IMPROVING SENTIMENTAL ANALYSIS OF HINDI-ENGLISH CODE-MIXED TEXTS THROUGH ENSEMBLE LEARNING

SURBHI MALIK

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2069257

COMMITTEE

Prof. Javad Pourmostafa Roshan Sharami
Dr. Dimitar Shterionov

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 2, 2022

ACKNOWLEDGMENTS

Dear reader,
Thank you for taking the time to read my thesis, which is an effort to create a reliable sentiment classification system using ensemble learning to identify hate speech in code-mix Hindi-English tweets. I express my sincere gratitude to my supervisor, Prof. Javad, for his unwavering guidance and support throughout the process. For all of their support, motivation and company over the past few months, I also want to thank my parents, brother, my managers (Eva and Marco), and friends (Isha, Daisy and Deepak).
Best Regards, Surbhi Malik

IMPROVING SENTIMENTAL ANALYSIS OF HINDI-ENGLISH CODE-MIXED TEXTS THROUGH ENSEMBLE LEARNING

SURBHI MALIK

Abstract

In multilingual communities, social media engagement has become more widespread, with a higher percentage of online communication performed in code-mixed language. Predicting Twitter sentiment in the context of the spread of hate speech for code-mix language proves to be a challenging task due to the dearth of methodologies, language diversity, complex semantics, and scarce resource availability. In an attempt to bridge the gap, this study's primary goal is to identify hate speech in tweets by introducing a robust sentiment classifier based on an ensemble model. From GitHub, two labelled datasets of tweets with profanity were obtained; they had 17,000 and 3,000 data points, respectively. Back translation and synonym substitution are data augmentation techniques that are used to enlarge the dataset since the volume is quite low. To minimise the noise from the code-mix text, rigorous preprocessing is done to clean the datasets. Both datasets were tokenized using the word2vec embedding method before being input into the models. The baseline ensemble model of BiLSTM is extensively contrasted with the proposed ensemble to reflect on the improvements. To create ensemble models, three neural networks were used: 1) CNN, a text classification supervised learning approach that extracts higher-level information from the sentences; 2) LSTM: a feedforward artificial neural network for classifying sentences; 3) BiLSTM: a bidirectional LSTM for processing sequence data and finding word associations in the target text. The study reveals that: 1) Word2vec is a prominent word embedding technique for text-to-numeric conversion; 2) The volume of the dataset has a significant impact on the training of the model; and 3) The proposed model was able to outperform the baseline model of BiLSTM, with an F1 score of 0.74. Consequently, the ensemble model with distinct classifiers is an effective strategy for the sentimental analysis of code-mix text. However, to arrive at conclusion, it is crucial to understand the loss of interpretability via knowledge distillation.

1 DATA SOURCE/ CODE/ ETHICS STATEMENT

In this research, two datasets used are obtained from the website GitHub [1], which provides a cloud-based Git repository hosting service to help developers manage, store, and track changes to their code. The data that will be analysed in this project is collected from Twitter [2], a social media platform for microblogging that links users by broadcasting posts in the form of text, photos, and videos, also known as tweets. Two publicly available datasets from GitHub will be deployed in this project to validate the generalisation of the proposed ensemble model; Training dataset 1: SemEval-2020 Task 9, an international NLP research workshop and competition to predict the sentiment of a given code-mixed tweet organised by SIGLEX (Bansal, 2020), Training dataset 2: Offensive Hinglish Tweet Classification from EMNLP (Empirical Methods in Natural Language Processing) 2018, a leading conference in the area of NLP and AI (Sawhney, 2018). It is acknowledged that I have no legal right to access the aforementioned data or code in question. To assess this work, I adhere to the "Ethics checklist for Student research with human beings". On GitHub, the code for this thesis is accessible to everyone at `https://github.com/surbhimalik/Masters_Thesis_Code_Mix.git` and the datasets are available at `https://github.com/surbhimalik/Master_thesis_datasets`

2 INTRODUCTION

In recent years, social media has become an integral part of everyone's daily routine. On social networks like Twitter, a tremendous amount of data is generated every second through blogging, commenting, and sharing photos and videos, known as tweets. For the purpose of online socialisation, people frequently share ideas, thoughts, opinions, sentiments, or individual characteristics in these tweets. Social media posts reflect people's emotions and important recurring patterns of interpersonal interaction. Sentimental analysis, also known as opinion mining, is a field of study that attempts to extract information about these opinions, feelings, and emotions from textual sources in order to determine the intent of the expression (Tariku, Meshesha, Hunegnaw, & Lemma, 2022).

Due to the social media platforms' limitless convenience, lack of restrictions, freedom, and anonymity, user interaction can frequently take the form of harassment and hate speech with increasing diversity (Juwita,

[1] `https://github.com`

[2] `https://twitter.com`

Effendi, & Pandin, 2021). However, it is critical to combat the unchecked spread of hatred, which has the potential to seriously harm our society and, in particular, marginalised people or groups. Therefore, sentimental analysis is a powerful technique that helps identify such damaging dialogue used to abuse, troll, or bully and succinctly classify the polarity of such posts on social media platforms.

With no restriction on using social media, there are boundless diasporas interacting with one another every day in their respective preferred languages. India is one such nation with numerous languages and a large diaspora. The number of languages spoken or written simultaneously within the same region is unfathomable, leading to the phenomenon of the "union of languages," also known as "Code-Mix," which helps the community communicate effectively. Given the low literacy rate and lack of cyber awareness, people are more easily persuaded to spread harmful information (Biradar, Saumya, et al., 2022). People have been using social media more frequently over the past ten years, and more intriguingly, they are communicating in code-mix script rather than just Hindi or English. The dissemination of hate speech via code-mix regional languages, however, is still unregulated (V. Agarwal, Rao, & Jayagopi, 2021).

Social networks face a significant challenge as a result of this linguistic phenomenon as people become more accustomed to speaking in code-mix languages. The traditional Natural Language Processing (NLP) system for sentimental analysis, on which businesses and social networks currently rely, can only handle a limited number of multilingual resources for the detection of hateful content. Although deep learning and machine learning models are capable of extracting semantic information from textual data, the code-mix data is noisy by nature and there is not enough data to fine-tune the models due to low data resource availability for Hinglish (Shorten, Khoshgoftaar, & Furht, 2021). In order to assist deep learning-based models in achieving better overall results, we hypothesise that using the ensemble learning technique will be beneficial. The goal of ensemble learning is to increase accuracy over a typical classifier by stacking different classifiers and combining their individual predictions (Gonçalves et al., 2022).

In this study, I'll outline an effort to accurately predict the sentiment expressed in code-mix Hinglish text with foul language using an ensemble learning model of three classifiers, Convolutional Neural Network (CNN), Long short-term memory (LSTM), and Bidirectional LSTM (BiLSTM), in an effort to prevent public abuse, bullying, trolling, and harassment on social networks. I'll also compare the model's performance to that of an ensemble model based on BiLSTM.

2.1 *Motivation*

Due to the fact that Hindi uses the Devanagari script [3], whereas English uses Roman script [4], Hindi and English have very different writing systems. Despite being the official language of India, people prefer to communicate on social media using a combination of English words and transliterated Hindi using the QWERTY [5]keyboard due to familiarity. This combination is known as "Hinglish" and is regarded as the "union language of Hindi and English." Users of Hinglish occasionally display hatred on social media in the form of derogatory language or intonation with the intent to harm a person or members of the public due to their religion, sexual orientation, or other characteristics (Sengupta, Bhattacharjee, Akhtar, & Chakraborty, 2022). If sentences from Hinglish are translated into English, it is challenging to analyse them precisely (Thakur, Sahu, & Omer, 2020).

From a societal perspective, it is crucial to find a solution to the problem of Hinglish users on social media platforms not being recognised for their negative code-mixed interactions. In order to prevent hate speech from spreading within Hinglish user communities, more attention must be paid to this specific instance of improved hate speech detection.

The use of Hinglish has evolved alongside the popularity of online communication. The classifiers used for the sentimental analysis for the polarity detection of monolingual text are still insufficient for use with multilingual text, which leads to obscurity in terms of the aspect understanding of the text or comment and ultimately raises the issue of incognizance (Chakravarthi et al., 2022). As a result, interpreting the meaning of code-mix statements prior to classification is extremely difficult. Therefore, from the scientific perspective, in order to prevent and minimise the misclassification of negative comments and tweets, a more capable sentiment classifier for code-mix text is thus necessary.

In addition to the significant social and scientific benefit of identifying the polarity of code-mix text for hate speech. Businesses cannot completely disregard the applications of sentimental analysis since it goes beyond polarity detection to reveal an individual's emotional state, feelings, and intentions in human speech. The ability to recognize the feelings of Hinglish users and use that information to improve customer loyalty and retention through improved customer service can prove to be a significant competitive advantage (Bueno, Carrasco, Ureña, & Herrera-Viedma, 2022).

---

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

[4] https://en.wikipedia.org/wiki/Latin_script

[5] QWERTY: a keyboard layout for the Roman script. https://en.wikipedia.org/wiki/QWERTY

2.2 *Research Questions*

Identifying and filtering code-mix text for sentiment analysis is a challenging task. Firstly, code-mixed text is commonly used in casual contexts where people's communication styles range from one another or from group to group. Understanding and recognizing task-specific annotations is fundamentally hampered by the coexistence of code-mixed language with noisy, monolingual text. Furthermore, the code-mix text also violates all grammatical conventions, which makes the text even more ambiguous (Srivastava & Singh, 2021). It is highly unlikely that one approach will be adequate to capture the human-level proficiency of readability because of the code-mix language's diverse characteristics.

Despite the fact that several techniques have been demonstrated to be effective for detecting hate speech in monolingual text using supervised and unsupervised machine learning models (Shahid Ul Islam & Sharma, 2021), it is still unclear how well an ensemble learning model will work to recognise task-specific annotations and ambiguity with hate content at different aspect levels. Therefore, we will be looking into the main research question that follows in order to take into account all the challenges:

> *To what extent can an ensemble learning model with three classifiers, CNN, LSTM, and BiLSTM, one trained on word-level profanity identification, the other trained for the entire sentence, and the last one trained on the surrounding associated words with the target obscene word within a window, respectively, accurately predict the polarity of the sentiment of code-mix Hinglish tweets for hate speech and abusive language detection?*

Word embedding is a helpful NLP process to record data vocabulary for the purpose of quantification of textual data that encodes the meaning of the words or sentences. By extracting the various textual data to create a numerical representation, the method serves as a link between human and machine language comprehension. It is expected that the words that are closer together in the vector space will be similar. For code-mix learning, the conventional word embedding techniques might not be the best option as a mixed set of languages poses a tedious exercise (Pratapa, Choudhury, & Sitaram, 2018).

Encoding textual information into an embedding space is a component of both Word2vec and Bag-of-Words. The Word2Vec consists of two models: Skip-gram and CBOW [6]. Skip-gram predicts the sentence context,

[6] Continuous Bag of Words Model

whereas CBOW predicts the word in a specific context. The Bag-of-Words model is another way to gauge the presence of well-known vocabulary words because it only considers the frequency of the terms. So, the first subquestion contrasts the two-word embedding strategies that we believe will be most useful for our research.

SQ1 *Among Word2Vec (Skip-gram and CBOW) and bag-of-word (BoW), which one is the most effective word embedding method for converting text data to numeric data for code-mix Hinglish tweets to provide better results?*

The ideal word embedding for the second task can be identified using the information in the previous subquestion. A deep learning-based ensemble model for hate speech detection for code-mix text will be used as a baseline in order to thoroughly compare it with the suggested 3-classifier-based ensemble model. The model needs to be trained on the text using three different inputs: input 1 is word-level embedding, input 2 is sentence-level embedding, and input 3 is embedding based on surrounding associated words with the target obscene word within a window. The purpose of the subsequent subquestion is to train on various input levels in order to enable the subject to independently determine sentence polarity and to indicate whether it can perform better than the conceptual model or not.

SQ2 *Does the previously proposed model outperform the ensemble deep learning model of BiLSTM for the polarity identification of the Hinglish text?*

In addition, the problem of low data resource availability for code mix languages like Hinglish is being addressed by using an easy data augmentation technique like back translation and synonym replacement to increase the volume of the dataset. On the augmented dataset and the original dataset, this task compares the classification performance of an ensemble model based on three classifiers.

SQ3 *To what extent does the ensemble model's performance change before and after the increase in the volume of the low-resource datasets for Hinglish tweets with the data augmentation technique such as Easy Data Augmentation (Back translation and synonym replacement)?*

2.3 *Main Findings*

The major finding of this thesis is that proposed ensemble models outperform baseline ensemble models based on BiLSTM both before and after applying data augmentation. Surprisingly, the F1 score for each of the three classifiers was likewise quite high. Additionally, Word2vec exceeded the BoW for word embedding for the Hinglish text and did so significantly

better. Additionally, there is a significant difference in the model's training processes before and after the data augmentation approach was used. However, the loss of interpretability caused by knowledge distillation, which is the foundation of the whole ensemble learning paradigm, is a problem that we cannot disregard. In order to do better research and detect hate speech or any other semantic information effectively, a Hinglish-based corpus is required in NLP. To lessen the dataset's sparsity and noise as well as to speed up the preparation process, better-preprocessing methods must be created. Finally, to understand the emotion of the regular users of Hinglish on social media networks, additional studies in the area of the Hinglish code-mix language has to be conducted.

3 RELATED WORK

The literature that has already been published is examined in this section to learn more about how the techniques used to ascertain the sentiment polarity of code-mix texts. In addition, the limitations and solutions for different algorithms are also reviewed, along with related research.

Despite the fact that the majority of earlier research work concentrated on improving the tools primarily used for monolingual or high-data resource languages such as English, researchers have recently become more interested in code-mixed languages after observing the non-standard writing style by the activity of the users on the social media platforms (Ananny & Crawford, 2018). Recently, researchers have started to look into and suggest models that are more accurate and have a better understanding of code-mix text. According to Rajeswari et al. (2020), there are three different approaches for the classification of sentiments, the lexicon-based approach, the machine-deep learning approach, and the fusion of various models using hybrid approaches or ensemble methods. Besides that, the introduction of transformer-based models has incentivised innumerable researchers to use the model for text classification and has firmly established the model as a state-of-the-art sequential modelling strategic approach (Tho, Heryadi, Kartowisastro, & Budiharto, 2021).

3.1 *Lexicon Based Approach*

At the moment, Code-mixing is highly prevalent in sentimental analysis. One of the methods used in the semantic analysis is a lexicon-based strategy. The method employs a gold-standard sentiment lexicon dictionary that is either manually or automatically generated and is composed of labelled words with a corresponding semantic score. The generalised rules of grammar incorporation, logical phrase construction, word orientation, and

arrangement of syntax are usually used to determine the intensity of the word score. A semantic score of how positive, negative, or neutral a word is, is produced by averaging the sentiment scores for the words in the prepared lexicon document, which contains each word and its corresponding sentiment score (N. Gupta & Agrawal, 2020).

Using supervised learning techniques on the annotated corpus of tweets with Hinglish, Bohra et al. (2018) conducted one of the groundbreaking studies for the detection of hate speech. The method focused on lexicon-based feature extraction at the character and word levels as well as lexicon-based features. However, there is an abundance of unstructured text data for code-mixing language syntax. Small datasets can be handled by this method, but it is highly dependent on lexicon resources, which are scarce for languages with limited resources. Due to the algorithm's lack of consideration for the context or aspect-level depiction, this approach was noticeably ineffective in determining the polarity of the sentence.

3.2 *Machine and Deep Learning-based Approaches*

In line with this, researchers have studied machine learning and deep learning techniques for sentiment analysis. In a recent study, Pravalika et al. (2017) suggested two methods for analysing sentiment in data that had Hindi and English code mixed together (Swamy, Kundale, & Jadhav, 2022). A lexicon is first created with an extensive list of the sentiments that are present in the sentence. Based on the generated lexicon list, the sentiment combination rules were inferred to determine the polarity of the sentence. The second method uses machine learning models that were trained on mixed-language social media data in order to extract features like grammatical transitions and frequently used patterns for determining the polarity of user comments. The experimental analysis's findings demonstrated that, with an accuracy score of 86%, the lexicon-based approach outperformed the machine learning model when using real-world data. However, this approach is still domain-specific.

The performance of features with deep learning-based models, however, has dramatically improved as a result of the models' extensive work in detecting hatred. Another study by Shalini et al. (2018) uses the deep learning model of CNN for the sentimental analysis of Indian languages, namely Bengali and Telugu. For tasks like sentence modelling, semantic parsing, and query search in computer vision and speech processing, CNN is a useful technique. The proposed model is an artificial neural network and uses activation functions such as ReLU with a single hidden layer for the semantic classifications. Due to the morphologically rich nature of one

language to the other, the model applied to the data produced varying results for different code-mixed languages.

The research by Santosh and Aravind (2019) examined an alternative deep neural network technique for detecting hate speech in code-mix text. The study compared and contrasted two different ways of using LSTM in order to ascertain which method worked better and produced a result that was statistically significant. They carried out two different sets of deep learning experiments using the hierarchical LSTM model with attention based on phonemic sub-words and the sub-word level LSTM model. The sub-word level LSTM model had a 69.8% accuracy rate. With phonemic sub-word attention, a hierarchical LSTM model provided an accuracy of 66.6%. With an F1 score of 48.7, the comparative study found that hierarchical LSTM outperformed sub-level LSTM in performance.

3.3 *Hybrid and Ensemble Learning-based Approaches*

Over the past few years, text classification techniques using hybrid and ensemble learning have garnered a lot of interest. While handling the more complex dimensional data, both methods have demonstrated improved results. The techniques show an efficient way to combine different learning algorithms to enhance prediction and utilise information fusion in various ways. In contrast to ensemble learning, which combines multiple homogeneous weak learners who work independently for the level of their individual output and can be grouped by majority voting techniques to get an outcome, adaptive hybrid models use multiple simple algorithms to solve problems while predicting a single outcome (Ardabili, Mosavi, & Várkonyi-Kóczy, 2020).

A hybrid approach using two classifiers was proposed in a recent study by Mathur et al. (2018) to categorise hateful and abusive tweets. The CNN-LSTM neural network is used in the hybrid approach as the architecture for the MIMCT model, Multi-Input Multi-Channel learning. In order to outperform the naive transfer learning model, the primary model makes use of multiple embeddings (word2vector and FastText), while CNN-LSTM simultaneously makes use of secondary semantic features. Together, the two classifiers identify the features that are most helpful in identifying abusive words and forecast a single result. With an F1 score of 0.83 and the Twitter word2vec embedding, the hybrid approach appeared to be functioning well, but it was still unable to comprehend the annotators' indirect animosity toward the entire sentence during preprocessing.

In a study that was motivated by the previous hybrid approach, V. Gupta et al. (2021) employed CNN and LSTM models based on deep learning but preferred to use the ensemble learning technique to identify the profanity

in the Hinglish code-mix content rather than the hybrid approach. The proposed ensemble architecture was developed as a stacked model using the FastText embedding technique with multiple dimensions, where each word will be replaced with its corresponding FastText embedding vectors. With an F1-score of 0.87, the model outperformed all baseline models based on basic machine and deep learning methods. The model, however, was unable to extract the text's context because the study only looked at the effects of dimension attributes of the embedding in the data. Working with character- or word-level embedding, which is potentially a more reliable method to comprehend the context in a better way even when there isn't a larger corpus of Hinglish text, was not included in the proposed model.

For the purposes of this study, an ensemble learning strategy based on three different classifiers, each trained on a different feature attribute, will be used to accurately predict the polarity of a text while capturing the context, regardless of the noisy nature of the dataset and the semantic orientation of the sentences.

3.4 *Transformer based Approaches*

Additionally, BERT [7] a transformer-based architecture, is a pre-trained, cutting-edge model that is gaining popularity for handling Hinglish code-mix text. Researcher Vashistha et al. (2020) uses a hybrid strategy combining BERT and BiLSTM since both models have the capacity to capture the sentence-level and utterance-level knowledge of the sequences. Despite the noisy and repetitive nature of the data, the BERT-based model supported the sentences' multilingual aspect and was able to comprehend the annotation. Although the BiLSTM's presence had no impact on the model's overall performance. The model did, however, demonstrate a significant improvement, but tends to have a bias problem because it is trained to recognise the precise places in a text where code-mix occurs. The model would not perform as well when presented with code-mix text where the code switch occurs at different locations than what the model has been trained to identify, which could result in errors or misclassification.

[7] Bidirectional Encoder Representations from Transformers

4 METHODOLOGY AND EXPERIMENTAL SETUP

4.1 *Simulation Framework*

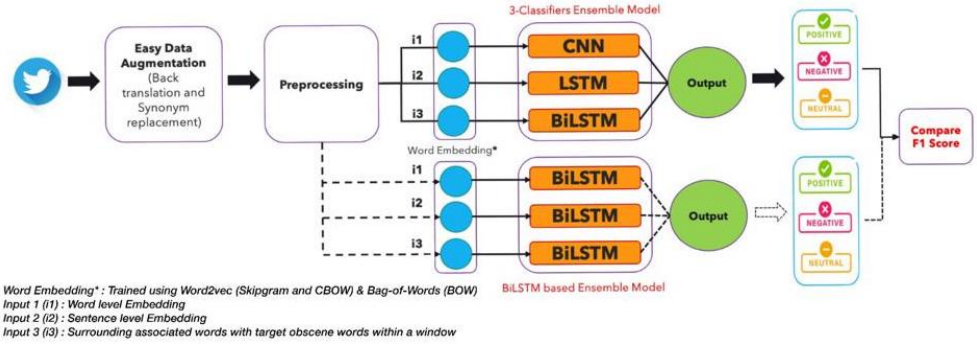

Figure 1: Simulation Framework

4.2 *Dataset*

The data that will be analysed in this project is collected from Twitter. Two publicly available datasets from GitHub will be deployed in this project to validate the generalisation of the proposed ensemble model; Training dataset 1: SemEval-2020 Task 9, an international NLP research workshop and competition to predict the sentiment of a given code-mixed tweet organised by SIGLEX (Bansal, 2020), Training dataset 2: Offensive Hinglish Tweet Classification from EMNLP (Empirical Methods in Natural Language Processing) 2018, a leading conference in the area of NLP and AI (Sawhney, 2018).

Table 1: Raw data

Dataset Name	Training Data	Labels
SemEval 2020 Task 9	17,133	Negative - 0 Neutral - 1 Positive - 2
Offensive Hinglish Tweet (Modified labels)	3,000	Negative - 0 Neutral - 1 Positive - 2

However, due to low-data resource availability for code-mix Hinglish text, we had to adapt data augmentation techniques to get a more gener-

alised result. The preferred technique will be Easy Data Augmentation techniques using back translation and synonym replacement.

4.3 *Easy Data Augmentation Technique (EDA)*

In all of the languages that are used regularly, code-mix sentences have an extensive vocabulary and intricate syntactic structures as they constitute two or more languages at the same time. Due to the informal and spontaneous character of code-mix interactions, it is essential to have a strong command of both languages in order to comprehend the content accurately. According to K. Agarwal and Narula (2021), Hinglish code-mix and profanity are very noticeable together in social media networks, but it is still difficult to collect enough data specifically containing hate content to improve their performance on automatic polarity detection. However, it is essential to have a large dataset to prevent over-fitting.

By purposefully altering the existing data points to produce additional data points, data augmentation is a technique that will be used to increase the volume of the dataset. It serves as a regulariser to expose the model to various versions of the data and broaden its generalizability. For the sake of this study, we are recommending a technique that enables the quick creation of a corpus optimised for code-mix dataset that is language-neutral, accessible, and can also keep the sentence's meaning while maintaining the labelling.

For code-mixed datasets, numerous data augmentation methods have been proposed but not yet tested. The proposed method for multilingual code-mixed data augmentation is inspired by Jahan and Oussalah (2021). Our proposed method exhibits two approaches, back translation and synonym replacement. Back translation, is the act of translating from one target language to another and then back to the original source language. When a text is translated, it is most useful for text reconciliation. The goal of back translation is to ensure that the text's meaning is maintained even after it has been translated into another language (Feldman & Coto-Solano, 2020). This approach is very successful in producing high-quality data with little confusion, ambiguity, or mistakes. In addition, one effective method for producing textual data is synonym replacement, which adheres to the idea of Easy Data Augmentation (Duong & Nguyen-Thi, 2021). In the newly generated sentence, the words that have the most comparable synonyms are substituted, retaining the statement's utility by adding new vocabulary and keeping the meaning intact.

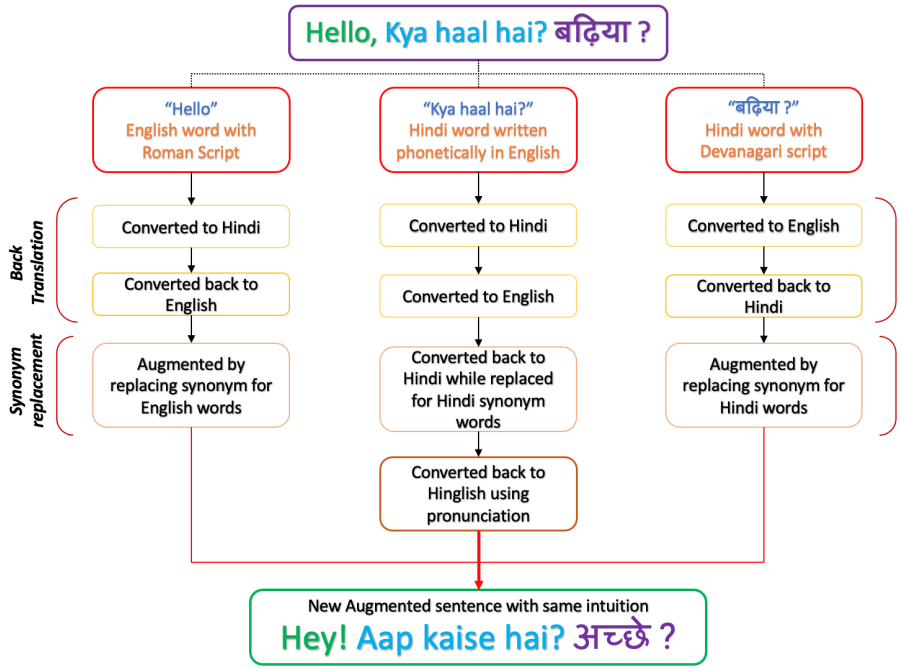

Figure 2: Data augmentation simulation framework

Hindi tokens, English tokens, and Hindi words rendered phonetically in English tokens are the three different types of tokens utilised in the datasets that will be used to train the models. It is difficult to tackle each token in a similar way due to language and script variations. Therefore, each token will be reformed into a meaningful new sentence with a different approach. Both English and Hindi words in a sentence are interchangeably translated into each other and follow the concept of back translation for the generation of a new sentence with the same intact meaning but having different vocabulary based on the synonym replacement. Following that, the token will then be treated as a Hindi word for the production of a new sentence with the synonym replacement if the phrase contains a Hinglish-based token. In the final step of handling the Hinglish word, a python-based library of google translate will be used to convert the Hindi word to Hinglish based on their pronunciation with the goal of continuing to investigate the code-mix text rather than translating it into a particular language and handling it as a monolingual text.

By putting the described data augmentation approach into practice, we were able to increase the amount of dataset 1, from 17K training data points to 40K augmented sentences, and similarly for dataset 2, from 3K to 30K increase in the dataset volume while also resolving the issue of class imbalance.

Table 2: Raw dataset before and after the use of Easy Data Augmentation

Dataset	Training Data Before Augmentation	Training Data After Augmentation
Dataset 1	17,133	**40,000**
Dataset 2	3,000	**30,000**

4.4 *Pre-Processing*

Prior to augmentation and before feeding them to the model, we first process both datasets. In this section, we will introduce the methods and strategies for processing the raw data.

The code-mix dataset frequently has a somewhat haphazard structure. It allows flexibility to communicate effectively between languages, yet causes noisy text, which is a commonly noticed pattern. Cleaning the dataset and making the data more comprehensible involves a number of clearly defined issues, such as incorrect word placement or spelling inconsistencies, informal or creative writing styles, spotting linguistic ambiguity, missing context, and improperly inserted special characters (Babanejad, Agrawal, An, & Papagelis, 2020).

A significant chunk of the excessive noise in the code-mix text is composed of punctuation and negations (Srivastava & Singh, 2020). First, all special characters such as backslashes, hyperlinks, periods, ellipses, and mentions from each data point in both datasets were removed because they offer little to no insight into the sentiment of a statement. To decrease case sensitivity in the dataset, all text has been changed to lowercase and emoji's have been converted to text descriptions. The phrase "http" and any other redundant tags associated with it were decided to be removed because the word "URL" does not have word embedding representation in the models. Next, inappropriate word spelling and spelling variations including repeated letters were eliminated using regex. To reduce the noise in the datasets, unusual words have been identified, eliminated, and in some cases, negative short forms have been replaced with standard word punctuation. To make new words and keep the sentence's context intact, words have been concatenated. Pos-tagging has been disregarded, nevertheless, as a result of how it affects categorization accuracy. Last but not least, exclamation and question marks were left alone since they are valuable markers of the strength of a text's mood.

4.5 *Word Embedding Approaches*

The crucial NLP technique of word embedding has improved computers' ability to represent and understand textual input. It takes data from words or texts and transforms it into a language that NLP engines can comprehend. While encoding words in a real-valued numeric vector format, the approach preserves syntactic and semantic information, allowing related words to have comparable vector values but in a lower dimensional space. Thus, the algorithm processes the text input after converting it to a numeric representation and then learns the representation. Additionally, the problem of a sparse matrix resulting from a small vector size is also resolved by word embedding, as is the problem of a high computational cost of training resulting from a large input vector size (Li & Yang, 2018).

In 2013, Google created the Word2vec technique (Desai et al., 2022). The approach is used to address complex text categorisation issues and is based on the distributional hypothesis (List, 2022). The rationale behind the hypothesis entails repeatedly reading through a corpus of text and discovering that similar words or phrases have significant semantic affinities while mapping those same words to the geometrically close embedding vector using Cosine Similarity metrics. Two shallow variants of neural networks, skip-grams, or CBOW, each with an input layer, output layer, and projection layer, are used in the model.

Two distinct model architectures, CBOW and Skip-gram, produce word embedding by following the intuition that has been theorised. In contrast to CBOW, which uses a deep learning classification model to predict the target word using input from the context, skip Gram is also an unsupervised learning technique used to predict the associated context words based on the target word. In CBOW, context words are input into an embedding layer that is initialised with random weights, followed by a lambda layer to average out word embedding, and then a dense softmax embedding layer that predicts the target and updates the weights based on the computed loss. Similarly to this, since skip gram must predict several words from a single target word, pairs of context and target are sent to the embedding layer to provide dense word embedding for the two words. It functions by increasing the likelihood that words will be predicted by the words in their context. To determine whether to output 0 or 1, the dense sigmoid layer receives the dot product value of the two-word embeddings from the merged layer (B. Liu, 2020).

Bag-of-words is another popular and straightforward text encoding method. Under the erroneous premise that each word occurs independently of the others, the text modelling approach for feature extraction totally depends on the word frequencies (Yan, Li, Gu, & Yang, 2020). The method

is basic and adaptable; it counts the number of words used while ignoring any semantic information, such as grammatical intricacies or word order. We can also convert variable-length text into a fixed-length vector, as a machine or deep learning models prefer structured numerical data over textual data at a finer level.

Both approaches have benefits and drawbacks over one. Since each word in BoW is represented as a scalar quantity, the problem of corpus sparsity arises since the feature dimensionality is strongly dependent on the uniquely tokenized words even when the corpus is enormous. There are many approaches to partially address this issue by preparing the data in a particular way, such as deleting stop words, stemming, and lemmatization, but doing so risks eradicating important information required to grasp the context of the phrase. Additionally, the technique lacks a mechanism for maintaining the linkage between tokens. Similarly, there are multiple shortcomings to Word2vec, such as the linear relationship between feature vectors and the black box. Based on the given training dataset, the approach is rather domain-specific. Additionally, even though each word's morphology is the same, it would be treated as a new distinct tokenized vector.

By taking into account both the strategies' advantages and drawbacks, for the purpose of this study, Word2vec will be implemented for code-mix data.

4.6 *Models*

4.6.1 *Convolutional Neural Network (CNN)*

A CNN is a profound deep learning, feed-forward artificial neural network algorithm that has the capacity to evaluate various aspects of an image or text while allocating weights and biases based on learnable information. The model has shown a promising result in the sentence classification space, boosting their credibility for a range of NLP tasks. Text data is a one-dimensional array. Three layers make up a CNN model; the first layer is a convolutional network, the second is an additional pooling layer, and the third is a fully connected layer. First and foremost, the sentences must be broken up into word embedding for low-dimensional representation by using a series of filters of different dimensions. The original sentence matrix's dimensions are decreased into a matrix with a lower dimension of 1D in order to extract semantic features from the text. The majority of computations are handled by the convolutional layer, which convolves with a number of kernels. The convolutional layer maps out the features, and layers are then pooled to offer different filters on the input. With each layer

that is added, the CNN model becomes more sophisticated as it searches the data for patterns or significant information. After that, the CNN uses pooling to streamline computation, maintain important features from one layer to the next, and minimise the output's size from the preceding layer. The maps are then collapsed into a single column, and the pooled output from the preceding layer of stacked feature maps is fed into the final fully connected layer of CNN.

Therefore, we employ a 1D convolutional layer CNN model in this work to handle tokenized word-based datasets at the character level embedding. Then, to decrease learning parameters and processing, a pooling layer is introduced. Finally, two dense layers are developed. The first is based on a 10-unit layer computing ReLu, which allows for quicker training and reduces the likelihood of the gradient disappearing. The last layer has simply a 1-unit layer that computes the softmax probability.

Without having any prior knowledge of the syntactic or semantic structure of the languages, CNN can be applied straight to the unique set of words. The capacity to ignore the noise in data for which segmentation is not possible is another advantage of the methodology (Zhang, Zhao, & LeCun, 2015).

4.6.2 *Long Short Term Memory (LSTM)*

The following model implemented in the ensemble model for this study is the LSTM, a version of RNN [8] that performs better than the conventional RNN due to its memory capacity. One of the primary characteristics of the model is its ability to memorise and retain the most crucial information while rejecting redundant data that is necessary for classification or prediction from sequential data. Here, a forward pass unidirectional LSTM model with various layers of data persistence is being taken into consideration. The model's framework is composed of the forget gate, input gate, and output gate. To determine whether incoming information must be learned by the model, the forget gate in the first layer is in charge and receives two inputs, which are then sent through a sigmoid function that eliminates the data that has a calculated inclination toward zero while allowing the remaining necessary data to flow through the gate. The cell state is saved with the most recent calculated information regarding the pertinent information by multiplying the output with the forget gate. The input layer, which is the second layer, evaluates the significance of the information from the cell state and stores just the essential information in the memory. The sigmoid function, which governs the network and lessens

[8] Recurrent Neural Networks

bias, is applied to the two created inputs, respectively. By conducting pairwise addition using the output from the input gate and the newly acquired information, the cell state is updated once more in order to provide the neural network with fresh values. The output gate of the circuit then determines the network's subsequent hidden state, where data is once again processed by the sigmoid activation function. In our implementation, dropout regularization is used to reduce the complexity of the model by dropping different groups of features from each sample to avoid overfitting. As a result, the information that should be transported was decided by the final output layer. At sentence-level embedding, the LSTM model is highly beneficial. It works well for remembering key information with word strings and conducting semantic parsing to determine the polarity for categorising sentiments (Staudemeyer & Morris, 2019).

Other ML or DL algorithms are typically trained just on numerous words as independent inputs where the words do not actually have a sentence-level meaning, and the prediction is based on the statistical output and not the actual context. With the proper use of embedding layers and encoding, LSTM creates its own unique features that enable it to accurately forecast the outcome and determine the input's true meaning. Additionally, the model purges unnecessary data, greatly reducing the computational cost. Consequently, LSTM is an effective method for classifying texts.

4.6.3 *Bidirectional Long Short-Term Memory (BiLSTM)*

BiLSTM, the study's final ensemble learning model, excels at solving sequential modelling issues and is frequently applied to text classification. In order to capture long-term dependencies without keeping redundant context information, the model is made up of two parallel LSTM units that run in both directions, forward and reverse. The BiLSTM model has a unique architecture that allows the first model to take the input as-is and the second model to take the input in the opposite way, effectively improving the quality of data that is available to the network with richer context.

The attention-based BiLSTM with convolutional layer model described by G. Liu and Guo (2019) is utilised in this work. The word embedding vectors are employed by the convolutional layer to extract the higher-level phrase representation, and the BiLSTM is then used to access the context representation for both the previous and subsequent words using the 2 layers of bidirectional LSTM. To deliver the information output from the BiLSTM's hidden layers of distinct foci, attention mechanisms are used. To categorise the processed data, two thick layers of softmax 10-layer units are utilised, followed by a 1-layer unit of ReLu classifier. Hence, the phrase-

level local features as well as the overall sentence semantics can both be captured by BiLSTM (Jain, Kumar, & Garg, 2020).

By enabling the model to automatically extract meaning from lengthy word sequences, the BiLSTM model is able to manage the sensitive representation of polysemous words. The model is the most favoured one for our study since it will enable the model to prioritise the target words while also emphasising the nearby connected words for sentence classification. The model will be able to extract the more subtle elements from the phrases' rich context. In light of this, it is an effective tool for modelling the sequential relationships between words and phrases in both directions of sequences.

4.6.4 *Ensemble Model*

In this project, I will present an algorithm to enhance sentimental analysis for code-mix text using ensemble learning techniques with CNN, LSTM, and BiLSTM. One classifier will be trained on the word-level classification of profane words, the other on the entire sentence, and the final classifier will be trained on the terms associated with the obscene words in a window, respectively. In order to increase accuracy in the task of determining the polarity of the code-mix statement, all three classifiers will be combined. While LSTM is better able to handle semantic parsing for sentence-level feature classification, the CNN model performs better for character-based tokenized embedding. BiLSTM, on the other hand, can automatically extract information and polysemous words from lengthy sequences. Combining CNN, LSTM, and BiLSTM is a novel approach for incorporating word embedding and semantic information from long sequences of polysemous words. The ensemble model of CNN, LSTM, and BiLSTM is anticipated to capture the predictions using bagging for the most accurate polarity identification and generalisation.

4.7 *Baseline model of BiLSTM*

The baseline model is a straightforward but imperfect benchmark model that yields decent results and doesn't take technical knowledge to construct. In order to better comprehend how well the suggested model is functioning when compared, it is crucial to reflect on your current data by constructing a reference point from it. Understanding the cost-benefit trade-off and allocating the increased performance are the two advantages of setting a baseline (Kodali et al., 2022). The creation, upkeep, and training of machine-deep learning models for huge datasets are computationally costly. Looking at the baseline provides you with a preview of the data-related observations

that can aid in model selection even before creating a highly complicated model with little predictive potential. Second, it is simpler to start from a place of knowledge with the benchmark performance indication from the baseline model knowing the parameters that need to be adjusted and engineered to increase the performance of the suggested model.

Based on BiLSTM, a baseline model was developed in order to compare performance with the proposed methodology. The baseline model's parameters are identical to those of the suggested model since both use the same input levels. The BiLSTM model will be trained on each of the three input levels in turn. The models are then contrasted with one another for performance improvement based on the F1 score.

4.8 *Evaluation Metrics*

The process of classifying tweets into positive, negative and neutral sentiments can be treated as a classification problem. The objective of sentiment classification is to make the best predictions of sentiment on unknown datasets in relation to the actual dataset. The evaluation of sentiment categorization for code-mix Hinglish tweets was initially conducted using a variety of other evaluation criteria, including accuracy, precision, recall, and F1 score. Accuracy is a measure of how close a value is to both its predicted value and its true value. It has been used as an evaluation metric, but it appears to be unreliable in two situations: first, when the dataset has a multiclass label distribution, and second when the dataset is severely unbalanced. Both of the factors are present in the dataset, making it exceedingly challenging to determine whether the classes are predicted equally or whether a higher level of accuracy is being attained by basing the prediction on the most prevalent class value. Hence, the datasets employed in this study are labelled and since the distribution of the code-mix hate speech text is inherently imbalanced, precision, recall and F1-score are the most appropriate evaluation metrics to reflect the ability of the ensemble models to correctly identify polarity.

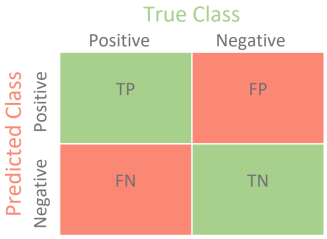

Figure 3: Contingency table

Using a unique contingency table known as the confusion matrix, the prediction performance assessment of any classification method with two or more classes as an output can be illustrated and summarised. In all dimensions, the table has identical sets of class-wise distribution mapping out the actual vs. predicted values to which the data belong. This thesis will create a 3*3 size matrix because it deals with a multi-class classification problem with three separate labels. Calculations will be made for each class's True Positive (TP), True Negative (TN), False Positive (FN), and False Negative (FN) values. Precision, Recall and F1-score are the metrics we'll be evaluating in this paper (Tahata, 2022).

$$Precision = \frac{TP}{TP + FP}$$

When comparing the total projected positive values to the actual positive instances, precision tries to determine the computed fraction of real positive cases. A situation where there are more negative instances than positive ones, which denotes a high incidence of false positives, is known as a type I error. An array of class imbalance datasets are used in this study. As a result, precision is superior since it can discriminate between the classes and is more focused on the positive class than the negative one.

$$Recall = \frac{TP}{TP + FN}$$

Recall makes an effort to calculate the percentage of accurately predicted positive cases compared to the actual positive values. When there are more positive than negative cases, which denotes a high incidence of false negatives, the situation known as type II error is said to exist. The recall is seen as a good metric in the same way that accuracy is since it places more emphasis on the positive classes than the negative ones.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The F1 score represents a harmonic mean of precision and recall. Because it combines contributions from both into a single evaluation metric, this metric was chosen. If the default beta value is less than 1, precision is given more weight, but recall receives more weight if the beta value is

larger. The metrics will be used to compare the baseline BiLSTM ensemble model with the proposed 3-classifier-based ensemble model.

Training datasets will be divided into two-fold cross-validation with a proportion of 80% for training and validation (development set) and 20% for testing the model in order to evaluate the performance of the algorithm on test data. Additionally, the datasets will be divided into two-fold with the same proportion after the easy data augmentation technique (back translation and synonym replacement) has been used to extend them in order to observe the F1 score of test data with a larger volume of the datasets. On the second training dataset, the model will be assessed for generalisation.

4.9 *Software and Hardware*

To develop all of the models and transform the data, the following software was employed. Python was chosen as a programming language for this thesis. The packages being used are Pandas, NumPy, Matplotlib, Keras, Tensorflow, pickle, and Seaborn.

This research is carried out using Google Colaboratory Pro and Jupyter notebook, which has a high-performance GPU unit with additional storage and RAM capacity.

5 RESULTS

This section will present, using the F1 score, the performance of the proposed 3-classifier-based ensemble model. Each of the models, CNN, LSTM, and BiLSTM, will be trained on various aspect levels of sentence interpretation. The model will then be contrasted with the baseline model produced using the same ensemble learning architecture for BiLSTM with identical input levels. The appropriate word embedding technique for working with the textual dataset will also be discussed. Finally, utilising data augmentation techniques on both the deployed dataset and the model, before and after comparisons will be made.

5.1 *Word Embedding*

In this experiment, we used word embedding techniques based on a machine learning algorithm to determine the accuracy rate of text-to-number conversion. Words were tokenized for Bag-of-words based on the frequency of each word occurrence, and over 17K features were retrieved for information retrieval. Using hierarchical softmax, the pre-trained model

for Word2vec was developed using Gensim [9], which incorporates both skip-gram and CBOW. While collecting the intensity and meaning of each word in a dataset, the words are mapped to a vector of the actual numbers. A simple statistical machine-learning model of logistic regression was constructed to capture the accuracy of the augmented dataset at a base level in order to evaluate the performance of both techniques. According to table 3, Word2Vec surpassed Bag-of-words with a 92% accuracy rate, while Bag-of-words stayed at 78%. As a result, Word2vec succeeds when it detects semantic similarity inside a long sequence using cosine similarity. Based on the results, we chose Word2vec to further tokenize the words.

Table 3: Accuracy for word embedding after the augmentation on dataset 1 based on logistic regression.

Models	Accuracy After Augmentation
Word2Vec (Skipgram and CBOW)	92%
Bag-of-Words	78%

5.2 *Sentiment classification on BiLSTM ensemble model*

The BiLSTM ensemble model was built identically to the proposed model for the baseline, with three distinct input levels. The BiLSTM was used to train each of the input levels to determine how well the model understands the difference in the input presented. The ensemble model outperformed dataset 1 when compared to dataset 2, owing to the larger size of the training dataset. As shown in the current table 4, the F1 score for dataset 1 is greater than that of dataset 2.

Table 4: F1 scores for baseline BiLSTM ensemble model on the negative class before and after the augmentation on dataset 1 and 2.

BiLSTM ensemble model	F_1 score Before Augmentation	F_1 score After Augmentation
Dataset 1	0.44	**0.59**
Dataset 2	0.36	**0.64**

[9] https://radimrehurek.com/gensim/models/word2vec.html

5.3 *Sentiment classification for ensemble model*

Table 5 shows the absolute number of tweets predicted by the suggested technique per model for the negative class before augmentation. Surprisingly, the CNN model outperformed the precision in terms of recall, implying that the model will predict more false negatives. Although, as compared to the expanded dataset results, the ensemble model of both datasets provides relatively low precision and recall.

Table 5: Classification report on the negative class before augmentation for 3-classifier ensemble model

		Classification	
Dataset	Models	Precision	Recall
Dataset 1	CNN	0.57	0.63
	LSTM	0.45	0.49
	BiLSTM	0.31	0.35
	Ensemble Model	**0.42**	**0.55**
Dataset 2	CNN	0.49	0.53
	LSTM	0.42	0.51
	BiLSTM	0.34	0.29
	Ensemble Model	**0.36**	**0.45**

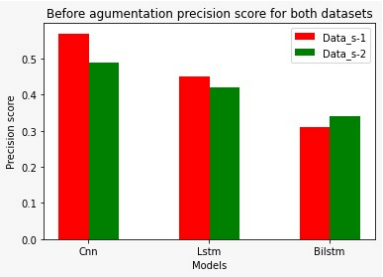

Figure 4: Precision before augmentation

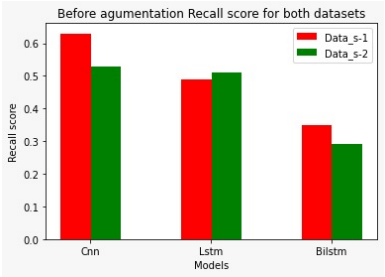

Figure 5: Recall before augmentation

Similarly, table 6 describes the absolute number of anticipated tweets for the negative class following data augmentation. The ensemble model precision in dataset 1 is 0.72, indicating that the model will predict more False Positives than in dataset 2, which has a precision of 0.67. Whilst the recall score for data set 1 has also improved, implying that the model would return the relevant result. As a consequence, a high precision and recall score indicates that the classifier is producing accurate results with

the majority of all positive findings. As an outcome, Dataset 1 outperforms Dataset 2, owing to the smaller training dataset available for dataset 2.

Table 6: Classification report on the negative class after augmentation for 3-classifier ensemble model

		Classification	
Dataset	Models	Precision	Recall
Dataset 1	CNN	0.78	0.81
	LSTM	0. 74	0.79
	BiLSTM	0.69	0.71
	Ensemble Model	**0.72**	**0.77**
Dataset 2	CNN	0. 74	0.78
	LSTM	0.63	0.67
	BiLSTM	0.65	0.65
	Ensemble Model	**0.67**	**0.69**

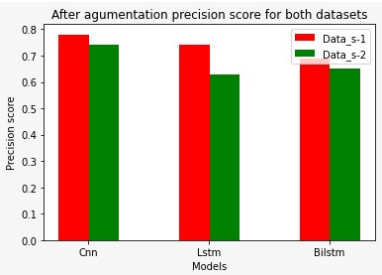

Figure 6: Precision before augmentation

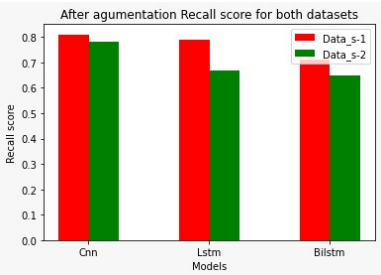

Figure 7: Recall before augmentation

Finally, using the ensemble model, we will compare the generated F1 score for both datasets. The higher the F1 score, the greater the balanced capacity to record both positive cases (recall) and to be accurate with the ones it does capture (precision). Before and after data augmentation, dataset 1 has an F1 score of 0.48 and 0.74, respectively. Following data augmentation, the model improved the outcome. The F1 score for the 3-classifier ensemble model is 0.74, which is considered a suitable result for any further classification based on the complexity of text data acknowledged by the model.

5.4 *Models Comparison*

Following extensive training on both datasets, a detailed comparison between the two ensemble models revealed further information on the pref-

Table 7: F1 scores on the datasets before and after augmentation respectively.

		F_1 score	
Dataset	Models	Before Augmentation	After Augmentation
Dataset 1	CNN	0.60	0.79
	LSTM	0.47	0.76
	BiLSTM	0.33	0.70
	Ensemble Model	**0.48**	**0.74**
Dataset 2	CNN	0.51	0.76
	LSTM	0.46	0.65
	BiLSTM	0.31	0.65
	Ensemble Model	**0.40**	**0.68**

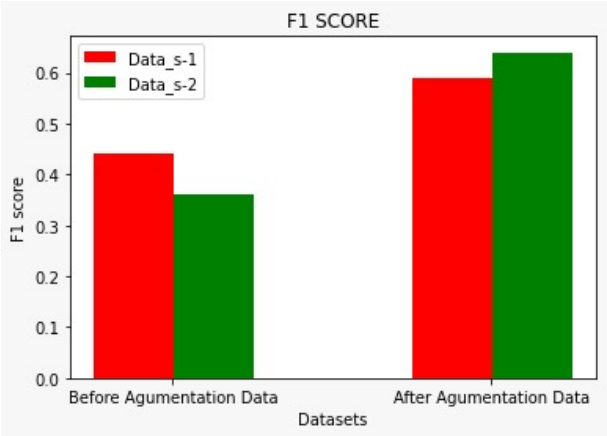

Figure 8: F1 score comparison of ensemble model before and after augmentation on dataset 1 and dataset 2.

erence for hate speech detection. With an F1 score of 0.74 on dataset 1 and 0.68 on dataset 2, the 3-classifier ensemble model beats the BiLSTM model. Both models did not perform well prior to the augmentation, but the results improved with the larger dataset. As a result, we may conclude that the 3-classifier ensemble model outperforms the other.

Table 8: F1 scores comparison for baseline 3-classifier based ensemble model and BiLSTM ensemble model before and after the augmentation on dataset 1 and 2.

		F_1 score	
Dataset	Ensemble Models	Before Augmentation	After Augmentation
Dataset 1	3-classifier	0.48	**0.74**
Dataset 1	BiLSTM	0.44	**0.59**

Table 9: F1 scores comparison for baseline 3-classifier-based ensemble model and BiLSTM ensemble model before and after the augmentation on dataset 2.

		F_1 score	
Dataset	Ensemble Models	Before Augmentation	After Augmentation
Dataset 2	3-classifier	0.40	**0.68**
Dataset 2	BiLSTM	0.36	**0.64**

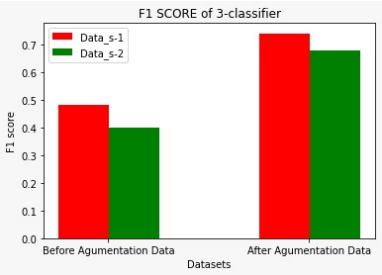

Figure 9: F1 score of 3-classifier ensemble model (dataset 1 and dataset 2

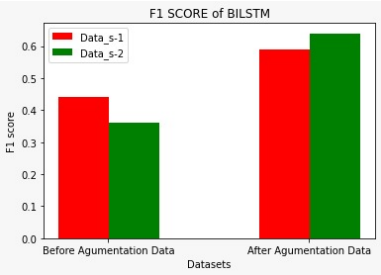

Figure 10: F1 score of BiLSTM ensemble model (dataset 1 and dataset 2

6 DISCUSSION

The ensemble model's classifiers all performed well due to their intensive training on datasets 1 and 2. Since the size is a critical feature for training the models and is reflected in the findings, 80% of the training data points were analysed for both datasets, suggesting that there is a substantial difference in the F1 score for both. According to the results, the CNN model had a very high F1-score of 0.79 on dataset 1 and 0.76 on dataset 2; the LSTM model was trained using the same datasets and had an F1-score of roughly 0.76 and 0.65, respectively, whereas the BiLSTM model, which was also trained using Word2vec, had an F1 score of 0.70 and

0.60 of the target's nearby related words were correctly identified. When considering the sentence's intricacy, this is also an enhanced and reasonable outcome. As seen in the results during the implementation of the models, the observed high score of all the instances of the data being trained on the CNN, LSTM and BiLSTM model indicated a possible issue of overfitting which is something that is resolved by the implementation of a callback for early-stop for CNN, and drop-out for LSMT and BiLSTM while monitoring the loss over the validation set.

The ensemble model's overall performance is extremely noteworthy because it was able to provide an F1 score of around 0.74 approximately for the test data of dataset 1 and 0.68 for another dataset. In terms of the outcome, Mathur et al. (2018) implemented a model, which utilised a hybrid strategy based on the CNN and LSTM and obtained an F1 score of 0.83, outperforming our suggested model. But, due to the inclusion of the third BiLSTM classifier, which was explicitly trained on the surrounding words to capture that semantic feature attribute for a text, our model was also able to understand the indirect annotators. Since the BiLSTM model can capture the utterance level comprehension of the text when predictions may be made while comprehending both the upcoming and the preceding word. Moreover, when the test results for the proposed ensemble model and the BiLSTM-ensemble model were compared, the BiLSTM model performed as predicted, with an average F1 score, since it is more suited for maintaining the connection and order within the phrase to grasp the context but pushed the word-level feature extraction into disarray. As an outcome, the combination of several classifiers enabled benefiting from the strengths of each model for ensemble learning. Consequently, the suggested model surely introduces a novel technique to confront the semantic comprehension of the model.

6.1 *Limitations*

The notion of an ensemble model, which includes merging many classifiers to get a prediction, is thought to be one of the most reliable methods for training a model with different perspectives and minimising variation, but it poses the issue of significant computational expenses. The model requires a lot of processing power to train and infer the output for the test sets. With the aim of reducing overfitting in the dataset while having limited capabilities, bootstrap aggregation, also known as bagging, is frequently employed to minimise variance and increase stability. However, when applied to real-world data, the model has a tendency to overfit the dataset it was initially trained on, which poses the issue of loss of interpretability when the model is applied to generalise on the other dataset.

Furthermore, it is important to recognise that compressing and transferring learned knowledge from a large model to a smaller model via knowledge distillation, an intuition for ensemble learning, has a profound impact on the model's capacity, which may not be completely utilised. As a result, it's possible that using only one model, which is more performant, may allow us to attain more stability and resilience than using an ensemble model that uses the output of singular models as input.

In addition, because of Hinglish's rich morphological structure and distinctive spelling patterns, it has a more complicated writing style than the preponderance of code-mix languages. The amount of time required for cleaning, preprocessing, and reducing input text noise is quite high. The dataset contains a very little amount of data and is frequently quite inconsistent which makes preprocessing a crucial step for the sentimental classification since it enables the effective utilisation of the lexicon that is already accessible but still difficult to handle. Hence, to determine if our conclusions are still valid, more studies using more reliable data that has been annotated by qualified experts are required.

6.2 *Societal Impact*

The study has evaluated different NN[10]-based ensemble learning models for the purpose of predicting the sentiment in textual data for hate speech. The ensemble model was trained on the augmented dataset to provide insight into the classification rate to accurately predict the polarity of the Hinglish tweets. The social impact is that the implementation of this robust classifier for social media networks where the influx of Hinglish users is high can be implemented to prevent anyone from getting abused publicly.

Additionally, this research is not only focused on the Hindi-English code-mix or towards only hate speech detection, but the model can be generalised beyond the scope of this study for any other multilingual language or text description with respect to business context to conduct the analysis. However, more research is needed to be done for an even better model that predicts the sentiment even more accurately for hate speech where the dataset is inherently imbalanced. Our most significant discovery is that we can create a reliable sentiment classification model that will consider the entire context of the phrase regardless of the existence of any profanity at the word level.

[10] Neural Networks

6.3 *Future work improvements*

One viable area for future study is to determine whether the ensemble model's performance can be improved by including a transformer learning-based model, like BERT or RoBERTa [11]. A modern architecture called transformer models tries to solve problems in a non-sequential way while addressing long-distance relationships with ease. Unlike BiLSTM, which has a very small vector window size as its aim. The approach is also reported to require less processing power to operate.

For improved outcomes, especially for the suggested model, another strategy of augmenting just the training dataset available should be a more appropriate approach to be considered in future work. Additionally, it is important to create a specialised, sizable Hinglish-based corpus that is aimed at improved word embedding. According to the investigation, the Hinglish term does not have a specific library. Therefore, the creation of new corpora for Hinglish will be extremely beneficial for any further study of code-mix text found on social media sites.

Hence, to determine if our conclusions are still valid, more studies using more reliable data that has been annotated by qualified experts are required.

7 CONCLUSION

The goal of this work is to develop a robust sentiment classification model that can correctly categorise tweets written in Hinglish with profanity. Regardless of the existence of profane words, we are interested in sentiment analysis that takes into account the context of the entire phrase. Hate speech tweets are not available to the general public since Hinglish is a language with low data resources and because of Twitter's policy against hateful conduct. We used a data augmentation strategy based on back translation and synonym replacement to increase the volume of the dataset in order to extensively train.

To summarise, RQ1, CNN model for word-level classification, LSTM for sentence-level classification, and BiLSTM for surrounding associated target words were tested separately on a labelled test set to check each model's performance. All three models were successful in predicting the polarity of the sentence. The ensemble model employed all three models concurrently to handle input and output for the observed pattern with an F1 score of 0.74. The most effective method for word embedding using

[11] Robustly Optimized BERT Pre-training Approach

Word2vec (skip-gram or CBOW) and BoW was evaluated beforehand in order to convert the text-to-numeric data for a better performance rate. The results of the ensemble model were then compared to the BiLSTM baseline model to examine for any improvement with only one classifier, as opposed to three classifiers trained for various semantic factors.

Finally, it is determined there is a variation in the sentiment prediction before and after training the ensemble model, first on low-volume data, and then on a sizably big training dataset. Hence, the study concludes that there is a lot of room for improvement in hate speech identification utilising ensemble learning models, but additional research using alternative deep-learning or transformer-based models should be done.

REFERENCES

Agarwal, K., & Narula, R. (2021). Humor generation and detection in code-mixed hindi-english. In *Proceedings of the student research workshop associated with ranlp 2021* (pp. 1–6).

Agarwal, V., Rao, P., & Jayagopi, D. B. (2021). Towards code-mixed hinglish dialogue generation. In *Proceedings of the 3rd workshop on natural language processing for conversational ai* (pp. 271–280).

Ananny, M., & Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*, *20*(3), 973–989.

Ardabili, S., Mosavi, A., & Várkonyi-Kóczy, A. R. (2020). Advances in machine learning modeling reviewing hybrid and ensemble methods. In *International conference on global research and education* (pp. 215–227).

Babanejad, N., Agrawal, A., An, A., & Papagelis, M. (2020). A comprehensive analysis of preprocessing for word representation learning in affective tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5799–5810).

Bansal, K. (2020). *Semeval-2020 task 9*. https://github.com/keshav22bansal/BAKSA_IITK/tree/master/data/hinglish. GitHub.

Biradar, S., Saumya, S., et al. (2022). Fighting hate speech from bilingual hinglish speaker's perspective, a transformer-and translation-based approach. *Social Network Analysis and Mining*, *12*(1), 1–10.

Bohra, A., Vijay, D., Singh, V., Akhtar, S. S., & Shrivastava, M. (2018, June). 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). New Orleans, Louisiana, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W18-1105 doi: 10.18653/v1/W18-1105

Bueno, I., Carrasco, R. A., Ureña, R., & Herrera-Viedma, E. (2022). A business context aware decision-making approach for selecting the most appropriate sentiment analysis technique in e-marketing situations. *Information Sciences*, *589*, 300–320.

Chakravarthi, B. R., Priyadharshini, R., Muralidaran, V., Jose, N., Suryawanshi, S., Sherly, E., & McCrae, J. P. (2022). Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, 1–42.

Desai, A., Zumbo, A., Giordano, M., Morandini, P., Laino, M. E., Azzolini, E., ... others (2022). Word2vec word embedding-based artificial

intelligence model in the triage of patients with suspected diagnosis of major ischemic stroke: A feasibility study. *International Journal of Environmental Research and Public Health*, *19*(22), 15295.

Duong, H.-T., & Nguyen-Thi, T.-A. (2021). A review: preprocessing techniques and data augmentation for sentiment analysis. *Computational Social Networks*, *8*(1), 1–16.

Feldman, I., & Coto-Solano, R. (2020). Neural machine translation models with back-translation for the extremely low-resource indigenous language bribri. In *Proceedings of the 28th international conference on computational linguistics* (pp. 3965–3976).

Gonçalves, C. A., Vieira, A. S., Gonçalves, C. T., Camacho, R., Iglesias, E. L., & Diz, L. B. (2022). A novel multi-view ensemble learning architecture to improve the structured text classification. *Information*, *13*(6), 283.

Gupta, N., & Agrawal, R. (2020). *Chapter 1-application and techniques of opinion mining.* Hybrid Computational Intelligence for Pattern Analysis and Understanding ....

Gupta, V., Sehra, V., Vardhan, Y. R., et al. (2021). Ensemble based hinglish hate speech detection. In *2021 5th international conference on intelligent computing and control systems (iciccs)* (pp. 1800–1806).

Jahan, M. S., & Oussalah, M. (2021). A systematic review of hate speech automatic detection using natural language processing. *arXiv preprint arXiv:2106.00742*.

Jain, D., Kumar, A., & Garg, G. (2020). Sarcasm detection in mash-up language using soft-attention based bi-directional lstm and feature-rich cnn. *Applied Soft Computing*, *91*, 106198.

Juwita, E. T., Effendi, A. Z., & Pandin, M. G. R. (2021). The effect of anonymity on twitter towards its users based on derek parfit's personal identity theory.

Kodali, P., Sachan, T., Goindani, A., Goel, A., Ahuja, N., Shrivastava, M., & Kumaraguru, P. (2022). Precogiiith at hinglisheval: Leveraging code-mixing metrics & language model embeddings to estimate code-mix quality. *arXiv preprint arXiv:2206.07988*.

Li, Y., & Yang, T. (2018). Word embedding for understanding natural language: a survey. In *Guide to big data applications* (pp. 83–104). Springer.

List, N. (2022). How can we investigate ancient greek categories without the influence of our own? exploring kinship terminology using word2vec. *International Journal of Lexicography*, *35*(2), 137–152.

Liu, B. (2020). Text sentiment analysis based on cbow model and deep learning in big data environment. *Journal of ambient intelligence and humanized computing*, *11*(2), 451–458.

Liu, G., & Guo, J. (2019). Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing*, *337*, 325–338.

Mathur, P., Sawhney, R., Ayyar, M., & Shah, R. (2018, October). 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). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W18-5118 doi: 10.18653/v1/W18-5118

Pratapa, A., Choudhury, M., & Sitaram, S. (2018). Word embeddings for code-mixed language processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3067–3072).

Pravalika, A., Oza, V., Meghana, N. P., & Kamath, S. S. (2017). Domain-specific sentiment analysis approaches for code-mixed social network data. In *2017 8th international conference on computing, communication and networking technologies (icccnt)* (p. 1-6). doi: 10.1109/ICCCNT.2017.8204074

Rajeswari, A., Mahalakshmi, M., Nithyashree, R., & Nalini, G. (2020). Sentiment analysis for predicting customer reviews using a hybrid approach. In *2020 advanced computing and communication technologies for high performance applications (accthpa)* (pp. 200–205).

Santosh, T. Y., & Aravind, K. V. (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* (p. 310–313). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3297001.3297048 doi: 10.1145/3297001.3297048

Sawhney, R. (2018). *Offensivehinglishtweetclassification.* https://github.com/ramitsawhney27/OffensiveHinglishTweetClassification/blob/master/HOT_Dataset_modified.csv. GitHub.

Sengupta, A., Bhattacharjee, S. K., Akhtar, M. S., & Chakraborty, T. (2022). Does aggression lead to hate? detecting and reasoning offensive traits in hinglish code-mixed texts. *Neurocomputing*, *488*, 598–617.

Shahid Ul Islam, D., & Sharma, P. (2021). An efficient machine learning approach for twitter sentimental analysis for low classification error rates.

Shalini, K., Ravikurnar, A., Reddy, A., Soman, K., et al. (2018). Sentiment analysis of indian languages using convolutional neural networks. In *2018 international conference on computer communication and informatics (iccci)* (pp. 1–4).

Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text data augmentation for deep learning. *Journal of big Data*, *8*(1), 1–34.

Srivastava, V., & Singh, M. (2020). Phinc: A parallel hinglish social media code-mixed corpus for machine translation. *arXiv preprint arXiv:2004.09447*.

Srivastava, V., & Singh, M. (2021). Challenges and limitations with the metrics measuring the complexity of code-mixed text. *arXiv preprint arXiv:2106.10123*.

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Swamy, S., Kundale, J., & Jadhav, D. (2022). Sentiment analysis of multilingual mixed-code, twitter data using machine learning approach. In *International conference on innovative computing and communications* (pp. 683–697).

Tahata, K. (2022). Advances in quasi-symmetry for square contingency tables. *Symmetry*, *14*(5), 1051.

Tariku, W., Meshesha, M., Hunegnaw, A., & Lemma, K. (2022). Sentiment mining and aspect based summarization of opinionated afaan oromoo news text. *American Journal of Embedded Systems and Applications*, *9*(2), 66–72.

Thakur, V., Sahu, R., & Omer, S. (2020). Current state of hinglish text sentiment analysis. In *Proceedings of the international conference on innovative computing & communications (icicc).*

Tho, C., Heryadi, Y., Kartowisastro, I. H., & Budiharto, W. (2021). A comparison of lexicon-based and transformer-based sentiment analysis on code-mixed of low-resource languages. In *2021 1st international conference on computer science and artificial intelligence (iccsai)* (Vol. 1, pp. 81–85).

Vashistha, N., Zubiaga, A., & Sharma, S. (2020). An online multilingual hate speech recognition system. *arXiv preprint arXiv:2011.11523*.

Yan, D., Li, K., Gu, S., & Yang, L. (2020). Network-based bag-of-words model for text classification. *IEEE Access*, *8*, 82641–82652.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, *28*.