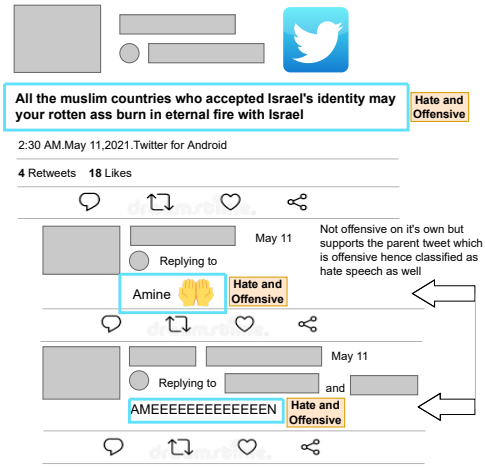

Figure 1: Example Code-Mixed tweet in an online conversation

Social media platforms like Facebook, Twitter, etc., have millions of active users. These are common platforms to express ideas, opinions, concerns, and creativity to a wide audience. Each user on a social media platform can create and consume the content. Recently, there has been an enormous increase in fake, hostile, and inappropriate content on social media. Prevalence of such content in social media impacts the user experience and can result in unwanted incidents. They create a negative atmosphere [23] and this negativity gets directly associated with the platform itself [28]. Sometimes, the hateful content is targeted directly or indirectly at a person or a community. This targeted offense leads to demotivation and depression in individuals [4], it also creates a bullying effect as the hatred keeps building up against an individual or community [19]. Filtering such content becomes necessary to make a positive, healthy, and clean online atmosphere.

Some social media platforms have recently started filtering hate content targeted at specific groups or individuals. Still, the chain of hostile content increases much faster and targets a broader range

of audiences with minimal control. Twitter on 9th July 2019[1] announced that it would be filtering hate content targeted at any religious group. According to recent data released by Facebook, between April 2021 and June 2021, Facebook removed or flagged 31.5 million contents containing hate speech. This is a massive increase compared to the first quarter number of 25.2 million as reported by the same organization [2].

Despite the efforts made by these platforms, they are still facing challenges in recognizing such hostile content due to the diverse nature of posts and responses. The users' variety of languages and dialects while generating posts or responses makes identifying and controlling such content even more challenging.

The hostility detection problem has been in multiple levels and setups. In the first setup, hostile contents are identified in English text. It is a relatively easy task due to the availability of English resources [10]. The second challenging setup is setting up multilingualism: social media posts can be written in any language. The classification of text becomes more complicated when different languages are involved in the contents [1]. Although large deep learning models are capable of learning language constructs, the problem with multilingualism is the scarcity of labeled data in different languages. These models need a sufficient amount of labeled data for training to perform and learn better. The third challenging setup is Code-Mixing. Since online platforms are informal modes of communication, the language constructs may not be followed. Users can use a mixture of languages within a single sentence or text, also called Code-Mixed text. The texts can include words or phrases from multiple languages, which leads to complex structure, and it poses an additional challenge in the modeling [14],[15].

One more challenging setup involves conversational Code-Mixed data, which is the focus of this paper. To be more specific, we consider hostility detection in conversational Hindi-English code mixed data in this work. Hostility detection in conversational code-mixed setup poses additional challenges due to the conversational type of structure. The conversational posts on social media platforms have a typical hierarchy structure which is a tuple of <POST, COMMENT, REPLY > as shown in Figure-3. There can be several comments for a given post, and for each comment, there can be several replies. For English-Hindi data, each tuple element can be Hindi-English Code-Mixed, only in English, only in Hindi, in romanized Hindi, combinations of these, etc., which leads to complex input patterns. The label for replies or comments is highly influenced by parent text/context. For example, in Figure-1 the last reply text is neutral but supports the Hate and offensive (HOF) post, so the assigned label for reply is HOF. Moreover, in a chain of several <POST, COMMENT, REPLY> tuples, identification of relevant context is essential and challenging. In summary, although the Code-Mixing conversational style is preferable and practical scenarios in social media platforms, they pose significant challenges for hostility detection.

We approach this challenging setup with hierarchical neural network modeling. Following the conversational structure of online tweets, we propose a hierarchical model. As the parent context is key for final predictions (particularly for comments and replies), the hierarchical model provides abstractive and selective context. The posts are stand-alone texts for which no previous context is available. For rich input text representations we leverage pre-trained models like mBERT [8], XLM-Roberta [5] and MuRIL [16]. We use popular classifiers like Random Forest, XGBoost, Logistic Regression, Voting Classifier, and direct fine-tuning of the pre-trained model. We took a recent baseline from literature and designed a few more strong baselines ourselves. The proposed model consistently outperformed all the baselines across two metrics.

The key contributions of the paper are mentioned below:

- We propose a novel hierarchical neural network architecture for hostility detection, which extracts abstractive and selective context representations for a given utterance.
- We designed a strong baseline with explicit context selection based on attention and heuristic weighting.
- We conducted an exhaustive analysis and ablation study to interpret the model output and prove the robustness.

2 RELATED WORK

This section reviews the two threads of existing literature on the hostility detection problem i.e., tradition and recent transformer-based approaches.

Most of the early approaches were based on classical machine learning algorithms and simple neural networks. [12] proposes a modeling approach using FastText and Word2Vec embeddings for multi-label toxicity detection, which works on tweets in Devanagari script. [2] compares classical machine learning classification models like SVM, Random Forest, Logistic Regression, and MLP, where SVM reported the best F_1-score for coarse-grain classification (involving two classes). Another approach to this problem is identifying and categorizing profane words. The work in [26] compares profane words in different languages and generations and categorizes words into different categories. An improvement over the lexical detection method is mentioned in [6] where the authors used crowdsourcing for collecting tweets and labels to distinguish between hostile, offensive, and neither hostile nor offensive using classical machine learning models like SVM, Naive Bayes, Decision Trees, etc. Amongst these models, the Logistic Regression and Linear SVM model performed better. Next, the attention was shifted towards CNN, LSTM, Bidirectional LSTM neural network models with FastText, Word2Vec, and GloVe word embeddings [27] for hostility detection.

Recently, Transformer based pre-trained models have gained attention for hostility detection across different languages and datasets. In [7], the authors propose a neural network-based hybrid model using LSTM and GRU with contextual representations from mBERT for hostility detection on Hindi posts. In [24], the authors have reported results with ANN and XGBoost models for fine-grained classification purposes where the representations were extracted from BERT. A comprehensive comparison of mBERT, IndicBERT models with LSTM, and BiLSTM with CNN was presented in [13] for coarse and fine-grained hostility classification on Hindi posts. The works in [1, 22] explore XLM-R and mBERT models by fine-tuning these models for hostile text classification problems. MuRIL representations were used with the CNN model for the same task in [21]. To tackle the problem of a limited number of labeled

[1] https://mspoweruser.com/twitter-bans-hate-speech-against-religious-groups/
[2] https://www.statista.com/chart/21704/hate-speech-content-removed-by-facebook/

examples in different languages, the authors in [17] propose a transfer learning approach based on fine-tuning of a pre-trained BERT model. In [3], the authors propose a cross-lingual transfer learning instead of creating an annotated dataset for low resource languages. The model uses bilingual word embedding-based classifiers. The authors use English as the source language and German as the target language.

There has been little work in the space of hostility detection in conversation-style text. [25] extracted representations from FastText for word embeddings and Doc2Vec for sentence embeddings before using the representations in SVM and Random Forest for hate speech detection in code-mixed data set. The concept of using context and multilingual BERT for representation was presented in [18]. The authors use a dual BERT encoder with averaging the representations from dense layers. Instead of using pre-trained embeddings, the work in [15] trains the representations on the Hindi-English Code-Mixed data set. Unlike this, we utilize pre-trained models because they generally boost the model performance, as reported across many NLP tasks [11, 29]. Closest to our work, [14] fine-tuned XLM-R model with additional MLP layer over raw and normalized Hindi-English Code-Mixed dataset, where normalization indicates romanization of Hindi text. The model does not consider the relevant context but rather only concatenates the previous text with the current one. Unlike this, we propose a hierarchical model for natural abstraction and context selection.

3 METHODOLOGY

This section includes a formal definition of the problem statement and then provides intuition and a description of the proposed hierarchical model.

3.1 Problem Statement

The Code-Mixed conversational structure on social media is a tuple of <POST, COMMENT, REPLY>. In hostility detection, the task is to determine whether a given Code-Mixed text (i.e., post/comment/reply) is Non-Hate-Offensive (NONE) or Hate and Offensive (HOF).

- (NONE) Non-Hate-Offensive - This post does not contain any Hate speech or profane, offensive content.
- (HOF) Hate and Offensive - This post contains Hate, offensive, and profane content.

Formally, for given post (P), comments ($C = C_1, C_2, \ldots C_N$) and replies ($R = R_1, R_2, \ldots R_K$) of Code-Mixed conversation, predict HOF or NONE labels for each of P/T , C_i and R_j text where i= 1,2,…N and j= 1,2,…K.

3.2 Proposed Model: Hierarchical Modelling

The proposed model focuses on considering the inherent hierarchy of social media conversation-type posts. We leverage this structure to build a hierarchical neural network model. As discussed in Section 1, the model should extract relevant (selective) and abstractive context for comments and replies.

In hierarchical model, first we feed available Code-Mixed inputs (i.e., only <POST> or <POST, COMMENT> or <POST, $R_{CONTEXT}$, REPLY>) through pre-trained multilingual (PT) models to obtain their individual contextual representations. If post has only one reply then the $R_{CONTEXT}$ for the reply is parent comment only. In other extreme when the post has k replies then the $R_{CONTEXT}$ for t^{th} replies is concatenation of comment and 1^{st} to $(t-1)^{th}$ replies. We consider mBERT, XLM-R, and MuRIL as pre-trained models in our experimentations. The contextual representation for POST (P), COMMENT/$R_{CONTEXT}$ (C) and REPLY (R) are h_p, h_c and h_r respectively. If the input is only post, we use a two-layer Multi-layer Perceptron (MLP) before obtaining softmax (SL) logits (h_s). To predict the label for the comment, contextual representations of post and comment are concatenated ($[h_c; h_p]$) and passed through the MLP layer. In the end, the MLP hidden output (h_L) is passed through the softmax layer (SL) to obtain the logits (h_s). In this way, the decision regarding the class of the comment is influenced by the content of the original post as well. Finally, for the prediction of any given reply, first contextual representations of the corresponding post and $R_{CONTEXT}$ are concatenated ($[h_c; h_p]$). The concatenated representation is passed through MLP to obtain an abstractive single hidden representation (h_{L1}). This h_{L1} is concatenated with the contextual representation of the reply under consideration ($[h_r; h_{L1}]$). We view these representations as selective representations, which further pass through another MLP layer (MLP') to obtain a new hidden state (h_{L2}). In the end, h_{L2} is passed through softmax-layers (SL) to obtain the logits values (h_s). Figure-2 presents an architectural diagram of the proposed hierarchical model. The formal modeling steps are shown below:

Processing for POST/COMMENT/$R_{CONTEXT}$/REPLY

POST(P)

- Step 1: h_p = PT(P)
- Step 2: h_L = MLP(h_p)
- Step 3: h_S = SL(h_L)

$R_{CONTEXT}$/COMMENT(C)

- Step 1: h_p= PT(P) , h_c = PT(C)
- Step 2: $h_{pc} = h_p;h_c$
- Step 3: h_L = MLP(h_{p_c})
- Step 4: h_S = SL(h_L)

REPLY(R)

- Step 1: h_p = PT(P), h_c = PT(C), h_r = PT(R)
- Step 2: $h_{pc} = h_p;h_c$
- Step 3: h_{L1} = MLP(h_{p_c})
- Step 4: $h_{rpc} = h_r; h_{L_1}$
- Step 5: h_{L2} = MLP'($h_{r_{pc}}$)
- Step 6: h_S = SL(h_{L_2})

4 EXPERIMENTAL SETUP

4.1 Data sets

We obtain the dataset from HASOC2021 challenge Sub-task 2 and use the Hindi-English Code-Mixed dataset[3] shared by the task organizers. The text in the dataset is in the usual Twitter posts format (has tuple <POST, COMMENT, REPLY>) with hashtags, URLs, mentions, etc. Each element of the tuple is annotated as either HOF (Hate and Offensive) or NONE (Non-Hate-Offensive) as shown in Figure-1.

[3]https://hasocfire.github.io/hasoc/2021/dataset.html

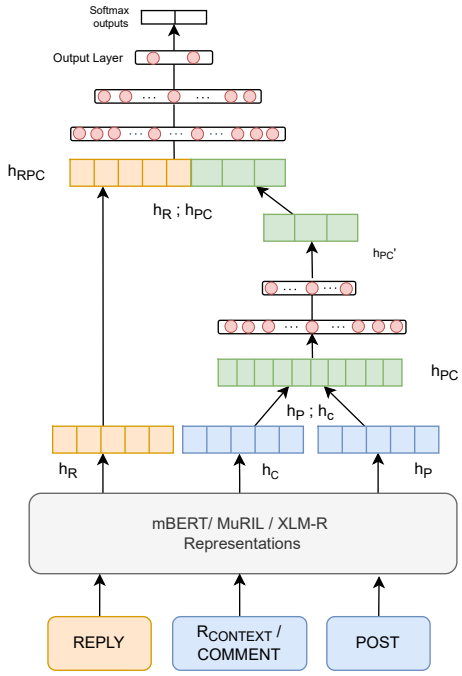

Figure 2: Architectural diagram of proposed Hierarchical model

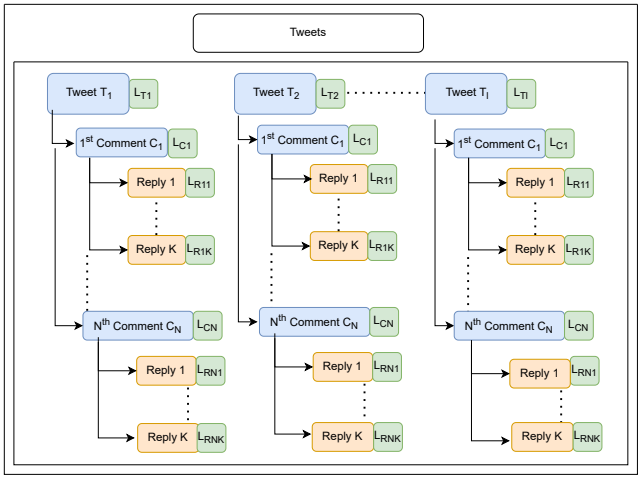

Figure 3: Conversational Code-Mixed Tweet structure

The dataset consists of 62 Code-Mixed conversations from Twitter. The dataset is divided into a 60:40 ratio to create a train and test dataset split. For the proposed hierarchical model, we flattened the examples from the train and test data. During this flattening, the examples are created by appending the previous context. As there is no context for a post, no context is added. For comment, the previous context is a post. Similarly, for a reply, the contexts are post, comment and previous replies (if any). The flattened views of the examples form the final train and test dataset.

The label distributions in both the training and test splits of the dataset are balanced. Detailed dataset statistics is shown in Table-1.

Dataset	#C	#E	HOF	NONE
Train set	37	2819	1309 (46%)	1510 (54%)
Test set	25	2092	1037 (50%)	1055 (50%)
Total	62	4911	2346 (48%)	2565 (52%)

Table 1: Conversational Code-Mixed dataset statistics, C = Conversations, E(Examples) = a flattened data point for the proposed model (see section-3.2 for detail).

4.2 Baselines

In this section, we describe the details of the baseline models. We use *Code-Mixed XLM-R* as a baseline from the literature [14]. Additionally, we propose a few more strong baselines due to the unavailability of other suitable baselines for the current task. The details of all the baselines (with mathematical descriptions) are presented in the appendix section.

4.2.1 Code-Mixed XLM-R. In this baseline, we use the model architecture from [14]. The architecture consists of learning the representation using XLM-R for normalized input texts. The representations are further pooled before passing through an MLP layer. Finally, the MLP output representations are fed through the softmax layer to generate the output logits.

4.2.2 Simple Concatenation Baseline (SCB). This model concatenates the post, comment, and reply text using a separator token ([SEP]) in the same order. The concatenated text is passed through the multilingual pre-trained model to obtain the contextual representation. The classification algorithms or fine-tuned models use these contextual representations to make predictions.

4.2.3 Weighted Context Baseline (WCB). In the most practical situation, simple concatenation is not a feasible solution. For example, in online posts, the context of any reply is more likely to lie in the most recent context than earlier ones. Inspired by this, we modified SCB and used the weighted pooling of the representations of the older context and the recent context. Weights are assigned heuristically. The current context is the last utterance of the conversation.

4.2.4 Selective Concatenation Baseline (SLCB). For any given utterance, the SCB baseline model concatenates all the utterances in a running conversation that happened before it (irrespective of different comments or reply chains). This model performs concatenation only if the utterance belongs to the same comment or reply chain.

4.2.5 Cosine Attentive Baseline (CAB). The context for a given utterance may lie anywhere within the conversational thread and not just in the recent reply. This baseline model relies on this intuition. We first obtain the representation of each element of the conversation tuple. Then, find the cosine distance between the representation of current text (post/comment/replies) with previous elements of conversation context. These cosine scores are passed through a softmax layer to obtain a probability distribution. We

take the weighted average of the representation of all contexts to obtain the final representation. The weights are the probability score.

The input text representation for all the baselines is obtained from three pre-trained models, mBERT, MuRIL, and XLM-R. We use four classifiers, i.e., Random Forest (RF), XGBoost (XGB), Logistic Regression (LR), and Voting Classifier (VC) for classification. Where VC internally consists of RF, XGB, and LR. Additionally, we also experimented with direct fine-tuning of the pre-trained models for the same task. The flattening approach of the conversational thread into model inputs for different baselines is different. Text concatenations were done using the [SEP] token between the texts (wherever applicable). The proposed and baselines models use the same pre-processing pipeline, which includes basic cleaning of text, i.e., removing HTML tags, web URLs, links, and mentions.

4.3 Implementation Details

The model extracts representations for <POST, COMMENT/$R_{CONTEXT}$, REPLY> using a fine-tuned representation learning model (m-BERT, XLM-R, and MuRIL). The representations of <POST, COMMENT/ $R_{CONTEXT}$> are concatenated and fed through a neural network that generates a latent representation of 300 dimension. This representation is concatenated with the representations of REPLY to obtain a vector representation of the context and text for classification. The concatenated 1068 dimension representation is passed through a neural network to generate the output logits and a softmax output that classifies between the other two classes. The model is trained with a learning rate of $1e-05$ for 4 epochs with AdamW optimizer and cross-entropy Loss. The code is and pre-trained models are publicly available.[4]

4.4 Evaluation Metrics

We use accuracy and F_1 score (macro) to evaluate proposed models and all the baselines.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$F_1 \text{ score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5 RESULTS AND DISCUSSIONS

The results for the baselines and the proposed model are shown in Table- 2. The order of different model performances according to the reported score is: Code-Mixed XLM-R < WBC < SCB < SLCB < CAB < Proposed Hierarchical Model. The results show that the proposed hierarchical (with direct fine-tuning) models consistently outperformed all the baselines across both evaluation metrics. This provides evidence that the proposed model automatically extracts selective and abstractive context.

The models are trained under different settings as discussed in section 4.2. The model performance varies based on the pre-trained models. The F_1 scores for WBC with MuRIL perform better as compared to mBERT and XLM-R pre-trained models. The F_1 scores of the proposed model increase when training is done under mBERT, MuRIL, XLM-R in that order. XLM-R is trained using a larger multilingual corpus. It is shown that an increase in model capacity increases model performance [9]. XLM-R gives better results for cross-lingual data [30] as it is trained with a large dataset in many languages.

As shown in Table-2 the performances of the models depend on the classifier being used. It shows that LR and VC perform better than other classifiers. The voting classifier gets the majority votes (mode) of the predictions from RF, XGBoost, and LR, which means that the prediction from VC does not entirely depend on one model; rather, it combines the leanings from multiple classifiers. XLM-R with LR performs better for cross-lingual classification task [30]. The proposed model gives the best results on direct fine-tuning. WBC performs closest to the hierarchical approach comparatively. The proposed model still has a 9% gain over WBC in terms of the F_1 score. Additionally, experimental results show that WCB performs better than SCB, indicating that giving more importance to the recent utterances is useful.

In summary, the proposed novel hierarchical model with XLM-R pre-trained multilingual model, which is fine-tuned to the task in hand, performs best on the online Code-Mixed Hindi-English conversational dataset.

6 ANALYSIS

6.1 Experimental Analysis-I

In this experimental analysis, we use LIME[20] analysis tool to explain the predictions from our best performing model (i.e., fine-tuned hierarchical XLM-R). The intuition behind this analysis is to visualize the attention of the model while assigning the prediction labels. A learned model will look at a specific set of words in the input sentence and generate predictions based on them.

Table-3 shows a few instances with the ground truth, model prediction values (for both the classes), and highlighted words that are focused by the model for generating the predictions. The table consists of examples for three different sets of input sentences; <POST, COMMENT, REPLY>. The first example is the post, along with its ground truth. Then the post is concatenated with comment (separated by a single space), and the ground truth is taken as that of comment. And finally, post + comment is concatenated with reply (also separated by a single space) with ground truth as that of reply. We follow the same pattern to construct examples 4,5,6 and 7,8,9. Figure-4 provides closer look at example-6 from Table-3 and it shows the predictions and corresponding word-level attention distribution.

Looking at this analysis of model attention, we can see that most of the instances with orange color highlighted words are, in fact, hate and offensive in nature (in the context of the input) that have a major contribution in predicting the label as "HOF." Similarly, instances with bluish color highlights are not hate and offensive and such instances are being predicted as "NONE." The model predicts "HOF" when encountering words like *"terrorism", "anti", "andh", "khujli"* whereas words like *"best", "Ok", "liberals", "Evidence"* contributes towards predicting "NONE". A more detailed look in Figure-4 for example 6 in Table-3 shows the contribution(attention placed by model) of words like "Kaha," "And," "Bhakts," "log" being focused to successfully predict "HOF," which is hate and offensive sentence. Also, some instances and predictions conflict with the ground truth in the same table. The instances in examples 7, 8, and

[4]https://github.com/AditiBagora/Hasoc2021CodeMix

Model	Method	Accuracy					F1 Score				
		RF	LR	XGB	VC	Direct FT	RF	LR	XGB	VC	Direct FT
CM-XLMR	XLM-R + Norm	-	-	-	-	0.61	-	-	-	-	0.46
SCB	mBERT	0.55	0.61	0.49	0.57	0.56	0.55	0.60	0.57	0.49	0.50
	MuRIL	0.50	0.40	0.45	0.46	0.57	0.50	0.29	0.45	0.45	0.51
	XLM-R	0.55	0.58	0.52	0.58	0.40	0.54	0.49	0.50	0.53	0.27
WBC	mBERT	0.62	0.59	0.61	0.62	0.66	0.61	0.57	0.60	0.61	0.64
	MuRIL	0.59	0.41	0.54	0.53	0.40	0.55	0.29	0.52	0.53	0.29
	XLM-R	0.64	0.64	0.59	0.64	0.66	0.60	0.62	0.57	0.61	0.65
SLCB	mBERT	0.64	0.55	0.60	0.62	0.66	0.58	0.57	0.54	0.58	0.61
	MuRIL	0.64	0.60	0.55	0.62	0.62	0.57	0.56	0.54	0.57	0.55
	XLM-R	0.64	0.62	0.61	0.65	0.40	0.62	0.60	0.59	0.63	0.27
CAB	mBERT	0.57	0.58	0.55	0.58	0.58	0.57	0.58	0.55	0.58	0.53
	MuRIL	0.60	0.59	0.61	0.65	0.58	0.60	0.58	0.61	0.64	0.54
	XLM-R	0.62	0.64	0.59	0.64	0.63	0.61	0.64	0.59	0.64	0.60
Hierarchial	mBERT	0.54	0.58	0.60	0.62	0.60	0.52	0.54	0.56	0.62	0.65
	MuRIL	0.59	0.63	0.62	0.64	0.63	0.55	0.61	0.60	0.64	0.67
	XLM-R	0.63	0.61	0.64	0.66	**0.68**	0.62	0.60	0.62	0.63	**0.72**

Table 2: Accuracy and F_1 scores for baselines and proposed model. Symbol '-' indicates that the results are not available. CM-XLMR = Code-Mixed XLM-R, RF = Random Forest, LR = Logistic Regression, XGB = XG-Boost, VC = Voting Classifier, Direct FT = Direct Fine-Tuning

9 are labeled as "NONE," but the model is predicting "HOF," which might be because even though the instances contain words like *"terrorism," "challenge," "khujli"*, the meaning of the sentences is not hate/offensive in nature. Because the sentences contained such words, the model is getting confused and predicting "HOF," which suggests that there is some room for improvement in the proposed model as future work.

6.2 Experimental Analysis-II

In this experiment, we hypothesize that the proposed model assigns the HOF labels for those sets of test examples that follow similar named entity distribution as the training dataset. To verify this hypothesis, we look at the distributions of named entities (with frequencies) in two subsets of datasets: (a) set of all the examples with HOF label from the training dataset and (b) set of all the examples with HOF predicted label (with proposed model) from the test dataset. From Table-5 we can conclude that the distribution trend is similar for both the datasets. This further concludes that the model follows similar entity distribution for HOF text. High correction coefficient scores for Pearson's and Kendall Rank Correlation, as shown in Table-4 validates the claims.

6.3 Experimental Analysis-III

To understand the prediction distribution of the model, in this section we have closer look at the confusion matrix as shown in the Table-5.

The ideal behavior of any prediction model should be - (a) lower false negative (FN) and false positive (FP) values and (b) higher true positive (TP) and true negative (TN) values. In practical scenarios, a higher false-positive value indicates a model predicting HOF even if it is NONE which is acceptable as it will cause alertness in public. But, having higher false negative values would lead to unrecognized hostile content floating in the community. Table-5 shows that the proposed model correctly predicts the actual label for most instances (TP+TN) and the predicted false-negative value (FN=188) is much lower than the false positive value (FP=483), which gives evidence that the model can be used in practical setups. However, high FP values provide room for improvement and the proposed model serves as a stepping stone for further research work.

6.4 Ablation Study

In the ablation study, we selectively remove some components from our proposed model and observe the model performance behavior. We carefully analyze the role and the necessity of each of these model components.

6.4.1 Based on Component Removal. There are five layers involved in the hierarchical model (see Figure-2). Layer 1 converts the concatenated post and comment to a 768 dimensional representation. Layer 2 projects these representations to 300 dimensions. The output from layer 2 is concatenated with the 768 dimensional representation of the reply text to obtain a concatenated representation of the 1068 dimension. Layer 3 projects this representation again to 768, which is passed through layer 4, followed by a softmax layer. In the ablation study, we will selectively remove some components from our proposed model and observe the predictions.

In Table-6, setup 1 is created by removing layer 1 and directly converting concatenated text to a 300 representational dimension. Setup 2 is formed by eliminating layer 2 and directly passing representations from layer 1. This will equally weight the input text to be classified and concatenate post and comment representations.

Exp.	GT	Predictions	Type	Texts with highlighted words
1	NONE	NONE 0.54 HOF 0.46	P	Ok but liberals are not 'anti-Hindu' ..
2	NONE	NONE 0.51 HOF 0.49	P+C	Ok but liberals are not 'anti-Hindu' .. You're anti-Hindu and non-liberal. Evidence maang lo bas, flood hon jaayega.
3	NONE	NONE 0.56 HOF 0.44	P+C+R	Ok but liberals are not 'anti-Hindu' .. You're anti-Hindu and non-liberal. Evidence maang lo bas, flood hon jaayega. Flood it.
4	NONE	NONE 0.62 HOF 0.38	P	Our prayers are with you KanganaRanaut . Get well soon .You are the best
5	HOF	NONE 0.50 HOF 0.50	P+C	Our prayers are with you KanganaRanaut . Get well soon .You are the best Andh Bhakts are still supporting her. Gadhi 🤣
6	HOF	NONE 0.16 HOF 0.84	P+C+R	Our prayers are with you KanganaRanaut . Get well soon .You are the best Andh Bhakts are still supporting her. Gadhi 🤣 kaha se aate he ye log
7	NONE	NONE 0.26 HOF 0.74	P	Religious conversion has become the biggest national challenge in India after terrorism. आपदा_में_धर्मपरिवर्तन_का_खेल
8	NONE	NONE 0.25 HOF 0.75	P+C	Religious conversion has become the biggest national challenge in India after terrorism. आपदा_में_धर्मपरिवर्तन_का_खेल If someone change his religion by his choose then what is your problem?
9	NONE	NONE 0.25 HOF 0.75	P+C+R	Religious conversion has become the biggest national challenge in India after terrorism. आपदा_में_धर्मपरिवर्तन_का_खेल If someone change his religion by his choose then what is your problem? Appne dekhona bhai, kyu khujli horahi he?

Table 3: LIME Analysis, *GT = Ground Truth, P = POST, C = COMMENT, R = REPLY

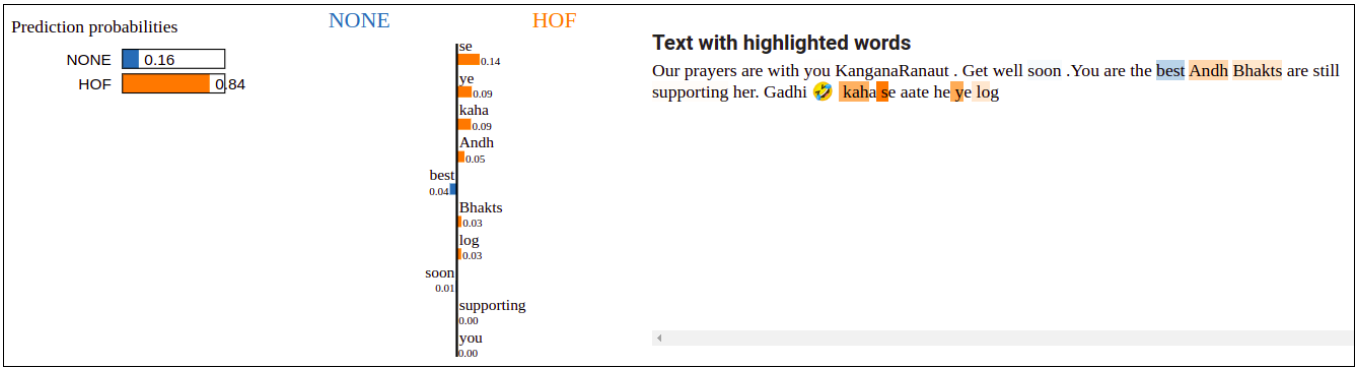

Figure 4: Closer look of an instance using LIME.

	Correlation Coefficient	p-value
Pearson's Correlation	0.963	1.445e-10
Kendall Rank Correlation	0.841	1.388e-06

Table 4: Correlation scores between 0 and 1

	Predicted NONE	Predicted HOF
Actual NONE	572 (TN)	483 (FP)
Actual HOF	188 (FN)	849 (TP)

Table 5: Confusion Matrix for the propose model. Where TP=True Positive, TN=True Negative, FN=False Negative, FP=False Positive.

Setup 3 is formed by removing linear layer 3 and directly passing the concatenated post, comment, and reply representations to layer 4. Setup 4 is formed by removing a combination of layers 1 and 2 and directly passing the representations of the concatenated post, comment, and reply (input text to be classified) to layer 3.

6.4.2 Based on Context Removal. In this section, we try to remove the context passed to the hierarchical model as input.

Setup 5 is created by removing COMMENT/$R_{CONTEXT}$ from the proposed model's architecture which means that the reply is classified based on contextual information from the post only. Similarly, Setup 6 and Setup 7 are created by removing POST, POST and COMMENT/$R_{CONTEXT}$ respectively. Setup 6 classifies a reply based on context from comments and replies only whereas Setup 7 does not include any previous context when classifying. The ablation study results are reported in Table-6. We can conclude that every layer and component involved is essential for the proposed model. Inclusion

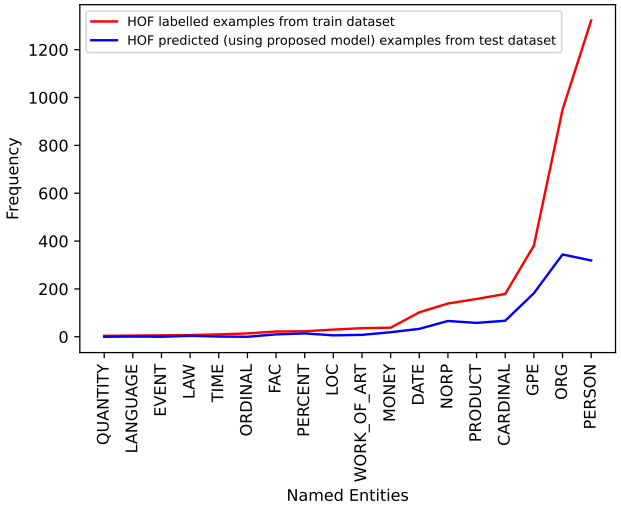

Figure 5: Distribution of Named Entities

Abalation Type	Setup	Accuracy	F_1 Score
No removal	**Hierarchical**	**0.679**	**0.716**
Component Removal	Setup 1	0.522	0.511
	Setup 2	0.524	0.408
	Setup 3	0.521	0.511
	Setup 4	0.518	0.510
Context Removal	Setup 5	0.519	0.506
	Setup 6	0.519	0.509
	Setup 7	0.512	0.393

Table 6: Ablation study results

of context is significant for the model as Setup 7 (without context) has the least F_1-score. Also, it shows that when the classification is done based on post only (Setup 5) or comment only (Setup 6), it gives similar results. The score for the context only is marginally better. Setup 2 shows that when the dimension of representations of context and reply is the same, the F_1 score decreases. Setup 4 indicates that if the representations of the post, comment, and reply to be classified are concatenated directly, then the model gives a slightly lesser F_1-score compared to Setup 1 and Setup 3. So, there is a need for all the model layers for better prediction accuracy and F_1 score.

7 CONCLUSION

This paper presented a novel hierarchical neural network architecture for detecting hate and offensive content in Hindi-English Code-Mixed conversations. It exploits the inherent hierarchy of the online social media conversational threads to tackle the discriminative task of classification. The proposed hierarchical model provides selective and abstractive context for a given utterance to boost the model performance. We showed that the model consistently outperforms all the baseline considered in this paper in terms of accuracy and F_1 score. Few natural extensions of the proposed model can be: (1) to detect offensive texts across different social media platforms, discussion forums, comment sections of news feeds, and chatbots (2) to detect hate and offensive content in multiple other languages and combination of such languages (Code-Mixed conversations) (3) to analyze content at the user level by looking at all the comments and replies made by a particular user for a given post, making it more personalized while detecting hate and offensive language.

REFERENCES

[1] Somnath Banerjee, Maulindu Sarkar, Nancy Agrawal, Punyajoy Saha, and Mithun Das. 2021. Exploring Transformer Based Models to Identify Hate Speech and Offensive Content in English and Indo-Aryan Languages. (11 2021).

[2] Mohit Bhardwaj, Md. Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. 2020. Hostility Detection Dataset in Hindi. *CoRR* abs/2011.03588 (2020). arXiv:2011.03588 https://arxiv.org/abs/2011.03588

[3] Irina Bigoulaeva, Viktor Hangya, and Alexander Fraser. 2021. Cross-lingual transfer learning for hate speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. 15–25.

[4] Ana-Maria Bucur, Marcos Zampieri, and Liviu P. Dinu. 2021. An Exploratory Analysis of the Relation between Offensive Language and Mental Health. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 3600–3606. https://doi.org/10.18653/v1/2021.findings-acl.315

[5] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[6] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 11. 512–515.

[7] Arkadipta De, Venkatesh Elangovan, Kaushal Kumar Maurya, and Maunendra Sankar Desarkar. 2021. Coarse and fine-grained hostility detection in Hindi posts using fine tuned multilingual embeddings. In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*. Springer, 201–212.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[9] Sumanth Doddapaneni, Gowtham Ramesh, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2021. A Primer on Pretrained Multilingual Language Models. *CoRR* abs/2107.00676 (2021). arXiv:2107.00676 https://arxiv.org/abs/2107.00676

[10] Ayush Gupta, Rohan Sukumaran, Kevin John, and Sundeep Teki. 2021. Hostility Detection and Covid-19 Fake News Detection in Social Media. *CoRR* abs/2101.05953 (2021). arXiv:2101.05953 https://arxiv.org/abs/2101.05953

[11] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*. PMLR, 4411–4421.

[12] Vikas Kumar Jha, Pa Hrudya, PN Vinu, Vishnu Vijayan, and Pa Prabaharan. 2020. DHOT-repository and classification of offensive tweets in the Hindi language. *Procedia Computer Science* 171 (2020), 2324–2333.

[13] Ramchandra Joshi, Rushabh Karnavat, Kaustubh Jirapure, and Ravirai Joshi. 2021. Evaluation of Deep Learning Models for Hostility Detection in Hindi Text. In *2021 6th International Conference for Convergence in Technology (I2CT)*. IEEE, 1–5.

[14] Aditya Kadam, Anmol Goel, Jivitesh Jain, Jushaan Singh Kalra, Mallika Subramanian, Manvith Reddy, Prashant Kodali, T. H. Arjun, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. Battling Hateful Content in Indic Languages HASOC '21. *Forum for Information Retrieval Evaluation (FIRE) 2021, CEUR Workshop Proceddings* abs/2110.12780. https://cdn.iiit.ac.in/cdn/precog.iiit.ac.in/pubs/2021_Sept_Battling_Hateful_Content_in_Indic_Languages_HASOC.pdf

[15] Satyajit Kamble and Aditya Joshi. December, 2018. Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models. *International Conference on Natural Language Processing, Patiala, India* abs/1811.05145 (December, 2018).

[16] Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. MuRIL: Multilingual Representations for Indian Languages. arXiv:2103.10730 [cs.CL]
[17] Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2019. A BERT-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*. Springer, 928–940.
[18] Ravindra Nayak and Raviraj Joshi. 2021. Contextual Hate Speech Detection in Code Mixed Text using Transformer Based Approaches. *Forum for Information Retrieval Evaluation (FIRE) 2021, CEUR Workshop Proceddings* abs/2110.09338 (2021).
[19] Thseen Nazir and Liyana Thabassum. 2021. Cyberbullying: Definition, types, effects, related factors and precautions to be taken during COVID-19 pandemic. *The International Journal of Indian Psychology* (2021).
[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 1135–1144.
[21] Debjoy Saha, Naman Paharia, Debajit Chakraborty, Punyajoy Saha, and Animesh Mukherjee. 2021. Hate-Alert@DravidianLangTech-EACL2021: Ensembling strategies for Transformer-based Offensive language Detection. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics, Kyiv, 270–276. https://aclanthology.org/2021.dravidianlangtech-1.38
[22] Ujwal Narayan Sayar Ghosh Roy, Tathagata Raha, Zubair Abid, and Vasudeva Varma. 2021. Leveraging multilingual transformers for hate speech detection. (2021).
[23] Jonas Paul Schöne, Brian Parkinson, and Amit Goldenberg. 2021. Negativity spreads more than positivity on Twitter after both positive and negative political situations. *Affective Science* 2, 4 (2021), 379–390.
[24] Chander Shekhar, Bhavya Bagla, Kaushal Kumar Maurya, and Maunendra Sankar Desarkar. 2021. Walk in Wild: An Ensemble Approach for Hostility Detection in Hindi Posts. *CoRR* abs/2101.06004 (2021). arXiv:2101.06004 https://arxiv.org/abs/2101.06004
[25] K Sreelakshmi, B Premjith, and KP Soman. 2020. Detection of hate speech text in Hindi-English code-mixed data. *Procedia Computer Science* 171 (2020), 737–744.
[26] Phoey Lee Teh, Chi-Bin Cheng, and Weng Mun Chee. 2018. Identifying and categorising profane words in hate speech. In *Proceedings of the 2nd International Conference on Compute and Data Analysis*. 65–69.
[27] Abhishek Velankar, Hrushikesh Patil, Amol Gore, Shubham Salunke, and Raviraj Joshi. 2021. Hate and Offensive Speech Detection in Hindi and Marathi. *Forum for Information Retrieval Evaluation (FIRE) 2021, CEUR Workshop Proceddings* (2021).
[28] Michael Walsh and Stephanie Baker. 2021. Twitter's design stokes hostility and controversy. Here's why, and how it might change. (2021).
[29] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rJ4km2R5t7
[30] Huiling You, Xingran Zhu, and Sara Stymne. 2021. Uppsala NLP at SemEval-2021 Task 2: Multilingual Language Models for Fine-tuning and Feature Extraction in Word-in-Context Disambiguation. In *Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event / Bangkok, Thailand, August 5-6, 2021*, Alexis Palmer, Nathan Schneider, Natalie Schluter, Guy Emerson, Aurélie Herbelot, and Xiaodan Zhu (Eds.). Association for Computational Linguistics, 150–156. https://doi.org/10.18653/v1/2021.semeval-1.15

8 APPENDICES

A DATA SET CREATION FOR BASELINES

For all the baselines we have followed following terminology - post as (P, Lp), comments as $C = (C_1, L_{C_1}), (C_2, L_{C_2}), \ldots (C_N, L_{C_N})$ and replies as $R = (R_1, L_{R_1}), (R_2, L_{R_2}), \ldots (R_K, L_{R_K})$.

A.1 Simple Concatenation Baseline (SCB)

For a post (P_i) the data sample is

$$(P_i, L_p)$$

For a comment (C_1) it is

$$(P_i + C_1, L_{C_1})$$

For a reply (R_{1_i}) the data sample is

$$(P_i + C_1 + \sum_{1}^{i} R_{1_i}, L_{R_{1_i}})$$

Similarly, for comment (C_2) the text is

$$(P_i + C_1 + \sum_{1}^{i} R_{1_i} + C_2, L_{C_2})$$

,

$$(P_i + C_1 + \sum_{1}^{i} R_{1_i} + C_2 + \sum_{1}^{i} R_{2_i}, L_{R_{2_i}})$$

In general for post (P_k) comment (C_j) and reply (R_{j_i}) the text is

$$(P_k + \sum_{1}^{j} C_j + \sum_{1}^{j} \sum_{1}^{i} R_{j_i}, L_{R_{j_i}})$$

Thus for each instance in the data set there is a concatenated text from the root (i.e. main post corresponding to that instance) with its own ground truth label. For SCB we flatten data set by concatenating as mentioned above. Now, we have a concatenated text and a label for each data sample. The same process is applied on both train and test data sets

A.2 Selective Concatenation Baseline (SLCB)

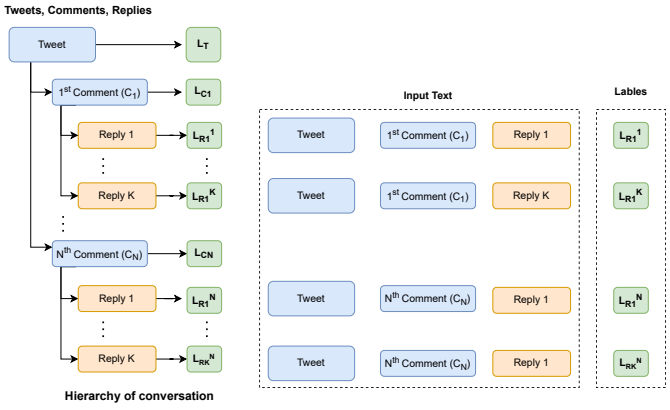

Figure 6: Selective Concatenation Baseline (SLCB) flattening

For a post (P_i) the data sample is

$$(P_i, L_p)$$

For a subsequent comment (C_1) it is

$$(P_i + C_1, L_{C_1})$$

For a subsequent reply (R_{1i}) the data sample is

$$(P_i + C_1 + \sum_{1}^{i} R_{1_i}, L_{R_{1_i}})$$

Similarly, for comment (C_2) the text is

$$(P_i + C_2, L_{C_2})$$

,

$$(P_i + C_2 + \sum_{1}^{i} R_{2_i}, L_{R_{2_i}})$$

In general for post (P_k) comment (C_j) and reply (R_{j_i}) the text is

$$(P_k + C_j + \sum_{1}^{i} R_{j_i}, L_{R_{j_i}})$$

Thus for each post and comment pair in the data set, there is a separate thread as shown in Figure-6. This is done to separate one comment and its subsequent replies from another and handle them separately instead of concatenating all of them together.

A.3 Weighted Context Baseline (WCB)

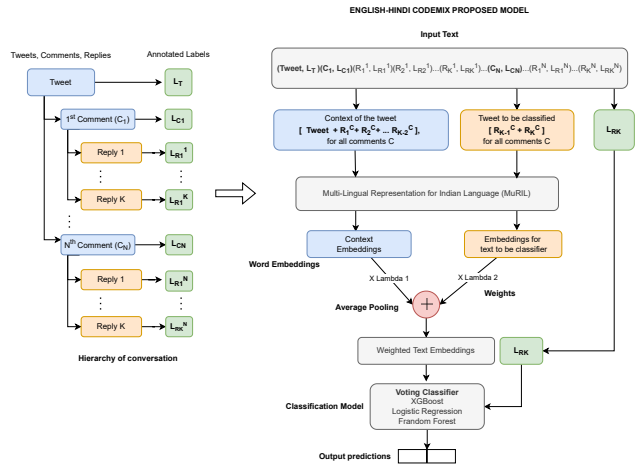

Figure 7: Weighted Context Baseline (WCB) model

For a post (P_i) the data sample is [CLS] representation of

$$(P_i, L_p)$$

For a subsequent comment [CLS] representation of (C_1) it is

$$(P_i + C_1, L_{C_1})$$

For a subsequent reply [CLS] representation of (R_{1_1}) the data sample is

$$(w1 * (P_i + C_1) + w2 * (R_{1_1}), L_{R_{1_1}})$$

Similarly, for reply (R_{1_2}) the text is

$$(w1 * (P_i + C_1) + w2 * (R_{1_1} + R_{1_2}), L_{R_{1_2}})$$

For reply (R_{1_3}) is

$$(w1 * (P_i + C_1 + R_{1_1}) + w2 * (R_{1_2} + R_{1_3}), L_{R_{1_3}})$$

In general for post (P_k) comment (C_j) and reply (R_{j_i}) the text is

$$(w1 * (P_k + C_j + \sum_{1}^{i-2} R_{j_i}) + w2 * (R_{j_{i-1}} + R_{j_i}, L_{R_{j_i}})$$

where w1 and w2 are weights such that $w2 > w1$ to given more preference to recent texts and w1+w2=1.

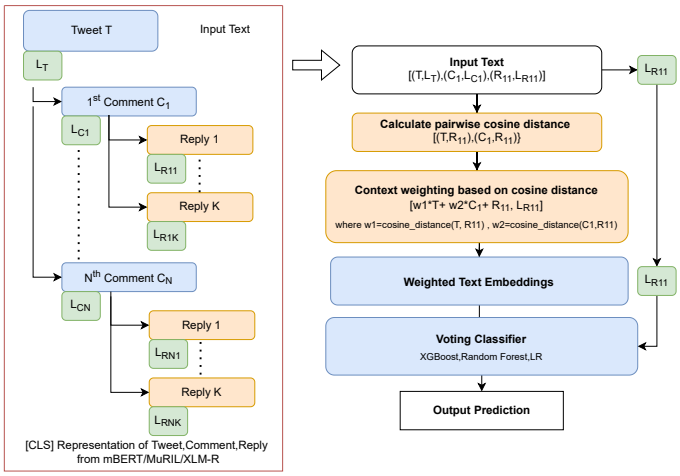

Figure 8: Cosine Attention Based (CAB) model

A.4 Cosine Attentive Baseline (CAB)

To create the data set for this method. We calculate the cosine distance between representation of pair of text within the thread as shown in Figure-8 we assign weights based on distances if the distance is smaller assign more weight. For e.g. For a post (P_i) the data sample is

$$(P_i, L_p)$$

For a subsequent (C_1) it is

$$(P_i + C_1, L_{C_1})$$

For a subsequent reply representation of (R_{11}) the data sample is

$$(w1 * P_i + w2 * C_1 + R_{1_1}, L_{R_{1_1}})$$

Similarly, reply (R_{1_2}) the text is

$$(w1 * P_i + w2 * C_1 + w3 * R_{1_1} + R_{1_2}, L_{R_{1_2}})$$

In general for post (P_k), comment (C_j) and reply (R_{j_i}) the text is

$$(w1 * P_k + w2 * C_j + \sum_{1}^{i-1} w_{i_{+2}} * R_{j_i} + R_{j_i}, L_{R_{j_i}})$$

The weights w1, w2, w3 ..wn corresponds to the cosine distances between the text(post/comment/reply) in the context thread and the text to be classified itself. The label are always taken as same as the reply to be classified for all the models.

A.5 Hindi-English Code-Mixed Model Architecture

The Figure-9 represents the model architecture for hi-en code-mix from [14].

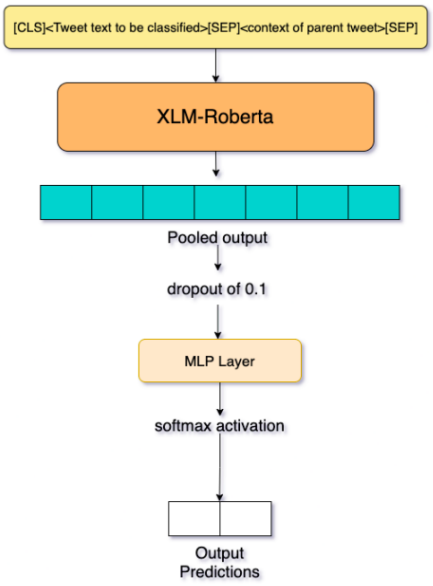

Figure 9: Hindi-English Code-Mix Model

B TIMESTAMP CONSIDERATION FOR TWEETS

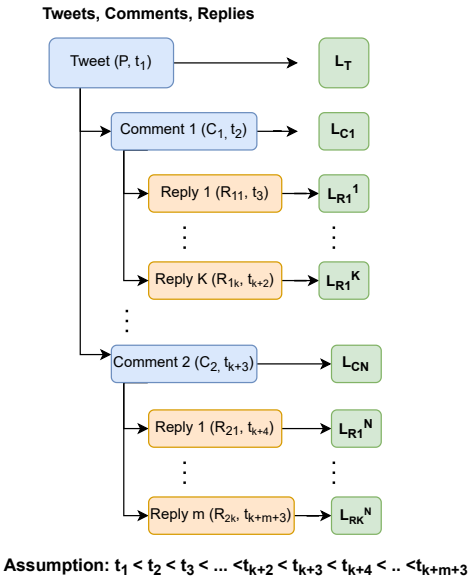

Figure 10: Timestamp order in the tweets

Timestamps of tweets are considered as shown in the Figure-10. All the models (baselines and proposed model) assume the order whenever it is applicable.