Did you offend me? Classification of Offensive Tweets in Hinglish Language

Puneet Mathur
NSIT-Delhi
pmathur3k6@gmail.com

Ramit Sawhney
NSIT-Delhi
ramits.co@nsit.net.in

Meghna Ayyar
IIIT-Delhi
meghnaa@iiitd.ac.in

Rajiv Ratn Shah
IIIT-Delhi
rajivratn@iiitd.ac.in

Abstract

The use of code-switched languages (*e.g.*, Hinglish, which is derived by the blending of Hindi with the English language) is getting much popular on Twitter due to their ease of communication in native languages. However, spelling variations and absence of grammar rules introduce ambiguity and make it difficult to understand the text automatically. This paper presents the Multi-Input Multi-Channel Transfer Learning based model (MIMCT) to detect offensive (hate speech or abusive) Hinglish tweets from the proposed Hinglish Offensive Tweet (HOT) dataset using transfer learning coupled with multiple feature inputs. Specifically, it takes multiple primary word embedding along with secondary extracted features as inputs to train a multi-channel CNN-LSTM architecture that has been pre-trained on English tweets through transfer learning. The proposed MIMCT model outperforms the baseline supervised classification models, transfer learning based CNN and LSTM models to establish itself as the state of the art in the unexplored domain of Hinglish offensive text classification.

1 Introduction

Increasing penetration of social media websites such as Twitter in linguistically distinct demographic regions has led to a blend of natively spoken languages with English, known as code-switched languages. Social media is rife with such offensive content that can be broadly classified as abusive and hate-inducing on the basis of severity and target of the discrimination. *Hate speech (Davidson et al., 2017) is an act of offending a person or a group as a whole on the basis of certain key attributes such as religion, race, sexual orientation, gender, ideological background, mental and physical disability*. On the other hand, *abusive speech is offensive speech with a vague target and mild intention to hurt the sentiments of the receiver*. Most social media platforms delete such offensive content when: (i) either someone reports manually or (ii) an offensive content classifier automatically detects them. However, people often use such code-switched languages to write offensive content on social media so that English trained classifiers can not detect them automatically, necessitating an efficient classifier that can detect offensive content automatically from code-switched languages. In 2015, India ranked fourth on the Social Hostilities Index with an index value of 8.7 out of 10 (Grim and Cooperman, 2014), making it imperative to filter the tremendously high offensive online content in Hinglish.

Hinglish has the following characteristics: (i) it is formed of words spoken in Hindi (Indic) language but written in Roman script instead of the standard Devanagari script, (ii) it is one of the many pronunciations based pseudo languages created natively by social media users for the ease of communication and (iii) it has no fixed grammar rules but rather borrows the grammatical setup from native Hindi and compliments it with Roman script along with a plethora of slurs, slang and phonetic variations due to regional influence. Hence, such code-switched language presents challenging limitations in terms of the randomized spelling variations in explicit words due to a foreign script and compounded ambiguity arising due to the various interpretations of words in different contextual situations. For instance, the sentence: *Main tujhe se pyaar karta hun* is in Hinglish language which means *I love you*. Careful observation highlights how the word *pyaar* meaning 'love' can suffer from phonetic variations due to multiple possible pronunciations such as *pyar*, *pyaar* or *pyara*. Also, the explicit word by word translation of the above sentence, *I you love do*, is grammatically incorrect in English.

Proceedings of the Second Workshop on Abusive Language Online (ALW2), pages 138–148
Brussels, Belgium, October 31, 2018.

We present deep learning techniques that classify the input tweets in Hinglish as: (i) non-offensive, (ii) abusive and (iii) hate-inducing. Since transfer learning can act as an effective strategy to reuse already learned features in learning a specialized task through cross domain knowledge transfer, hate speech classification on a large English corpus can act as source tasks to help in obtaining pre-trained deep learning classifiers for the target task of classifying tweets translated in English from Hinglish language.

Representation vectors constructed by CNN consider local relationship values while the feature vectors constructed by LSTM stress on overall dependencies of the whole sentence. The proposed MIMCT model employs both CNN and LSTM as concurrent information channels that benefit from local as well as overall semantic relationship and is further supported by primary features (multiple word embeddings) and secondary external features (LIWC feature, profanity vector and sentiment score), as described in Section 3.3. The complete MIMCT model is pre-trained on English Offensive Tweet (EOT) dataset, which is an open source dataset of annotated English tweets that was obtained from CrowdFlower[1] and is an abridged version of the original dataset created by Davidson et al. (2017), followed by re-training on the proposed HOT dataset.

The main contributions of our work can be summarized as follows:

- Building an annotated Hinglish Offensive Tweet (HOT) dataset[2].
- We ascertain the usefulness of transfer learning for classifying offensive Hinglish tweets.
- We build a novel MIMCT model that outperforms the baseline models on HOT.

The remainder of this paper is organized as follows. Sections 2 and 3 discuss the related work and methodologies in detail, respectively. Discussions and evaluations are done in Section 4 followed by conclusion and future work in Section 5.

2 Related Work

One of the earliest works on code switched languages was presented by Bhatia and Ritchie (2008) demonstrating cross-linguistic interaction on a semantic level. Several attempts to translate the Hindi-English mixed language into pure English have been made previously, but a major hindrance to this progress has been the fact that the structure of language varies due to relative discrepancies in grammatical features (Bhargava et al., 1988). Ravi and Ravi (2016) proved that a combination of TF-IDF features, gain ratio based feature selection, and Radial Basis Function Neural Network work best for sentiment classification in Hinglish text. Joshi et al. (2016) used sub-word level LSTM models for Hinglish sentiment analysis. Efforts to detect offensive text in online textual content have been undertaken previously for other languages as well like German (Ross et al., 2017) and Arabic (Mubarak et al., 2017).

Gambäck and Sikdar (2017) used a multi-channel HybridCNN architecture to arrive at promising results for hate speech detection in English tweets. Badjatiya et al. (2017) presented a gradient boosted LSTM model with random embeddings to outperform state of the art hate speech detection techniques. Vo et al. (2017) demonstrated the use of multi-channel CNN-LSTM model for Vietnamese sentiment analysis. The use of transfer learning enables the application of feature-based knowledge transference in domains with disparate feature spaces and data distribution (Pan and Yang, 2010). Pan et al. (2012) gave a detailed explanation about the application of transfer learning for cross-domain, instance-based and feature-based text classification. An important work in this direction of Hinglish offensive text classification was done by by Mathur et al. (2018b) by effectively employing transfer learning.

3 Methodology

3.1 Pre-processing

The tweets obtained from data sources were channeled through a pre-processing pipeline with the aim to transform them into semantic feature vectors. The transliteration process was broken into intermediate steps:

Step 1: The first pre-processing step was the removal of punctuations, URLs, user mentions {@mentions} and numbers {0-9}. Hash tags and emoticons were suitably converted by their textual counterparts along with conversion of all tweets into lower case. Stop words corpus obtained from

[1] https://www.crowdflower.com/

[2] The dataset and source code is available for research purposes at www.github.com/pmathur5k10/Hinglish-Offensive-Text-Classification

NLTK was used to eliminate most unproductive words which provide little information about individual tweets. This was followed by transliteration and then translation of each word in Hinglish tweet into the corresponding English word using the Hinglish-English dictionary mentioned in Section 4.1. At this step, the syntax and grammatical notions of the target language were ignored and the resultant tweet was treated as an assortment of isolated words and phrases to make them eligible for conversion into word vector representation.

Step 2: We used multiple word embedding representations such as Glove (Pennington et al., 2014), Twitter word2vec (Godin et al., 2015), and FastText (Bojanowski et al., 2016) embeddings for creating the word embedding layers and to obtain the word sequence vector representations of the processed tweets. Finally, the train-test split of both the datasets was been kept in the ratio of 80:20 for all experiments described in this paper.

3.2 Transfer Learning based Offensive Text Classification

Recently, Badjatiya et al. (2017) performed state of the art classification of tweets in English language as racist, sexist or neither using multiple deep learning techniques motivating exploration of similar models for our task. The problem of hate speech classification in Hinglish language is similar to that in English due to the semantic parallelism but suffers from the drawback of syntactic disassociation when Hinglish is translated into English. The proposal to apply transfer learning is inspired by the fact that despite having a small-sized dataset, it provides relative performance increase at a reduced storage and computational cost (Bengio, 2012). Deep learning models pre-trained on EOT learn the low-level features of the English language tweets. The weights of initial convolutional layers are frozen while the last few layers are kept trainable such that when the model is retrained on the HOT dataset, it learns to extract high level features corresponding to syntax variations in translated Hinglish language.

One major drawback of CNN models is the fact that it finds only the local optimum in weighted layers. This disadvantage is somewhat overcome by LSTM's since they are well-suited to classify, process and capture long term dependencies in text. This makes them an excellent choice to learn long-range dependencies from higher-order sequential features. The aim of three-label offensive tweet classification is achieved by using both CNN and LSTM models, respectively. In the first stage of experiments, the respective models are trained and tested on HOT to serve as a benchmark. The same models are reinitialized and run from scratch on the EOT dataset followed by retraining on the HOT dataset by keeping only the last dense layers as trainable. The models are finally then tested on the testing section of HOT and results compiled in Table 7. We hypothesize that the performance of both CNN and LSTM should comparatively enhance due to transfer learning as compared to the benchmark due to syntactical degradation of tweets during the pre-processing step. If this process leads to an overall enhancement of model performance on HOT dataset, then the intuition to use transfer learning for transferring pre-learnt semantic features between two syntactically obscure language would hold ground. As per (Park and Fung, 2017), proposed CNN and LSTM architecture for these experiments were designed to have shallow layers as the small size of our dataset runs the risk of overfitting on the data.

3.3 MIMCT Model

The architecture of the MIMCT model is shown in Figure 1, consisting of two main components: (i) primary and secondary inputs and (ii) CNN-LSTM binary channel neural network. The following subsection describes the application of primary and secondary inputs in MIMCT.

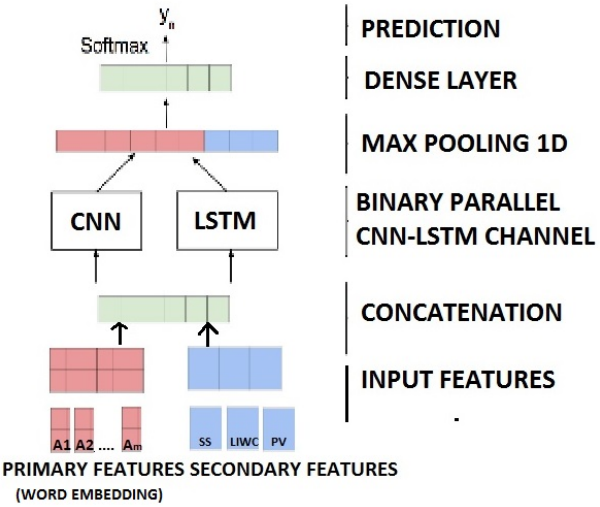

Figure 1: The MIMCT model

3.3.1 Primary and Secondary Inputs

Word embeddings help to learn distributed low-dimensional representations of tweets and differ-

ent embeddings induced using different models, corpora, and processing steps encode different aspects of the language. While bag of words statistics based embeddings stress on word associations (doctor-hospital), those based on dependency-parses focus on similarity in terms of use (doctor-surgeon). Inspired by the works of (Mahata et al., 2018b), it is natural to consider how these embeddings might be combined to obtain the set of most promising word embeddings amongst Glove, Twitter Word2vec and FastText. Assuming m word embeddings with corresponding dimensions d_1, d_2, ...d_m are independently fed into the MIMCT model as primary inputs. Thus the input to MIMCT will comprise of multiple sentence matrices A_1, A_2, ...A_m, where each $A_l \in R^{s*d_l}$ having s as zero-padded sentence length and d_l as dimensionality of the embedding. Keeping the embedding dimension constant to 200 in each case, we obtained independent feature vectors for each set of embeddings that are known as primary inputs. Apart from the regular embedding inputs, additional hierarchical contextual features are also required so as to complement the overall classification of the textual data. These features additionally focus on the sentiment and tailor-made abuses that may not be present in regular dictionary corpus. This helps to overcome a serious bottleneck in the classification task and could be one of the prominent reasons for high misclassification of abusive and hate-inducing class in baseline and basic transfer learning approaches. The multiple modalities added to the MIMCT model as secondary inputs are:

- **Sentiment Score (SS)**: We have used tweet sentiment score evaluated using SentiWordNet (Baccianella et al., 2010) as a feature to stress on polarity of the tweets. The SS input will be a unidimensional vector denoted by +1 for positive, 0 for neutral and -1 for negative sentiment.

- **LIWC Features**: Inspired by (Sawhney et al., 2018a), Linguistic Inquiry and Word Count (Pennebaker et al., 2007) throws light on various language modalities expressing the linguistic statistical make-up of each text. Table 1 portrays the cumulative linguistic attributes calculated by LIWC2007 to form a LIWC attribute vector of 67 dimension (67D). Moreover, we have excluded numbers and punctuation in LIWC features as these are removed in pre-processing steps.

- **Profanity Vector**: Swearing is a form of expressing emotions, especially anger and frustration (Jay and Janschewitz, 2008). Section 4.1 describes the Hinglish Profanity list with corresponding English translation. An integer vector of dimension 210 (210D) is constructed for each tweet such that the presence of a particular bad word is demarcated by its corresponding profanity score while its absence is demarcated by null value to emphasize the presence of contextually subjective swear words.

4 Evaluation

We provide an extensive description of the sources, ground truth annotation scheme and statistics of the proposed Hinglish Offensive Tweets (HOT) dataset in Section 4.1. Next, we discuss implementation details of baseline, transfer learning and MIMCT model in Section 4.2 followed by results analysis in Section 4.3.

4.1 Dataset

Table 2 spells out tweet distributions across EOT and HOT datasets. HOT is a manually annotated dataset that was created using the Twitter Streaming API[3] by selecting tweets having more than 3 Hinglish words. The tweets were collected during the interval of 4 months of November 2017 to February 2018. The tweets were mined by imposing geo-location restriction such that tweets originating only in the Indian subcontinent were made part of the corpus. Inspired by the work of Rudra et al. (2016), tweets were mined from popular Twitter hashtags of viral topics popular across the news feed. Bali et al. (2014) pointed out that Indian social media users have high activity on Facebook pages of a few listed prominent public entities. Hence, we crawled tweets and responses from Twitter handles of sports-persons, political figures, news channels and movie stars. The collected corpus of tweets initially had 25667 tweets which was filtered down to remove tweets containing only URL's, only images and videos, having less than 3 words, non-English and non-Hinglish scripts and duplicates. The annotation of

[3] https://developer.twitter.com/

LIWC Categories	Attributes
Linguistic Statistics	word count (mean), words per sentence, dictionary words, total pronouns, words >6 letters, total functional words, personal pronouns
Grammatical Structures	1st person singular, 1st person plural,2nd person, 3rd person singular, 3rd person plural, impersonal pronouns, articles, common verbs, auxiliary verbs, past, present and future tense, adverbs, prepositions, conjunctions, negotiations, quantifiers, swear words
Textual Category	sexual, body, health, ingestion, relativity, motion, space, time,
Psychological Processes	social processes, family, friends, humans, effective processes, negative and positive emotions, anxiety, anger, sadness, cognitive processes, insight, causation, discrepancy, tentativeness, certainty, inhibition, inclusive, exclusive, perceptual processes, see, hear feel, biological processes,
Current Concerns	work, achievement, leisure, home, money, religion, death
Spoken Categories	assent, non-fluencies, fillers

Table 1: LIWC linguistic attributes used in the MIMCT model

Label	EOT	HOT
Non-Offensive	7274	1121
Abusive	4836	1765
Hate-inducing	2399	303
Total	**14509**	**3189**
Train	11608	2551
Test	2944	637

Table 2: Tweet distributions in EOT and HOT.

Tweet	Label
(i) Tum ussey pyar kyun nahin karti? (ii) Why don't you love him? (iii) you him love why no	Non-offensive
(i) Ch*d! Yeh sab ch*tiye hain! :/ (ii) F**k! They all are c*nts! :/ (iii) F**k they all c*nts are	Abusive
(i) M*d*rch*d Mus*lm**n sE nafrat (ii) m*therf*ck*r m*sl*m hate (iii) m*therf*ck*r m*s*l*m**n hate	Hate-inducing

Table 3: Examples of tweets in the HOT dataset. Categories (i), (ii) and (iii) denote the Hinglish tweet, its corresponding English meaning and its transliterated and translated version. The authors have modified some bad words in original tweets with '*' to not offend the readers.

HOT tweets were done by three annotators having sufficient background in NLP research. The tweets were labeled as hate speech if they satisfied one or more of the conditions: (i) tweet used sexist or racial slur to target a minority, (ii) undignified stereotyping or (iii) supporting a problematic hashtags such as #ReligiousSc*m. The label chosen by at least two out of three independent annotators was taken as final ground truth for each tweet. In case of conflict amongst the annotators, an NLP expert would finally assign the ground truth annotation for ambiguous tweets. In this way, 386 tweets needed expert annotation, while 2803 tweets were labeled through consensus of annotators with an average value of Cohen Kappa's inter-annotator agreement $\kappa = 0.83$. Table 5 shows the internal agreement between our annotators.

A curated list of profane words was extracted to form the Hinglish profanity list[4], which was created by accumulating Hinglish swear words from curated social media blog posts (Rizwan, 2016) and dedicated swear word forums[5]. Each swear word was assigned an integer score on the scale of (1-10) based on the degree of profanity. This assignment of profanity scores was accomplished through discussion amongst four independent code-switching linguistic experts having an extensive background in social media analysis.

The task of transliterating Hinglish words into Devanagari Hindi was achieved using datasets provided by Khapra et al. (2014). The words so obtained were further translated into Roman script using a Hindi-English dictionary consisting of 136110 word pairs mined from CFILT, IIT Bombay[6]. Additionally, the English translations present of the words in Hinglish Profanity list were added to form a map based Hinglish-English dictionary. A pertinent challenge in dealing with Hinglish language was the presence of spelling variants, homophones and homonyms that are used frequently in a loose context. Thus the spelling variations of various popular Hinglish words were added to the corpus. The Hinglish-English dictionary thus formed, comprising of 7193 word pairs, was used as the basis for all further Hinglish to English tweet conversions. Table 4 gives detailed examples of word pairs in Hinglish-English dictionary along with a few swear words and their profanity scores.

Approximately, 29% of the tokens in pre-processed tweets are in Hinglish, a whopping 65% of the tokens are in English, while the remaining are Hinglish named entities like persons, events, organizations or places. The higher instances of

[4] www.github.com/pmathur5k10/Hinglish-Offensive-Text-Classification

[5] http://www.hindilearner.com/hindi_words_phrases/hindi_bad_words1.php

[6] http://www.cfilt.iitb.ac.in/~hdict/webinterface_user/

Hinglish	English	Tag
acha	good	adjective
neeche	under	adposition
abhi	now	adverb
aur	and	conjunction
nahin	no	determiner
ghar	home	noun
ek	one	numeral
saath	with	particle
tum	you	pronoun
pyar	love	verb
*s**la*	blo*dy	swear (1)
*k*tta*	dog	swear (2)
*s**ver*	pig	swear (3)
*har**mi*	bast*rd	swear (4)
*ch*tiya*	f*cker	swear (5)
*bh*dve*	p*mp	swear (6)
*g**nd*	a*s	swear (7)
*r*ndi*	ho*oker	swear (8)
*b*h*nch*d*	s*sterf*ck*r	swear (8)
*m*d*rch*d*	m*therf*ck*r	swear (10)

Table 4: Examples of word pairs in Hinglish-English dictionary and Hinglish Profanity List with their profanity score

the named-entities in the HOT dataset is a result of the way the data is sourced. Around 1.4% of Hinglish words in HOT share the same spellings with some English words because of transliteration of Hindi text to Roman script. The t-SNE (Maaten and Hinton, 2008) plot of the HOT dataset shows the probability distribution of words in terms of the tokens used in tweets as represented by Figure 2. We also computed a few metrics to understand code-switching patterns in our dataset, so as to rationalize the performance of the classification models.

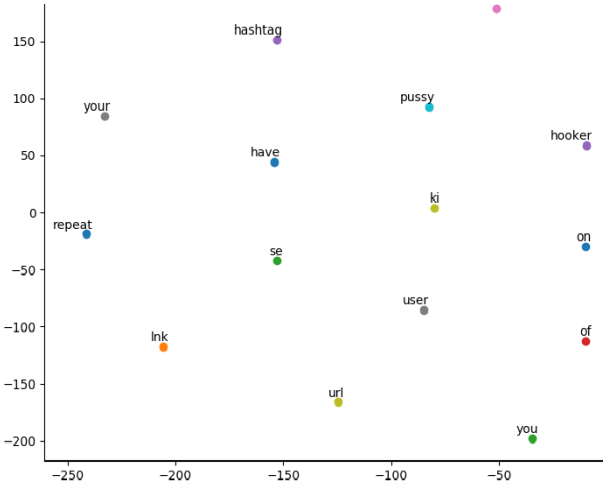

Figure 2: T-SNE plot of the HOT dataset

Multilingual Index (M_i): It is a word-count-based measure that quantifies the inequality of the language tags distribution in a corpus of at least two languages (Barnett et al., 2000). Let k be the total number of languages and p_j is the total number of words in the language j over the total number of words in the corpus. The value of M_i ranges between 0 and 1 where, a value of 0 corresponds to a monolingual corpus and 1 corresponds to a corpus with equal number of tokens from each language. Equation 1 depicts the M_i which is approximately equal to 0.601, indicating that a majority of words are in Hinglish.

	A_1	A_2	A_3
A_1	–	0.76	0.84
A_2	0.76	–	0.88
A_3	0.84	0.88	–

Table 5: Cohen's Kappa for three annotators A_1, A_2 and A_3

Integration Index (I_i): Integration Index is the approximate probability that any given token in the corpus is a switch point (Guzmán et al., 2017). This metric quantifies the frequency of code-switching in a corpus. Given a corpus composed of tokens tagged by language $\{l_j\}$, i ranges from 1 to $n-1$, where n the size of the corpus. S(l_i, l_j) = 1 if $l_i = l_j$ and 0 otherwise in Equation 2. The value of I_i computed is approximateedly 0.079 portraying a high frequency of code-switching points.

$$M_i = \frac{1 - \sum {p^2}_j}{(k-1)\sum {p^2}_j} \quad (1)$$

$$I_i = \frac{1}{n-1} \sum_{1<=i<j-1<=n-1} S(l_i, l_j) \quad (2)$$

4.2 Implementation Details

4.2.1 Baseline

Several baseline models were experimented such as Support Vector Machine (SVM) and Random Forests (RF). The supervised models were trained using *k*-fold cross-validation with 10 splits (k=10) each. The hyper-parameters for Random Forest classifier were fine tuned and the results were found to be optimal when n_estimators, max_depth and max_features were fixed at 1000, 15 and log2, respectively. Other parameters for the SVM classifiers were initialized to default values. Inspired by Badjatiya et al. (2017) and Mathur et al. (2018a), various features were extracted from pre-processed tweets to be used as input to the baseline models such as (i) Character n-grams, (ii) Bag of

Words Vector (BoWV) and (iii) TF-IDF count vector and the results have been summarized in Table 6.

4.2.2 Transfer Learning

The number of trainable and static layers were toyed with to get the best combination giving suitable results. For the classification task, both CNN and LSTM models are trained using 10-fold cross validation to identify the best hyper-parameter settings as presented below:

- **CNN**: Convolutional 1D layer (filter size=15, kernel size=3) → Convolutional 1D (filter size=12, kernel size=3) → Convolutional 1D (filter size=10, kernel size=3) → Dropout (0.2) → Flatten Layer → Dense Layer (64 units, activation = 'relu') → Dense Layer (3 units, activation = 'softmax')

- **LSTM** : LSTM layer(h=64, dropout=0.25, recurrent dropout=0.3) → Dense (64 units, activation = 'relu') → Dense (3 units, activation = 'sigmoid')

The final layer of both CNN and LSTM models is the compile layer with categorical cross-entropy as the loss function, *Adam* as the optimizer, learning rate kept at 0.001 and L2 regularization with strength of 1E-6. The CNN and LSTM models were tested using three flavors of word embeddings : (i) Glove, (ii) Twitter word2vec and (iii) FastText separately. The dimensions of input word embeddings were kept constant at 200 as for consistency across all embeddings. The hyper parameters were chosen by grid search by running the experiments over a wide range. The batch size was experimented from size 8 to 128. Similarly, the number of epochs were limited at point were the model training loss plateaued by exploring different values from 10 to 50 in intervals of 5. The epochs and batch size were fixed to 20 and 64 respectively so as to maintain consistency in performance evaluation in each case without compromising on the optimality of the results corresponding to each configuration as summarized in Table 7.

4.2.3 MIMCT Model

Distinct word embedding representation are generated from each participant embedding layer that are concatenated along with secondary features and fed to the MIMCT model as independent inputs to both CNN and LSTM channels. The features after passing through both the channels are merged and passed to the Max-pooling 1D layer. The resultant vector is reshaped and fed into a final softmax layer to perform tertiary classification. The architecture of CNN channel comprises of three successive Convolutional-1D layers with filter size chosen as 20, 15 and 10 respectively. This is followed by a dropout layer of value 0.25 and flatten layer. This is immediately followed by a single dense layer of 3 units with *softmax* activation. The LSTM channel is simply a layered structure comprising of a LSTM layer (128 units and dropout value of 0.2) and a dense layer (3 units and *softmax* activation). The MIMCT model uses *Adam* optimizer (Kingma and Ba, 2014) along with L2 regularization to prevent overfitting in the model. MIMCT was initially trained on the EOT dataset and the complete model is re-trained on the HOT dataset so as to benefit from the transfer of learnt features in the last stage. The model hyper-parameters were experimentally selected by trying out a large number of combinations through grid search.

Feature	Char N-grams		BoWV		TF-IDF	
Classifier	SVM	RF	SVM	RF	SVM	RF
Precision	0.679	0.565	0.688	0.579	**0.721**	0.655
Recall	0.708	0.587	**0.731**	0.664	0.724	0.678
F1-Score	0.688	0.574	0.703	0.639	**0.723**	0.666

Table 6: Baseline results for non-offensive, abusive, hate-inducing tweet classification on HOT

4.3 Results and Discussion

Table 6 clearly show that SVM model supplemented with TF-IDF features gives peak performance in terms of F1-score and precision when compared to other configurations of baseline supervised classifiers. The general inference that can be drawn at this stage is that the SVM classifier outperforms Random Forest. Another useful observation is that TF-IDF is the most effective feature for semantically representing Hinglish text and gives better performance than both Bag of Words Vector and Character N-grams on respective classifiers. These observations are in agreement with the results presented by Badjatiya et al. (2017) who also used supervised classification for offensive tweet classification in English.

Table 7 shows results (in terms of F1-score, precision, and recall) for the classification task

Embedding	Glove						Twitter Word2vec						FastText					
Model	CNN			LSTM			CNN			LSTM			CNN			LSTM		
Data	EOT	HOT	TFL	EOT	HOT	TFL	EOT	HOT	TFL	EOT	HOT	TFL	EOT	HOT	TFL	EOT	HOT	TFL
Precision	0.843	0.734	0.789	0.819	0.753	0.802	0.856	0.762	0.793	0.821	0.756	0.810	0.800	0.730	0.758	0.799	0.746	**0.823**
Recall	0.841	0.804	0.820	0.834	0.764	0.819	0.861	0.811	0.817	0.835	0.779	**0.846**	0.820	0.805	0.827	0.807	0.677	0.838
F1-Score	0.841	0.755	0.801	0.816	0.752	0.813	0.857	0.799	0.815	0.835	0.765	**0.830**	0.811	0.772	0.793	0.800	0.755	0.823

Table 7: Results for non-offensive, abusive, hate-inducing tweet classification on EOT, HOT and the HOT dataset with transfer learning (TFL) for Glove, Twitter Word2vec and FastText embeddings

of transfer learning on CNN and LSTM models. Macro metrics are preferred in experimentation evaluation as the class imbalance is not severe enough to skew the outcomes. Training and testing the same models from scratch on HOT without transfer learning reports a sharp downfall in performance of CNN and LSTM, which is quite expected as the Hinglish tweets in the HOT dataset suffer from syntactic degradation after transliteration and translation. But following the transfer learning methodology, the model performances on HOT improve significantly strengthening the argument that there was a positive transfer of features from English to Hinglish tweet data. A relative comparison between several configurations of CNN and LSTM with corresponding word embeddings reflects that LSTM's are slightly better in each case relative to the corresponding CNN models and the Twitter word2vec outperforms its contemporary embeddings in most cases.

Lastly, the observation of the MIMCT model gives us useful insights to examine the effects of using multiple inputs. While the combination of Twitter word2vec (Tw) and FastText (Ft) shows superior performance than other embedding combinations, the addition of sentiment score has little affect on the overall classification performance. In contrast, the usage of profanity vector and LIWC features boosts the metric values and the best classifier performance is recorded when all the secondary features are used together in conjugation with Twitter word2vec and FastText embeddings. MIMCT shows significant performance improvement over the baselines presented in our work to emerge as the current state of the art in the task of Hinglish offensive tweet detection. MIMCT model (Tw + Ft + SS + PV + LIWC) out performs SVM supplemented with TF-IDF features and the Twitter-LSTM transfer learning model by 0.166 and 0.165 F1 points, respectively.

4.4 Error Analysis

Some categories of error that occur in MIMCT:

MIMCT Features	**Precision**	**Recall**	**F1**
Glove (Gl)	0.819	0.849	0.805
Twitter Word2vec (Tw)	0.867	0.810	0.852
FastText (Ft)	0.860	0.777	0.831
(Gl) + (Tw)	0.859	0.745	0.844
(Tw) + (Ft)	**0.861**	**0.854**	**0.857**
(Gl) + (Ft)	0.800	0.850	0.812
(Gl) + (Tw) + (Ft)	0.819	0.795	0.804
(Tw) + (Ft) + [SS]	0.782	0.902	0.858
(Tw) + (Ft) + [LIWC]	0.793	0.925	0.885
(Tw) + (Ft) + [PV]	0.618	0.890	0.888
(Tw) + (Ft) + [SS + PV]	0.759	0.904	0.886
(Tw) + (Ft) + [SS + LIWC]	0.732	0.865	0.889
(Tw) + (Ft) + [PV + LIWC]	**0.851**	0.905	0.893
(Tw) + (Ft) + [SS + PV + LIWC]	0.816	**0.928**	**0.895**

Table 8: Results of the MIMCT model with various input features HOT compared to previous baseline. Primary inputs are enclosed within parentheses, e.g., (Tw), and secondary inputs are enclosed within square brackets, e.g. [LIWC].

1. **Creative word morphing**: Human annotators as well as the classifier misidentified the tweet *'chal bhaag m*mdi'*, which translates in English as *'go run m*mdi'*, as non-offensive instead of hate-inducing. Here *'m*mdi'* is an indigenous way of referring to a particular minority that has been morphed to escape possible identification.

2. **Indirect hate**: The tweet *'Bas kar ch*tiye m***rsa educated'* was correctly identified by our annotators as hate-inducing but the classifier identified it as abusive. This is because pre-processing of this tweet as *'Limit it m*ther f*cking religious school educated'* leads to lose in its contextual reference to customs and traditions of a particular community.

3. **Uncommon Hinglish words**: The work in its present form dos not deal with uncommon and unknown Hinglish words. These may arise due to spelling variations, homonyms, grammatical incorrectness, mixing of foreign language, influence of regional dialect or negligence due to subjective nature of the

transliteration process.

4. **Analysis of code-mixed words**: It has been shown in previous research (Singh, 1985) that bilingual languages tend to be biased in favour of code-mixing of certain words at specific locations in text. Contextual investigation in this direction can be a useful to eliminate the subjective problem of Hinglish to English transliteration in future work.

5. **Possible overfitting on homogenous data**: The data usually present on the social media portals tend to be noisy and often repetetive in content. The skew in the class balance of dataset coupled with training on deep layered model may lead to overfitting of the data and may possibly induce large variation between expected and real-world results. We suspect this might be inherent in present experiments and can be overcome by extracting data from heterogenous sources to model a real-life scenario.

5 Conclusion and Future Work

We introduced a novel HOT dataset for multi-class labeling of offensive textual tweets in Hindi-English code switched language. The tweets in Hinglish language are transformed into semantically analogous English text followed by experimental validation of transfer learning for classifying cross-linguistic tweets. We propose the MIMCT model that uses multiple embeddings and secondary semantic features in a CNN-LSTM parallel channel architecture to outperform the baselines and naive transfer learning models. Finally, a brief analysis of the HOT dataset and its associated errors in classification has been provided. Possible future enhancements include applying feature selection methods to choose the most prominent features amongst those presented similar to the work done by (Sawhney et al., 2018b,c), extending MIMCT to other code-switched and code-mixed languages and exploring GRU-based models. Also, stacked ensemble of shallow convolutional neural networks can be explored for Twitter data as shown by Mahata et al. (2018a).

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. " i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Ruthanna Barnett, Eva Codó, Eva Eppler, Montse Forcadell, Penelope Gardner-Chloros, Roeland van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, Mark Sebba, et al. 2000. The lides coding manual: A document for preparing and analyzing language interaction data version 1.1–july 1999. *International Journal of Bilingualism*, 4(2):131–271.

Yoshua Bengio. 2012. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36.

Hemant K Bhargava, Michael Bieber, and Steven O Kimbrough. 1988. Oona, max and the wywwywi principle: generalized hypertext and model management in a symbolic programming environment. In *ICIS*, page 23.

Tej K Bhatia and William C Ritchie. 2008. The bilingual mind and linguistic creativity. *Journal of Creative Communications*, 3(1):5–21.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. corr abs/1607.04606.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153.

Brian J Grim and Alan Cooperman. 2014. Religious hostilities reach six-year high. *Pew Research Center, January*, 14.

Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. *Proc. Interspeech 2017*, pages 67–71.

Timothy Jay and Kristin Janschewitz. 2008. The pragmatics of swearing. *Journal of Politeness Research. Language, Behaviour, Culture*, 4(2):267–288.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491.

Mitesh M Khapra, Ananthakrishnan Ramanathan, Anoop Kunchukuttan, Karthik Visweswariah, and Pushpak Bhattacharyya. 2014. When transliteration met crowdsourcing: An empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control. In *LREC*, pages 196–202.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Debanjan Mahata, Jasper Friedrichs, Rajiv Ratn Shah, and Jing Jiang. 2018a. Did you take the pill?-detecting personal intake of medicine from twitter. *arXiv preprint arXiv:1808.02082*.

Debanjan Mahata, Jasper Friedrichs, Rajiv Ratn Shah, et al. 2018b. # phramacovigilance-exploring deep learning techniques for identifying mentions of medication intake from twitter. *arXiv preprint arXiv:1805.06375*.

Puneet Mathur, Meghna Ayyar, Sahil Chopra, Simra Shahid, Laiba Mehnaz, and Rajiv Shah. 2018a. Identification of emergency blood donation request on twitter. In *Proceedings of the Third Workshop On Social Media Mining for Health Applications*.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018b. Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Weike Pan, Erheng Zhong, and Qiang Yang. 2012. Transfer learning for text mining. In *Mining Text Data*, pages 223–257. Springer.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*.

James W Pennebaker, Roger J Booth, and Martha E Francis. 2007. Liwc2007: Linguistic inquiry and word count. *Austin, Texas: liwc. net*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Kumar Ravi and Vadlamani Ravi. 2016. Sentiment classification of hinglish text. In *Recent Advances in Information Technology (RAIT), 2016 3rd International Conference on*, pages 641–645. IEEE.

Sahil Rizwan. 2016. This Reddit Thread On The Best Indian Gaalis Will Increase Your Vocabulary, If Nothing Else. https://www.buzzfeed.com/sahilrizwan/speak-no-evil?utm_term=.kgPxblYyl#.suP6XO4VO. [Online; accessed 19-May-2018].

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 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*, pages 1131–1141.

Ramit Sawhney, Prachi Manchanda, Raj Singh, and Swati Aggarwal. 2018a. A computational approach to feature extraction for identification of suicidal ideation in tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 91–98.

Ramit Sawhney, Puneet Mathur, and Ravi Shankar. 2018b. A firefly algorithm based wrapper-penalty feature selection method for cancer diagnosis. In *International Conference on Computational Science and Its Applications*, pages 438–449. Springer.

Ramit Sawhney, Ravi Shankar, and Roopal Jain. 2018c. A comparative study of transfer functions in binary evolutionary algorithms for single objective optimization. In *International Symposium on Distributed Computing and Artificial Intelligence*, pages 27–35. Springer.

Rajendra Singh. 1985. Grammatical constraints on code-mixing: evidence from hindi-english. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 30(1):33–45.

Quan-Hoang Vo, Huy-Tien Nguyen, Bac Le, and Minh-Le Nguyen. 2017. Multi-channel lstm-cnn model for vietnamese sentiment analysis. In *Knowledge and Systems Engineering (KSE), 2017 9th International Conference on*, pages 24–29. IEEE.