Hate or Non-hate: Translation based hate speech identification in Code-Mixed Hinglish data set

Conference Paper · December 2021

DOI: 10.1109/BigData52589.2021.9671526

3 authors, including:

Shankar Biradar
Indian Institute of Information Technology Dharwad
10 PUBLICATIONS **26** CITATIONS
SEE PROFILE

Sunil Saumya
Indian Institute of Information Technology Dharwad
32 PUBLICATIONS **658** CITATIONS
SEE PROFILE

Hate or Non-hate: Translation based hate speech identification in Code-Mixed Hinglish data set

Shankar Biradar
Dept of Computer Science and Engineering
Indian Institute of Information Technology Dharwad
Dharwad, India
shankar@iiitdwd.ac.in

Sunil Saumya
Dept of Computer Science and Engineering
Indian Institute of Information Technology Dharwad
Dharwad, India
sunil.saumya@iiitdwd.ac.in

Arun Chauhan
Dept of Computer Science and Engineering
Graphic Era University
Dehradun, India
aruntakhur@gmail.com

***Abstract*—Hate speech identification in social media has emerged as a highly debated research topic in computational linguistics. Understanding linguistic phenomena in low-resource languages, in particular, remains a major problem in natural language processing. Code-mixing is a common phenomenon in social media writing, particularly in multilingual societies such as India. Traditional deep learning techniques trained on monolingual data will not perform well on code-mixed data, and training new models are challenging due to a lack of resources. Converting multilingual data into monolingual is an important solution to this challenge. TIF-DNN, a Transformer-based Interpretation and Feature Extraction Model is proposed in this work for hate speech identification. We used the IndicNLP and Englishtohindi libraries for transliteration and translation, respectively, and mBERT for feature extraction in our suggested model. Later, we compared our findings to various baseline and existing models.**

***Index Terms*—hate speech, deep learning, mBERT, Transformer.**

I. Introduction

The advent of social media helped to bridge the gap between borders and paved the way for people to communicate or express their opinions more easily than at any other time in human history (Edosomwan et al., 2011). Nowadays, social media has become inevitable part of our culture. Many of us rely on social media posts to get information; we believe whatever we see on social media without verifying the facts, and information on social media spreads so quickly that it reaches all inhabitants in a very short time. But, sadly, there is a dark side to it: because social media allows people to create and publish content, it allows people to spread propaganda through these mediums without being scrutinized. Hate news is one such propaganda that target, defame and marginalize people or group of people or particular community merely on the basis of their physical appearance, religion or sexual orientation (Pamungkas et al., 2020)(Chowdhury et al., 2020). It influences users by manipulating them for economic, religious, and political reasons, ultimately harming society as a whole.

Due to the growth of internet usage and a lack of media policy for verification, containing the spread of hate news in a multi-cultural and multi-lingual society like India is extremely difficult. India has become perfect test bed for spreading hate full content related to religion, language and politics. Some recent examples include hate mongering in social media reaches to its peak during recent West Bengal, India elections resulted in clashes between the workers of two parties' leads to death of some innocent people[1]. More recently, north-east, India inhabitants have suffered racial discrimination during the Covid-19 rise as a result of hateful propaganda published on social media against those individuals[2]. These facts have emphasized the need of preventing the transmission of hate news on social media, which is gaining attention among researchers and academics.

The majority of previous research on hate speech detection has focused on high resource languages such as English (Khan et al., 2020; Mossie and Wang, 2020; Senarath and Purohit, 2020). India, the world's second most populous country, also has the second largest number of English speakers, with Hindi being the native language of 41% of the people[3]. As a result, while conversing, people tend to mix their native language with English, and identifying hateful content in code-mixed Hindi-English(Hinglish) has gained the importance in recent times. Code-mixing (CM) is a linguistic phenomenon in which humans mix two or more languages in a single utterance. In CM, words and phrases from different languages are combined (Srivastava and Singh, 2021). Following are the some of the instances of code-mixed texts.

T1: "neeraj ka nam humesha yaad rahega because he won first gold medal for India in athletics!!!..".

[1] https://www.newindianexpress.com/nation/2020/nov/16/decoding-political-killings-2223960.html

[2] https://www.thehindu.com/news/national/other-states/northeast-citizens-faced-racial-discrimination-amid-covid-19-outbreak-says-govt-study/article34303162.ece

[3] https://en.wikipedia.org

T2: "muje apane manager se bahut nafarat hai, I want to kill him".

From the above instances T2 contain hate speech, while T1 is an instance of normal speech.

Several off-the-shelf tools have recently been developed to handle Indian languages, such as IndicNLP(Kunchukuttan et al., 2020), iNLTK(Arora, 2020), and stanza(Qi et al., 2020), however these are monolingual toolkits that cannot handle code-mixed data. The usage of shorter forms of words, spelling changes, and non-grammatical phrase structure make finding hate speech in code-mixed data more difficult (Santosh and Aravind, 2019). Furthermore, it is more challenging since it is a low-resource language with weakly defined semantic and grammatical norms (Chopra et al., 2020). Because of these challenges code-mixed social media text hasn't been explored much. Hence to explore the challenges of code-mixed scenarios, in this paper, we have proposed Transformer based Interpreter and Feature extraction model on Deep Neural Network (TIF-DNN) as explained in section III.

Main contribution to the paper includes:

1) TIF-DNN, a Transformer based Interpreter and Feature extraction model on Deep Neural Network for hate speech identification in code-mixed Hinglish language has been developed.
2) Efficiency of the proposed model is demonstrated by comparing proposed method with existing ones.

The remainder of the article is structured as follows: Section II gives a brief summary of the background literature. Section III contains specifics on the suggested approach and data set. Section IV discusses the experiment results as well as the experimental setup.

II. LITREATURE REVIEW

Most prior work on sentiment analysis has been done primarily on high resource languages such as English. However, code-mixed languages have received little attention due to their non-standard writing style and a shortage of data sets to train the models. As a result, researchers have just lately begun to investigate code-mixed data. The following are some of the approaches used to handle with CM data.

A. USING HANDCRAFTED LINGUISTIC FEATURES

The first such attempt was performed by (Bohra et al., 2018), they provided an annotated corpus of Hindi-English code-mixed text, comprising tweet ids and the accompanying annotations. They also demonstrated the supervised method for detecting hate speech in code-mixed text. They employed character n-grams, word n-grams, punctuation's, negation words, and hate lexicons as classification features. (Samghabadi et al., 2018) investigated statistical features such as char n-gram, word uni-gram, and word bi-gram with logistic regression and multinomial naive bayes and discovered that a combination of character n-gram where n varies from 2 to 4 along with word uni-gram when classified using logistic regression gave comparatively good results on Hindi data set and for English data set author has used pre-trained word2vec embeddings. (Ghosh et al., 2017) performed sentiment identification on code-mixed text data derived from social media. For their experiment, they used two code-mixed data sets English-Bengali and English-Hindi. They classified the data according to the polarity contradiction in the statement, such as positive, negative, or neutral. SentiWordNet word matches, opinion lexicon, and POS tags are employed, and the multilayer perception model is used to classify the polarity, with 68.5% percent accuracy. (Si et al., 2019) used statistical features such as tf-idf and linguistic features like emoji, part of speech, and emotion score to evaluate the performance of machine learning classifiers like XGBoost Classifier, Gradient Boosting Classifier (GBM), and Support Vector Machine (SVM) on three different datasets: English, Hindi, and Hinglish code-mixed. They obtained f1-scores of 68.13%, 54.82%, and 55.31% for the English, Hindi, and code-mixed datasets, respectively.

B. USING DEEP LEARNING MODELS

Recently, deep learning-based models have improved the performance of handcrafted feature models. A substantial amount of work has been done using deep learning models to detect hatred and inflammatory content. (Mathur et al., 2018) used a CNN-based transfer learning approach to detect abusive tweets. They also introduced the HEOT dataset and the Profanity Lexicon Set. In addition, (Mathur et al., 2018) classified hate speech in Hinglish using a Multi-Input Multi-Channel transfer learning architecture based on a CNN-LSTM network. (Kamble and Joshi, 2018)(Kumar et al., 2020) have built a domain-specific word embedding to detect hate speech in Hindi code mixed data and applied CNN, LSTM, and BILSTM as a classifier and found that word-level feature is the most contributing feature for detecting hate speech. (Santosh and Aravind, 2019) worked with existing code-mixed datasets for hate speech identification using two architectures: sub-word level LSTM model and Hierarchical LSTM model with attention based on phonemic sub-words. (Chopra et al., 2020) demonstrated how targeted hate embeddings combined with social network-based features outperform existing state-of-the-art models, both quantitatively and qualitatively. (Chakravarthi et al., 2020)(Kumar et al., 2020)(Saumya et al., 2021) presented a code mixed data set for Malayalam-English language obtained from offensive comments on YouTube and Twitter, also achieved a base line result of 75% F1 score using BERT's transformer model.

Because of the challenges mentioned in the preceding part, most of the models examined in Section II are not able to provide significant results. However, due to the availability of pre-trained models trained on larger corpora, hate speech detection can be performed better on monolingual data. As a result, we converted multilingual data to monolingual first and then performed feature extraction and classification in proposed approach.

III. METHODOLOGY

A. PROBLEM DEFINITION:

Let **S** =$\{s_1, s_2, s_3, ...s_n\}$ be the set of input tweets, and **L** = $\{l_1, l_2, l_3, ...l_n\}$ be the corresponding **n** labels for input **S**, where **S**$\in$$\{$Hate, Non-hate$\}$ denotes the presence and absence of hate speech, respectively. The goal of the proposed model is to predict the conditional label **'l'** for the given input **'s' i.e. P(l/s)**.

B. DATA SET DESCRIPTION AND PREPROCESSING:

The data set used to validate the proposed model was obtained from (Bohra et al., 2018). The data set include both normal and hate speech. The data collection contains 4575 code-mixed tweets, of which 1661 contain Hate speech, and the remaining 2914 code-mixed tweets in the data set consist of Non-hate speech. All of these tweets were scraped from Twitter using the Twitter Python API. The data set is slightly unbalanced, with two fields: Text and Label (Bohra et al., 2018). Several pretreatment methods were carried out on both the text and label fields to process model training data. The textual corpus had URLs, hyperlinks, figures, stop words, and capital characters. Various preprocessing exercises were carried out to simplify details, such as replacing punctuation with white spaces, removing URLs and Twitter account names that could not be used to identify hate news. Texts were also lowercased. Further, the data lemmatization was carried out to translate tweets' words into their useful basic form.

The proposed model (TIF-DNN) is built on three-layer architecture: the interpretation layer, the feature extraction layer, and the classification layer. Fig 1 illustrates the complete pipeline of the architecture.

C. INTERPRETATION LAYER:

The interpretation layer forms the first layer in our proposed model; cleaned and lemmatized tweets are input to this layer. The Interpretation is performed in the following four steps:

1) Each tweet is separated into several words at the start using Python's split() function.
2) In step 2, the Microsoft LID-tool[4] is used to annotate each word with its matching Lang-id. Language tags such as English, Hindi are used to annotate words.
3) In step 3, each annotated word is compared to its language id; if the Lang-id is English, the word is translated into the matching Devanagari term using Python's *Englishtohindi* module[5]. On the other hand, if Lang-id is Hindi, the word is transliterated to the Devanagari script using the Indic-transliteration function from the Indic-nlp-library[6].
4) Step 4, concatenates all transliterated and translated words to produce a single phrase that will be used as input to the feature extraction layer.

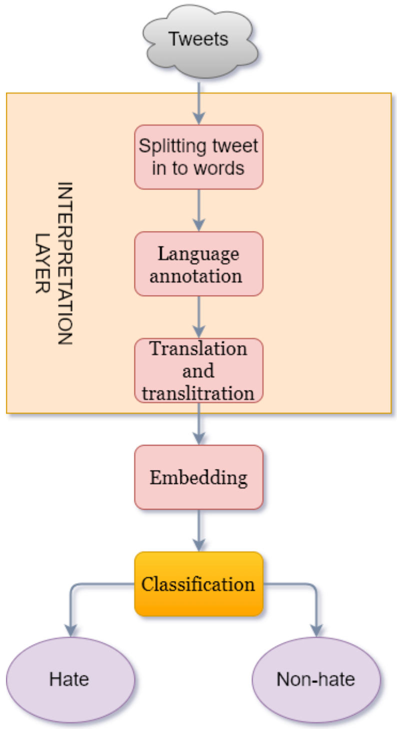

Fig. 1. Pipeline architecture of proposed model .

D. FEATURE EXTRACTION LAYER:

The feature extraction layer receives a monolingual tweet with the Devanagari script from the previous layer as input. The tokenizer is then given a Devanagari tweet to turn each tweet into a number of tokens, with each word in the tweet considered a separate token. Padding and masking for variable-length phrases were also performed in conjunction with tokenization. In proposed model, multilingual Bidirectional Encoder Representation (mBERT) tokenizer is used.

BERT is a bidirectional model that is built on the Transformer architecture (Devlin et al., 2018). The Transformer again relied on the attention mechanism (Vaswani et al., 2017). We employed multilingual BERT trained on 104 distinct languages from Wikipedia articles in proposed model for feature extraction. mBERT, like BERT, holds 12 attention heads and 12 transformer blocks. We only drew embedding from CLS tokens, which give full sentence embedding, which generates a 768-dimension vector for each tweet. These embeddings are sent into the classification layer as input.

E. CLASSIFICATION LAYER:

1) Baseline classifiers: A number of experiments were carried out on original code-mixed data without translation using various traditional machine learning algorithms such as Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RM), Naive Bayes (NB), and K Nearest Neighbors (KNN), which provide the baseline results for our proposed framework. These classifiers use the embeddings generated from mBERT for code-mixed data as input.

[4] https://github.com/microsoft/LID-tool

[5] https://pypi.org/project/English-to-Hindi/

[6] https://github.com/anoopkunchukuttan/indic$_{n}lp_{l}ibrary$

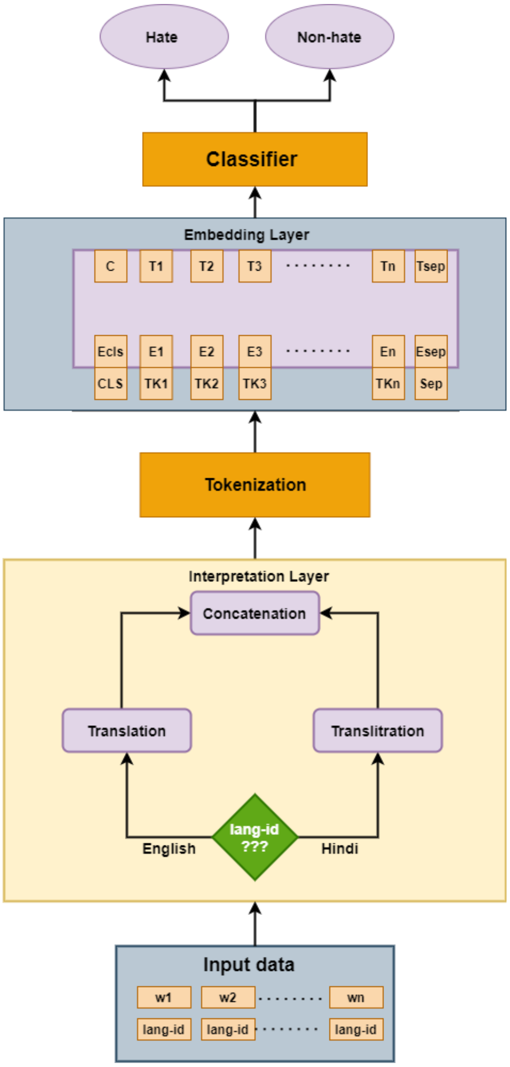

Fig. 2. Proposed TIF-DNN based architecture .

2) *Proposed model:* On hate speech data, we provide two types of models in our suggested model. First, we tested baseline classifiers on translated Devanagari script using mBERT embeddings. Later, we experimented with the Deep Neural Network (DNN) model, which acts as the second model in our suggested approach. DNN model comprises multiple dense layers, which aim to shape and compress the input in a meaningful fashion. Dense layers are those that are fully connected. A dropout layer follows each dense layer to avoid overfitting problems. We also used a batch normalization layer to normalize activation values; the normalization layer calculates new activation values as follows.

$$h^{norm}_{ij} = (h_{ij} - {}_j) \;/\; \sigma_j$$

$$h^{final}_{ij} = \gamma_j . h^{norm}_{ij} + \beta_j$$

Where $'\gamma'$ is mean, and $'\sigma'$ is the standard deviation. The detailed architecture of the proposed model is illustrated Fig 2

Classifier	Hyper-parameter
Logistic Regression	C=1, max-iter=500
Random forest	no-of-estimators=200
Naive bayes	var-smoothing=1e-09
Support Vector Machine	c=1, solver='lbfgs', kernel='linear'
K nearest neighbors	n-neighbours=24
TIF-DNN model	lr=1e-4, loss='binary-cross entropy', optimizer=adam

TABLE I
CLASSIFIER'S PARAMETERS

	Model	Accuracy	F1-hate	F1-nohate
Baseline classifiers	mBERT+LR	66	44	76
	mBERT+SVM	67	46	76
	mBERT+KNN	64	22	77
	mBERT+RF	64	21	77
	mBERT+NB	54	49	58
TIF based model	TIF-LR	70	48	77
	TIF-SVM	72	55	76
	TIF-KNN	68	44	78
	TIF-RF	66	28	77
	TIF-NB	56	57	55
	TIF-DNN	**73**	**56**	**78**

TABLE II
COMPARATIVE STUDY OF THE PROPOSED MODEL WITH BASELINE CLASSIFIERS

IV. EXPERIMENTAL SETUP AND RESULTS

Experiment is started with machine learning algorithms such as Logistic Regression, Random Forest, Naive Bayes, Support Vector Machine, and K-Nearest Neighbors. The parameters used during training for the aforementioned classifiers are shown in Table I. These algorithms were experimented with using the implementation provided by the Scikit-learn library, and trials were conducted using a train-test ratio of 70:30. These provide baseline results for proposed model. According to the results results shown in Table II, SVM and LR are the top-performing models, with an accuracy of 67% and 66% percent, followed by KNN and RF.

Interpretation layer is used in proposed model to transform multilingual input data to monolingual form since Hate speech identification on monolingual data sets has better accuracy. The translated data is subsequently processed by a mBERT for feature extraction. To classify extracted features, classical machine learning techniques are used; SVM achieves a higher accuracy of 72 percent among them. Later, we experimented with a Deep Neural Network-based model to improve the performance of existing techniques. In DNN model the mBERT input is initially passed through dense layers of sizes 1000, 500, 100, and 50; batch normalization and dropout of 0.4 are added to avoid over-fitting problems. Finally, findings are passed through the sigmoid layer for stance detection. Table II compares the outcomes of our proposed method with the baseline classifiers. We also compared proposed work with existing models, and the results are shown in Table III. On Twitter data, the proposed model outperformed existing methods for hate speech recognition.

A. *Limitations of our model:*

To understand the limitations of our model, we examine the outcomes of our models, which inspire new research

Model	Accuracy (HS)%
(Mathur et al., 2018)	72
(Bohra et al., 2018)(Random Forest)	65
(Bohra et al., 2018)(SVM)	71
(Santosh and Aravind, 2019)	71
Our TIF-DNN model	**73**

TABLE III
COMPARISON WITH EXISTING WORK

directions. Some of these limitations are as follows.

1) As shown in Table II,When compared to Hate speech, the proposed model exhibits higher accuracy for Non-hate speech identification. Unbalanced data used during training process might be one of the reason.
2) If the performance of the translator model used during interpretation is improved further, the performance of the proposed model can be improved. Our translation model struggles to find the exact translated term in Devanagari script for the few equivalent English words in a code-mixed tweet. As a result, there are significant mistranslations in our translated tweet.

V. Conclusion and future enhancements

The proposed approach investigates hate speech identification in a Hindi-English code-mixed Twitter data set. In this article, we proposed the TIF-DNN model for hate speech identification. We proved the efficacy of the proposed model by comparing the results of suggested model to baseline classifiers and past work. The findings also revealed that the proposed translator-based models outperform several baseline classifiers and existing work. However, better results may be obtained if a more powerful translation model is included in future studies. Furthermore, experiments detailed in this study can be repeated on other regional languages as part of future research, which is essential because India is a multilingual society with numerous local languages.

References

Arora, G. (2020). inltk: Natural language toolkit for indic languages. *arXiv preprint arXiv:2009.12534*.

Bohra, A., D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava (2018). A dataset of hindi-english code-mixed social media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pp. 36–41.

Chakravarthi, B. R., N. Jose, S. Suryawanshi, E. Sherly, and J. P. McCrae (2020). A sentiment analysis dataset for code-mixed malayalam-english. *arXiv preprint arXiv:2006.00210*.

Chopra, S., R. Sawhney, P. Mathur, and R. R. Shah (2020). Hindi-english hate speech detection: Author profiling, debiasing, and practical perspectives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 34, pp. 386–393.

Chowdhury, S. A., H. Mubarak, A. Abdelali, S.-g. Jung, B. J. Jansen, and J. Salminen (2020). A multi-platform arabic news comment dataset for offensive language detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 6203–6212.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Edosomwan, S., S. K. Prakasan, D. Kouame, J. Watson, and T. Seymour (2011). The history of social media and its impact on business. *Journal of Applied Management and entrepreneurship 16*(3), 79–91.

Ghosh, S., S. Ghosh, and D. Das (2017). Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*.

Kamble, S. and A. Joshi (2018). Hate speech detection from code-mixed hindi-english tweets using deep learning models. *arXiv preprint arXiv:1811.05145*.

Khan, M. U., A. Abbas, A. Rehman, and R. Nawaz (2020). Hateclassify: A service framework for hate speech identification on social media. *IEEE Internet Computing 25*(1), 40–49.

Kumar, A., S. Saumya, and J. P. Singh (2020). Nitp-ai-nlp@ hasoc-dravidian-codemix-fire2020: A machine learning approach to identify offensive languages from dravidian code-mixed text. In *FIRE (Working Notes)*, pp. 384–390.

Kunchukuttan, A., D. Kakwani, S. Golla, A. Bhattacharyya, M. M. Khapra, P. Kumar, et al. (2020). Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

Mathur, P., R. Sawhney, M. Ayyar, and R. Shah (2018). Did you offend me? classification of offensive tweets in hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pp. 138–148.

Mathur, P., R. Shah, R. Sawhney, and D. Mahata (2018). Detecting offensive tweets in hindi-english code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pp. 18–26.

Mossie, Z. and J.-H. Wang (2020). Vulnerable community identification using hate speech detection on social media. *Information Processing & Management 57*(3), 102087.

Pamungkas, E. W., V. Basile, and V. Patti (2020). Misogyny detection in twitter: a multilingual and cross-domain study. *Information Processing & Management 57*(6), 102360.

Qi, P., Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning (2020). Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Samghabadi, N. S., D. Mave, S. Kar, and T. Solorio (2018). Ritual-uh at trac 2018 shared task: aggression identification. *arXiv preprint arXiv:1807.11712*.

Santosh, T. and K. Aravind (2019). Hate speech detection in hindi-english code-mixed social media text. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pp. 310–313.

Saumya, S., A. Kumar, and J. P. Singh (2021). Offensive language identification in dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and*

Language Technologies for Dravidian Languages, pp. 36–45.

Senarath, Y. and H. Purohit (2020). Evaluating semantic feature representations to efficiently detect hate intent on social media. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pp. 199–202. IEEE.

Si, S., A. Datta, S. Banerjee, and S. K. Naskar (2019). Aggression detection on multilingual social media text. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5. IEEE.

Srivastava, V. and M. Singh (2021). Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.