CROSS-LINGUAL HATE SPEECH DETECTION

By

Samrudhi Keluskar

A thesis
submitted to Liverpool John Moores University
in partial fulfillment of the requirements for the degree
of
Master of Science in Data Science

December 2021

DEDICATION

This thesis is dedicated to my husband, who has been a constant source of support and motivation throughout my master's program and life challenges. I am very grateful to have him in my life. This work is also dedicated to my parents, who have always loved me unconditionally and encouraged me to work hard for the things that I aspire to achieve.

ACKNOWLEDGEMENTS

I would like to thank my supervisors, Mr. Shah Mohammad Azam, Dr. Manoj Jayabalan and Dr. Rupal Bhargava for all their help and invaluable advice throughout my thesis. I would also like to extend my sincere thanks to Miss Kriti Mamgain. It is their kind help and support that made my study effortless. Finally, I would like to express my sincere gratitude to my husband, parents and friends for their encouragement and immense support throughout my studies.

ABSTRACT

On social media, racist and sexist remarks are common forms of hate speech. Hate speech consists of a very wide range of expressions that promotes or justifies hatred, violence, and discrimination against an individual or group of individuals for a variety of reasons. This research aims on detecting hate speech in different languages. This work includes literature review of recent research papers in this domain. The data was gathered from reputable sources such as Kaggle, Hugging Face, and GitHub. The data contains texts labeled as hate or non-hate in languages like English, Hindi, Portuguese, Chinese, Danish, Arabic, Turkish, and Greek. Since English is the most extensively used language on social media, the labels of this dataset were cleaner. Hence, English language data, in conjunction with a multi-lingual BERT based large language model, was used to train and test the ML models. The best performing model was then tested on the multi-lingual data. The model performed the best on English and Portuguese test data. It also performed fairly well on the rest of the languages.

BERT, a text vectorization method for vectorizing texts in different languages is used and explained in this work. Classification algorithms like Logistic Regression, Random Forest, Gradient Boosting, K-Nearest Neighbors Support Vector Machines, Neural Network that were used to classify the text into hate vs non-hate are summarized in this report. The purpose of this study is to identify hate speech on social media and internet platforms so that measures can be taken to prevent the spread of hatred. The identification and segmentation of hate speech on social media platforms will contribute to the reduction of hatred and violence directed at specific groups.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUC-ROC…… Area Under the Curve-Receiver Operating Characteristic Curve

BERT…… Bidirectional Encoder Representations from Transformers

BiLSTM…… Bidirectional Long Short-term Memories

CNN…….. Convolutional Neural Networks

CV……. Cross-Validation

EDA……... Exploratory Data Analysis

FN…….. False Negative

FP……... False Positive

KNN……... K-Nearest Neighbor

LSTM……. Long Short-Term Memories

NLP…… Natural Language Processing

SVM……... Support Vector Machines

TF-IDF……..…. Term Frequency-Inverse Document Frequency

TN…….. True Negative

TP……... True Positive

TPR……..... True Positive Rate

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background of the Study

Hate speech refers to any type of expression that is intended to humiliate, criticize, or instigate hatred towards a group or a class of people based on race, religion, skin color, sexual identity, gender identity, ethnicity, disability, or national origin. In recent years, there has been a boom in interest in detecting abusive language, hate speech, cyberbullying, and trolling (Schmidt and Wiegand, 2017). Social media platforms have been under increasing pressure to address these issues. Because of the similarities between these subtasks, scholars have grouped them under the umbrella titles "abusive language," "harmful speech," and "hate speech.".

Hate speech is protected in the United States under the First Amendment's free speech provisions, but that is disputed in the legal body and in relation to speech codes on university campuses. Hate speech is a felony in several countries, including the United Kingdom, Canada, and France. Hate speech is defined as remarks directed at minorities in a manner that may end in violence or social disorder. Individuals who are guilty of employing hate speech must pay hefty penalties and may even face prison time. These regulations apply to online sites, prompting online sites to implement anti-hate speech policies of their own. In response to criticism that they are not adequately helping in preventing hate speech on their platforms, Facebook and Twitter have implemented policies prohibiting the usage of their websites for attacking people based on race, ethnicity, gender, or sexual orientation, as well as violent threats against others.

In extreme circumstances, this could include language that terrorizes or promotes violence, but if we only consider this definition then we would leave out a significant amount of hate speech. Importantly, the definition here excludes most of the occurrences of objectionable language because individuals frequently use expressions that are very offensive to specific individuals but are used in fundamentally many ways. For instance, some African Americans frequently use the term n*gga in a common online conversation (Warner and Hirschberg 2012). When quoting rap lyrics, rappers use phrases like h*e and b*tch, and young video game players commonly used homophobic slurs like f*g. On social media, this kind of language is common (Wang et al. 2014). As a result, boundary conditions are essential for any hate speech detection system.

The purpose of this research work is to build a classification model that can detect hate speech on different online platforms. These platforms can then take actions to prevent the spread of hatred on their platforms. Overall, this can help reduce violence and hate in society.

1.2 Aim and Objectives

The key aim of this research work is to propose a model to predict if a text input consists of hate speech or not. The goal of this research study is to recognize hate speech on social media sites such as Twitter so that action can be taken to prevent the spread of hate.

The following research objectives are framed based on the aim of this study:

- To preprocess the dataset by cleaning and vectorizing the text
- To propose multiple text classification models for detecting hate speech
- To evaluate and compare the performances of the implemented algorithms in order to determine the best text classification model

1.3 Significance of the Study

The research is contributing to detecting hate speech in different languages using predictive classification models. This research can be used to identify hate speech which then can help in implementing strict measures to prevent the spread of hatred. Knowing which users are the root cause of spreading hatred aids in blocking such accounts. It is critical to put a stop to online hate because it has the potential to incite violence. Social media and internet platforms are the beneficiaries of this research. This study will help these platforms detect hate speech and take appropriate measures to stop hate on their platform. The detection and removal of hate text from social media platforms will aid in reducing hate and violence against certain groups.

1.4 Scope of the Study

Due to the time restrictions, the scope of the research study is restricted as below:

- The hate speech detection models for this study are built using the data in English and the best performing model is tested on languages like Hindi, Portuguese, Chinese, Danish, Arabic, Turkish, and Greek
- Text vectorization is done using BERT
- A small set of machine learning algorithms like Logistic Regression, Random Forest, Gradient Boosting, K-Nearest Neighbors Support Vector Machines, Neural Network are used for this study

1.5 Structure of the Study

There would be six main chapters in this research thesis. The next five chapters' topic matter is described in this section.

Chapter 2: Consists of literature review of the latest papers in the hate speech domain.

Chapter 3: Consists of research methodology to be used for building prediction models to predict hate/non-hate speech. This chapter describes in detail steps like data collection and data cleaning, EDA, vectorization, predictive modeling techniques, evaluation metrics and required resources.

Chapter 4: Consists of analysis of the data after preprocessing, EDA, analysis of hate proportion, and hyperparameter tuning analysis.

Chapter 5: Consists of model results and discussion of the results. This chapter mentions the best performing model and the test data languages on which the model’s performance is outstanding.

Chapter 6: Consists of the final conclusion of the research work. This chapter also describes the limitations and contribution of this research work. Provides suggestions for future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

It has become crucial for all social media and internet platforms to keep their sites clean. Several researchers are working towards detecting hate speech on the internet. Following are few works that are done in this field.

2.2 Related Research

(Davidson et al., 2017) have mentioned the use of a crowd-sourced hate speech set of terms to gather tweets that contain hate speech keywords. A sample of these tweets was labeled into three classes: hate speech, offensive language, and neither using crowdsourcing. They first stemmed the tweets and then created unigram, bigram and trigram which were then processed through TF-IDF. They had trained multiple different models and found that Logistic Regression and Linear SVM tended to perform notably better than other models. The top-performing model had an overall precision of 0.91, recall and F1 score 0.90. They discovered that racist and homophobic tweets were more likely to be labeled as hate speech, whereas sexist tweets were frequently classified as offensive. It was difficult to classify tweets without explicit hate keywords.

In the work of (Gaydhani et al., 2018), they presented a method for categorizing tweets on Twitter into three categories: hateful, offensive, and clean. They experimented with n-grams as features and passed their term frequency-inverse document frequency (TF-IDF) values to several machine learning models using a Twitter dataset. They looked at three popular text classification machine learning algorithms: Logistic Regression, Naive Bayes, and Support Vector Machines. They attained 95.6 percent accuracy on test data after fine-tuning the logistic model that produced the best results. 4.8 percent of the offensive tweets were misclassified as hateful, according to the findings. More instances of offensive language that does not contain hateful terms can be obtained to remedy this problem.

16 different sources to undertake a broad evaluation of multilingual hate speech in 9 different languages were used by (Aluru et al., 2020). They discovered that basic models like LASER-based features with logistic regression perform best in low-resource settings, while BERT-based models perform better in high-resource settings. When it comes to zero-shot classification, languages like Italian and Portuguese perform well. Their proposed architecture could be useful for languages with limited resources. These models could also be used as a starting point for future multilingual hate speech detection projects.

The task to classify a tweet as racist, sexist, or neither is difficult due to the intricacy of natural language constructs. To deal with this complexity, (Badjatiya et al., 2017) conducted extensive experiments with several deep learning architectures to learn semantic word embeddings. They investigated using deep learning techniques to the task of detecting hate speech. They looked at char n-grams, word Term Frequency Inverse Document Frequency (TF-IDF) values, Bag of Words Vectors (BoWV) over Global Vectors for Word Representation (GloVe), and task-specific embeddings learned with FastText, CNNs, and LSTMs. The researchers found that deep learning methods outperform state-of-the-art char/word n-gram methods by 18 F1 points on a benchmark dataset of 16K annotated tweets.

Deep Neural Network structures as feature extractors, which are good at apprehending hate speech semantics were developed by (Zhang and Luo, 2018). Their methods surpass the highest performing method by up to 5 percentage points in macro-average F1 and 8 percentage points in the more difficult situation of recognizing hostile material. For evaluation, they used the conventional Precision, Recall, and F1 metrics.

(Waseem and Hovy, 2016) offered a list of critical race theory-based criteria, which they utilized to label a public corpus of over 16k tweets. For each tweet and the user description, they gather unigrams, bigrams, trigrams, and four grams. They ran a grid search over all relevant feature set combinations to find that character n-grams outperformed word n-grams by at least 5 F1-points. To assess the influence of various features on prediction performance and quantify their expressiveness, they utilized a logistic regression classifier and 10-fold cross-validation. They studied the impact of several non-linguistic factors in combination with character n-grams in

detecting hate speech. They also provide a lexicon based on the most relevant words found in their data.

Annotating big sets of data is exceedingly difficult due to the context-dependent nature of online aggressiveness. Previously researched datasets in abusive language detection were too small to build deep learning models efficiently. Hate Speech on Twitter, a considerably larger and more reliable dataset, was recently released. However, the full potential of this dataset has yet to be discovered. (Lee et al., 2018) conducted the first comparison research of several models on Hate and Abusive Speech on Twitter in this paper and highlighted the possibilities of incorporating more characteristics and context data to improve the model. Experiments demonstrate that bidirectional GRU networks with Latent Topic Clustering modules, trained on word-level features, are the best accurate model, scoring F1 of 0.805.

Hate speech data made up of thousands of words that were manually classified as containing or not containing hate speech was introduced by de (Gibert et al., 2018). The sentences were taken from Stormfront, a white supremacist discussion board. To complete the manual labeling work, a custom annotation tool was created, which allows annotators to choose whether to examine the context of a statement before labeling it. Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Recurrent Neural Networks with Long Short-term Memories were among the algorithms tested. Individual accuracy is shown in the results for hate and no-hate, as well as overall accuracy. In addition, the report includes a thorough qualitative and quantitative analysis of the produced dataset, as well as multiple baseline tests with various categorization models.

(Mathew et al., 2020) introduced HateXplain, a benchmark hate speech dataset addressing several elements of the topic in this research. Every post in their dataset is explained in three ways: the basic, frequently used three-class classification (hate, offensive, or neither), the target group (the set of people who are hate speech victims in the post), and the reasons, i.e., the segment of the post on which their labeling decision (as hate, offensive or normal) is based. They used current state-of-the-art models and discovered that models which work well in classification do not do well on explainability criteria such as model plausibility and faithfulness. CNN-GRU, BiRNN, and BERT are some of the models they've utilized. Accuracy, macro F1-score, and AUC-ROC score were among the parameters they reported.

English data was used by (Ranasinghe and Zampieri, 2020) to create predictions in languages with fewer resources by using cross-lingual embeddings and transferred learning. They made predictions based on similar data in Spanish, Bengali and Hindi reporting F1 macro values of 0.84 for Bengali, 0.85 for Hindi, and 0.75 for Spanish. Finally, they demonstrated that their method outperforms the best systems in recently shared tasks in these three languages, demonstrating the durability of cross-lingual transfer learning for this task.

(Vijayaraghavan et al., 2021) proposed a deep learning model in this paper that can (a) identify hate speech by efficiently capturing the text semantics as well as the socio-cultural context in which a certain hatred expression is generated, and (b) deliver explainable insights into our model's decisions. They employed baseline deep learning models as well as traditional models. Deep learning models clearly outperform classical models, according to their findings. They demonstrated that their model outperforms existing state-of-the-art hate speech classification algorithms by conducting a thorough review of alternative modeling strategies. Finally, they demonstrate the significance of social and cultural environment factors in identifying clusters linked with various types of hate.

(Plaza-del-Arco et al., 2021) tackled the problem of detecting hate speech in Spanish on social media in this research and provided a greater grasp of the possibilities of new machine learning approaches. Emojis, mentions, hashtags, elongated words, phrases, and URLs occur in the content, making tokenization challenging. They evaluated Deep Learning's performance to that of newly transfer learning-based pre-trained language models, as well as conventional machine learning models. Their key offering is the use of multilingual and monolingual pre-trained language models like BERT, XLM, and BETO to obtain promising outcomes in Spanish.

(Pamungkas et al., 2021) investigated hate speech identification in data-scarce languages by using zero-shot learning to transfer knowledge from a data-rich language, English. They tested a variety of neural architectures and suggested different models that use various cross-lingual language embeddings to transfer knowledge between languages. They also used information from a cross-lingual lexicon of abusive expressions to assess the influence of additional knowledge in their experiment. Their joint-learning models outperformed others in most languages, according to the

findings. An easy strategy that utilizes machine translation and a pre-trained English language model, on the other hand, provides a reliable result. Multilingual BERT, on the other hand, did not perform well in cross-lingual hate speech detection.

(Kamble and Joshi, 2018) investigated hate speech identification in English-Hindi code-mixed tweets in this paper. They employed three common deep-learning models to identify hate speech (CNN-1D, LSTM, and BiLSTM) and empirically verified their usefulness. Their models were not only good at obtaining the semantics of hate speech but also its context. They also proved the effectiveness of domain-specific word embeddings in bringing intrinsic value to the code-mixed environment. Their research uses a benchmark dataset to demonstrate how deep learning models can improve on well-known statistical classifiers work.

(Kapil' et al., 2020) suggested deep learning algorithms for detecting several sorts of hate utterances online in this research, which used diverse embeddings. They used CNN, LSTM, BiLSTM, and CharacterCNN to create 13 deep learning models. Identifying hate speech from a huge proportion of text, particularly tweets with insufficient contextual information, presents a number of practical difficulties. Their studies on three publicly accessible datasets from various domains reveal that accuracy and F1-score have improved significantly. For all three datasets, LSTM and BiLSTM based on recurrent neural networks scored best.

ETHOS hate speech detection dataset was used by (Rajput et al., 2021) in this paper and tested the classifier's performance by integrating the word embeddings (fastText (FT), GloVe (GV), or FT + GV) with static BERT embeddings (BE). After considerable testing on multiple deep neural networks, it was discovered that utilizing neural networks with static BERT embeddings can remarkably improve the model's overall performance, particularly in terms of specificity, showing that the model is great at correctly detecting non-hate speeches. This means that it identifies less-harmful non-hate speech as hate speech, which in return means protecting the idea of free speech. They also stated that with such a promising outcome, the same theory of combining static BERT embeddings with state-of-the-art models can be expanded to other natural language processing-based applications.

(Plaza-del-Arco et al., 2021) addressed the job of identifying Spanish language hate speech on social networks in this paper, as well as give a clearer picture of the ability of the latest machine learning-based techniques. They specifically compare Deep Learning techniques to recently pre-trained language models based on transfer learning as well as conventional ML models. Their important contribution has been to achieve promising results in Spanish using multilingual and monolingual pre-trained language models like BERT, XLM, and BETO.

(Miok et al., 2021) proposed a Bayesian method based on Monte Carlo dropout within the transformer models' attention layers to provide well-calibrated reliability estimates. They evaluated and visualized the results of the proposed approach to hate speech detection problems in several languages. They also investigated whether affective dimensions could improve the information extracted by the BERT model in hate speech classification. Their experiments show that Monte Carlo dropout is a viable mechanism for estimating transformer network reliability. It provides cutting-edge classification performance and can detect less trusted predictions when used in conjunction with the BERT model.

To detect hate speech, (Jiang and Zubiaga, 2021) proposed a translinguistic capsule network learning model (CCNL-Ex) that incorporates additional hate-related semantic features. CCNL, the model's main framework, is made up of two parallel architectures for source and target languages, with BiLSTM extracting contextual features and Capsule Network capturing hierarchical positional relationships. Finally, when compared to ten competitive benchmarks, your model outperforms all six language pairs. The findings show the importance of learning contextual information and spatial relationships in hate speech texts. Because using machine translation resources such as Google Translate is not always the best option when dealing with very informal language, such as that used to incite hatred on social media, they investigate culture-based approaches that take the context into account. In future works, there will be a parallel corpus. Extensive implementations for additional languages and data sets are also desired in order to assess your model's generalizability.

The main challenges were first identified in this study by (Qureshi and Sabih, 2021), and the complex problem of the automated classification of hate messages of several classes in the text

was solved with much better results. Ten separate binary classified datasets were created that contain several categories of hate speech. The data sets were balanced and diverse. (Qureshi and Sabih, 2021) have also been enriched with linguistic aspects. In order to create high-quality datasets, a list of effective, commonly used and recommended features from related text mining studies has been identified. These characteristics have been studied and identified in relation to the objective of the problem. They have found 2-4 gram characters, 1-5 gram words, dependency tuples, sentiment scores, and collecting 1st and 2nd person pronouns to be very effective. Latent Semantic Analysis (LSA) has also been used as a dimensionality reduction algorithm and has been shown to be very effective in dimensional classification problems of high tan. The datasets were examined in depth using tSNE multi-dimensional plots. These graphs will reveal issues such as lack of suitable discriminatory features, complex data overlaps, and non-linearity. As a result, complex and nonlinear models were used for classification, and the most popular and advanced machine learning model, CAT Boost, performed best in all datasets. CAT Boost to the average plus high scores in accuracy, F1 and AUC, with 89.03, 87.74 and 88.88, respectively. Likewise, the Gradient Boosting model performed similarly to CAT Boost with smaller differences, scoring 88.78, 86.04, and 87.69 under the same precision measurements, F1, and AUC, respectively. Random Forest was in third place with slightly different scores of 86.45, 85.53 and 86.76 for Precision, F1 and AUC, respectively. The final model performance is also comparable to that of related studies and their initial baseline. It should be noted that the model outperforms them all.

On Twitter and Gab datasets, (Das et al., 2021) used semi-supervised and supervised machine learning algorithms to detect hateful users. GNNs that take advantage of both textual features and user social connections significantly perform better than other models; the finest model attains a macro F1-score of 0.780 on Twitter and 0.791 on Gab while utilizing only 5% of labeled data. In order to better comprehend the models, they carried out a thorough error evaluation on AGNN and doc2vec, which are the greatest performing models utilizing text+network and text features, respectively. They discovered that doc2vec performs poorly when the number of hateful words in a user's post is small. In such instances, a user's neighborhood can help the AGNN model make more accurate predictions. They also notice that in a zero-shot setting, structural signatures learned from a network can be transferred to an unknown dataset. They conducted a rigorous post-facto analysis to determine how hateful posts and hateful users target various sections of society.

Automatic hate speech recognition on social networks is a critical job that ensures users have equal access to online platforms. (Agarwal and Chowdary, 2021) emphasized the task's significance during the global COVID19 outbreak when users were spending more hours online and covid related information was getting more sensitive. They have demonstrated, using facts, that there is a swift growth in hateful comments on digital social networks, precisely targeting a group or a country and its people. They have also highlighted the significance of this study during the US elections. They reported the sincerity with which social network sites seek to boost their algorithms for identifying hateful speech with zero need for manual investigation. To mark these concerns, they suggested a model for the task. Neural network techniques for extracting features were combined with an ensemble-based adaptive classifier to predict the target variable of the tweets in their proposed model. They conducted experiments on standard datasets as well as the most recent data on the US elections and COVID-19 and recorded the results under various experimental conditions. They also investigated and showed the flaws in the available datasets; as a result, they aimed to create a finer-grained dataset free of user overfitting in the future.

(Zhou et al., n.d.) investigated the effectiveness of multitasking learning in hate speech detection tasks in this article. The main idea was to use multiple function extraction units to share multitasking parameters so that the model can better share sentiment knowledge and then merge functions for hate speech detection using closed attention. The proposed model can fully utilize the target's sentiment information, as well as external opinion resources. They have shown that sharing knowledge of feelings improves system performance compared to baselines and promotes hate speech detection. Finally, a detailed analysis validates and interprets our model. In general, their experiences provide information on the relationship between detecting hate speech and analyzing sentiments through multitasking learning.

In this paper, (Ahmed et al., n.d.) presented the principles of three types of text classification methods, ELMo, BERT, and CNN, and used them to detect hate speech, then boosted the model performance by fusing the methods: the fusion of BERT, ELMo, and CNN classification results, and the fusion of 3 CNN classifiers with different parameters. The findings demonstrated that fusion processing is a feasible method for improving the performance of the model. It is possible to achieve the practical significance of performance at a small additional cost. This paper focuses on the fusion following separate classification; the degree of integration is insufficient. In the

future, they will place a greater emphasis on early collaboration prior to classification. They will attempt to replace the basic word vector expression in CNN with BERT or ELMo embedding technologies. This can deeply combine the benefits of excellent word embedding and powerful neural networks.

(Pronoza et al., 2021) aimed to detect ethnicity-targeted negative comments, studying hate speech, on Russian social networks in this research. To accomplish this, they produced the RuEthnoHate dataset, which contains texts that contain various Russian ethnic classes, and made annotations using the corpus. The annotation included three classes: positive, neutral and negative attitudes toward ethnic groups, with the latter implying hate speech directed at ethnic groups. They conducted hate speech recognition experiments using text-level binary attitude detection (BAD) and trinary instance-based attitude detection (IBAD) approaches, as well as classical machine learning and deep learning models. Simple unigrams, Word2vec trained on their large RuEthnics dataset (Word2vec-Ethno), the Russian National Corpus (Word2vec-RNC), and Conversational RuBERT embeddings were used for text representation (RuBERT-emb). Naive Bayes, SVM, Logistic Regression and their ensembles were the traditional machine learning models used. LSTMþGRU and Conversational RuBERT techniques were used to create deep learning models. The IBAD approach yielded the best results for them. Convers-RuBERT outperformed both traditional machine learning and the LSTMþGRU model.

In this paper, (Budi Herwanto et al., 2021) developed a prediction model for hate speech and abusive language based on a social media dataset. They used the word and contextual embedding approaches to provide a semantic representation of the tokens. They did not pre-process the data to demonstrate the robustness of the embeddings. In other words, using contextual embedding, the model can still work well without any pre-processing. As a result, they left this gap for future work. Furthermore, they used document recurrent neural network embedding to extract sentence sequence information. They recommended using stopword elimination, slang substitution, stemming, and other pre-processing techniques in the future. Due to the high noise in social media data, they believed that several pre-processing methods will improve the model's efficiency. They also recommend experimenting with new transformer models such as Bidirectional Encoder Representations (BERT) and Generative Pre-trained Transformers (GPT).

In this paper, (Mansourifar et al., 2021) presented a dataset gathered from an unexplored media source. They demonstrated that a voice-based social media platform, such as Clubhouse, has enormous potential to reveal a wide range of data on topics ranging from military to philosophy. It demonstrates that voice-based chat rooms can serve as a data collection center for researchers working in various fields of natural language processing. Candidates from 12 different nationalities participated in the debate about the latest Israel-Palestine dispute, according to self-identification data. They discovered that participants avoid using offensive language to express hate speech and instead try to talk in a civilized manner. That is why hate speech detection in Clubhouse is difficult. Using two different base classifiers, they tested three different feature extraction methods. In terms of all tested performance measures, their experimental results show that Google Perspective Scores perform better than traditional feature extraction approaches such as Bag of Words and Word2Vec.

(Vitiugin et al., 2021) developed a multilingual interactive attention network (MLIAN) model for identifying hate speech in social networks, regardless of language, in this paper. The central aim of MLIAN was to use 2 attention networks to interactively model the context of the text, where they used frame semantics theory to design a principled approach for appropriately guiding human feedback to provide target labels. They used simulated human feedback to identify the special tokens as target labels by labeling posts that include personal/group hatred. The model pays close attention to such important context elements and studies to pay more attention to potential semantic frame elements characterizing the hate speech in the post. Experiments on the SemEval-2019 dataset show that the MLIAN model outperforms several baselines and requires little human feedback to improve model performance. They presented extensive analyses to demonstrate the value of modeling with human feedback, which allows the model to be easily adapted to different languages and tasks. The use of the MLIAN model can help guide future research into multilingual hate speech analytics.

(Dhanya and Balakrishnan, 2021) focused on different approaches that are used for a NLP problem, specifically "hate speech detection in social media data." As a result, this paper presents several language-specific studies for automated hate speech recognition in Asian languages. As disrespectful texts became more common online, social media data analytics became a major

challenge among natural language problems. This was the impetus for them to conduct such a study in multiple languages. For this problem, various Machine Learning classification algorithms and Deep Learning algorithms were used. According to the findings of this study, SVM is a highly recommended algorithm for binary classification of this problem. The accuracy of each study, however, is determined by the type of dataset (balanced or unbalanced) and its size. According to the survey, accuracy increases with the size of the dataset. Some of the multi-label classification studies produced the best results in deep learning classification algorithms as well. The future of this study is an expansion of the survey to include global languages such as American and European languages. Researchers can gain a basic understanding of various approaches by using English.

Hate speech as a societal problem is an old research topic in the arts and humanities, but it is still a new topic in the computing domain. (Mullah and Zainon, 2021) investigated methods for detecting hate speech in social media using classical ML, Ensemble, and deep learning approaches. According to the findings of this study, there is a lot of research in hate speech recognition using classical machine learning techniques than ensemble and deep learning techniques. This means that researchers can conduct additional research on hate speech recognition using deep learning and ensemble methods. This article also identified some open challenges in hate speech detection, such as cultural differences, pandemic or natural disasters, data sparsity, imbalance dataset challenge, and dataset availability concerns. This research discovered that the current state-of-the-art does not address special characters and numeral symbols commonly used in Nigeria for constructing HS comments. In Nigeria, for example, the term "419" is commonly used to refer to unwholesome behavior. There has been no research into this.

(Ali et al., 2021) prepared a dataset for this study by collecting tweets in the Urdu language and had them labeled on aspect and sentiment levels by expert linguists. There was no Urdu hate speech dataset that has been annotated on aspect and sentiment levels before this research. They used cutting-edge techniques to address the 3 most common problems in ML-based sentiment analysis: sparsity, dimensionality, and class skew, and they observed a performance advancement over the base model. The classifier was trained using two ML algorithms, Multinomial Nave

Bayes and SVM. They used dynamic stop words filtering to reduce sparsity, variable global feature selection scheme to reduce dimensionality, and synthetic minority oversampling to reduce class imbalance (SMOTE). They used the micro F1 measure to compare performance to the baseline model. Their findings revealed that addressing class skew as well as the high dimensionality issue improves the overall performance of sentiment analysis-based hate speech detection the most. This research can be expanded by collecting more data from other social network sources and observing the outcomes of the approaches presented in this paper. Another thing that can be done to address the class skew issue is to include lexical scores of terms in the features set alongside the TF-IDF weights. Furthermore, their long-term goal is to solve the class imbalance problem for deep learning algorithms.

According to (Parihar et al., 2021), hate speech detection is a complex task that remains a societal issue. There is a fine line between what constitutes hate speech and what does not. A satire, for instance, may be regarded as a potential menace, but that is not hate speech. As a result, annotating and collecting data for the purpose of building a model for hate speech identification is a difficult task. As previously discussed, this issue can be resolved by narrowing the annotation criteria. Similarly, there is a need to concentrate research on code-mixed languages as well as regional languages. In the classification of hate speech, language models and deep learning models have shown promising results. Upsampling and downsampling techniques based on language models should be investigated for dealing with unbalanced data. The issues raised above must be addressed through additional research in the field so that the internet becomes more inclusive, welcoming, and free of hate.

(Zampieri et al., 2021) investigated a novel approach to designing a Hate Speech Detection system for short texts, such as tweets, in this work. They proposed adding multiword expression features to their DNN-based detection system. They used a three-branch neural network to integrate multi-word expression (MWE) features into a USE-based neural network. This network can consider both sentence-level and word-level features (USE embedding) (MWE categories and the embeddings of the words belonging to the MWEs). The findings were validated using two tweet corpora: HatEval and Founta. The models they suggested resulted in notable macro-F1 improvements over the baseline system (USE system). Furthermore, on the HatEval corpus, the

proposed system with MWE all categories and BERT embedding outperformed the state-of-the-art system FERMI, which placed first in the SemEval2019 shared task 5. These findings demonstrated that MWE features can be used to improve our baseline system. The proposed method can be applied to other NLP tasks such as sentiment analysis and automatic translation.

(Cinelli et al., 2021) used a model fine-tuned on a huge set of hand-annotated data to detect hate speech on a corpus of over one million comments on YouTube videos. According to their findings, there is zero proof of the presence of "serial haters," defined as active users who only post hateful comments. Furthermore, in line with the echo chamber hypothesis, they discovered that users who prefer one of the two types of video channels (questionable or reliable) are more likely to use improper, brutal, or hateful language within their opponents' community. Surprisingly, users who trust reliable sources use more toxic language than their counterparts. Finally, they discovered that the overall toxicity of the discussion increases with its length, as measured by both the number of comments and the length of time. Their findings show that, in accordance with Godwin's law, online debates tend to devolve into increasingly toxic exchanges of views.

(Markov and Daelemans, 2021) tested deep learning models in both in-domain and cross-domain hate speech recognition scenarios, and they present a Support Vector Machines approach that remarkably improves state-of-the-art results when combined with deep learning models via a simple majority-voting ensemble. The improvement is primarily due to a decrease in the false-positive rate. They demonstrated that combining deep learning models with a robust feature-engineered SVM approach can address one of the challenges in hate speech detection: erroneous false positive decisions. Within the in-domain and cross-domain settings, the results are consistent. This simple strategy significantly improves the results of state-of-the-art hate speech detection.

The global spread of social networks is increasing, which means that more and more people will be subjected to cyberbullying. Because of the pandemic, the popularity of online learning via the internet is increasing. Hateful messages must be automatically identified and blocked in order to protect children from cyberbullying. (Sultan et al., 2021) used ML models to identify

cyberbullying on the internet in this study. To accomplish this, they compiled a corpus of cyberbullying words in Kazakh, performed primary data processing and cleaning, and used ML algorithms to perform binary text classification. So far, they have detected cyberbullying with an accuracy of 87 percent. In the future, it is planned to apply Kazakh language features, replenish the corpus, consider undersampling and oversampling techniques for unbalanced data, and build a deep learning model for identifying cyberbullying.

(Perifanos and Goutsos, 2021) investigated the issue of hate speech on social medial, precisely racist speech against refugees in Greek, in this paper. Their approach is in line with recent trends in NLP and suggests that combining visual and textual modalities in a late-fusion multimodal learning setting can improve overall detection accuracy. As part of this effort, they have made a RoBERTa-based Language Model trained on Greek tweets publicly available. Their NLP models were created with the transformers python library, and their computer vision models were created with PyTorch. It should be noted that their study is limited to single tweets only. One promising research avenue would be to combine the multimodal learning approach with social-graph information in a single framework, essentially combining Graph Neural Networks with multimodal representations. Another approach would be to look into pre-trained convolution models rather than transformer-based ones, as recent research indicates that CNN can outperform Transformers in certain tasks.

2.3 Summary

Most of the data used in the above research papers comes from Twitter. (de Gibert et al., 2018) used data from Stormfront, a white supremacist discussion board. (Aluru et al., 2020) used 16 different sources to undertake a huge-scale investigation of multilingual hate speech in nine different languages. (Mansourifar et al., 2021) have used data from a voice-based social media platform called Clubhouse. (Das et al., 2021) made use of data from Gab, an American social networking service that is well known for its far-right user base. (Cinelli et al., 2021) used hand-annotated data to detect hate speech on a corpus of over one million comments on YouTube videos.

Emojis, mentions, hashtags, elongated words, phrases, and URLs occurred in the data, making tokenization challenging. Hence, the data was cleaned so that text embedding becomes easy. Traditional vectorization approaches such as Bag of Words, Word2Vec, Global Vectors for Word Representation (GloVe) and FastText were used by the researchers as the base vectorization methods. Some papers show the use of stemming the tweets and then creating unigram, bigram and trigram which are then processed through TF-IDF. Multilingual BERT, XLM, BETO and ELMo are some of the most advanced text embedding techniques that were explained in the above research papers. (Aluru et al., 2020) have used the LASER text embedding method. LASER offers multilingual sentence representations for performing various NLP tasks. It supports over 90 languages written in 28 different alphabets. Latent Semantic Analysis (LSA) has also been used as a dimensionality reduction algorithm and has been shown to be very effective in dimensional classification problems. Neural networks with static BERT embeddings remarkably improve the model's overall performance according to (Rajput et al., 2021)

Commonly tested models for text classification in the above research papers are Logistic Regression, Random Forest, Support Vector Machines, Gradient Boosting and Deep Neural Network. The performances of these classification models were evaluated using metrics like accuracy, precision, recall and F1 score.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

The methodology for this research work has been discussed in this chapter. The main objective of this research work is to detect hate speech in different languages. Before building the final text classification model, there are several steps that need to be considered. Each step is further explained in detail. An overview of the methodology that would be followed for this study is as follows:

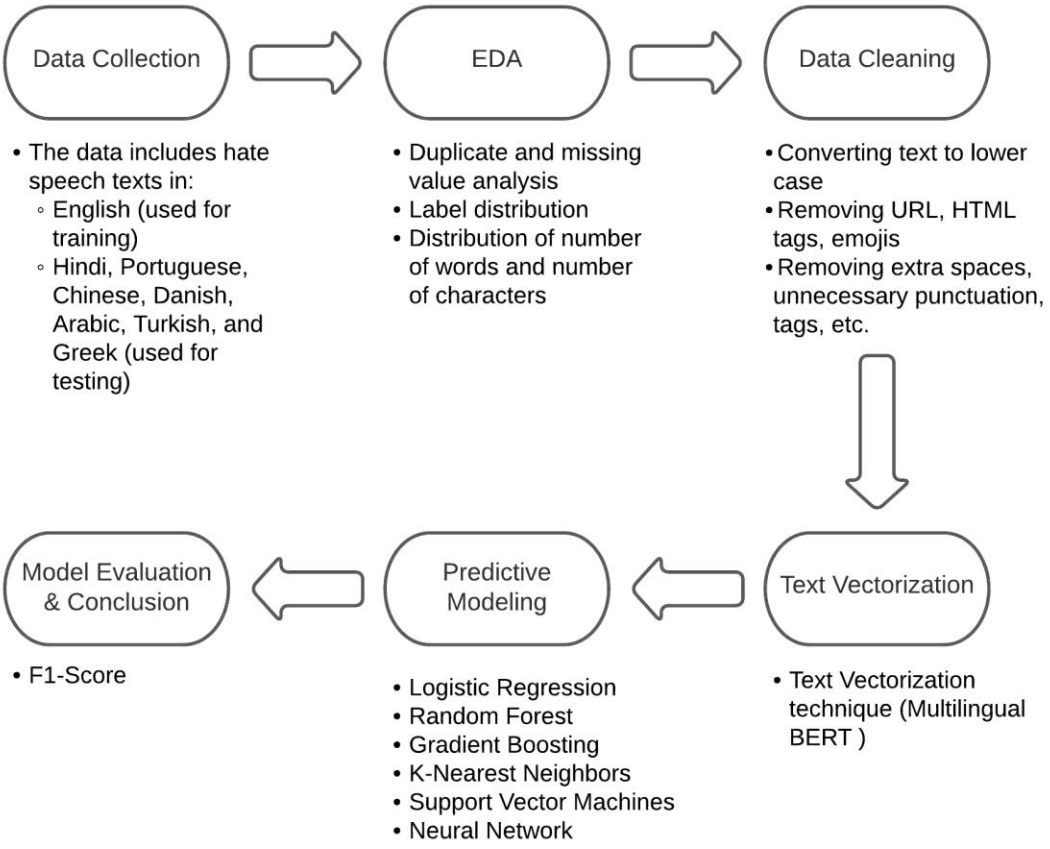

Figure 3.1 Research Methodology Overview

3.2 Data Collection and Description

The datasets for this research work come from a number of different sources because it is a cross-lingual hate speech detection study. The data is collected from reliable sources like Kaggle, Hugging Face and GitHub. The datasets include hate/non-hate speech texts in English, Hindi, Portuguese, Chinese, Danish, Arabic, Turkish, and Greek. There are two main columns in the data, one is the text column and another one is the label column. The hate texts are labeled as 1 while the non-hate texts are labeled as 0. Since English is the most widely used language on social media, there are cleaner datasets in English compared to rest of the mentioned languages. Hence, English language dataset was used for training the models. Even though the data is in English, the vectors are created using multi-lingual model. This model creates similar vectors for similar sentences in different languages. The best performing model was then tested on the rest of the multilingual datasets.

3.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method that uses a progressive approach to uncover the most essential and frequently hidden patterns in a data set. The following steps would be a part of this section:

- Duplicate and missing value analysis
- Label distribution diagram
- Distribution of the number of words and number of characters

3.4 Data Cleaning

Before performing the exploratory data analysis and processing the data through the model, the data needs to be cleaned. This can be done by:

- Converting text to lower case
- Removing URLs
- Removing HTML tags
- Removing emojis
- Removing extra spaces
- Removing unnecessary punctuations, tags, etc.

3.5 Text Vectorization

The text vectorization technique that was used is called BERT which is short for Bidirectional Encoder Representations from Transformers. BERT is based on the Transformer architecture. BERT has been pre-trained on a large corpus of unlabeled text, which includes the entire Wikipedia (2,500 million words!) and the Book Corpus (800 million words). BERT is a model that is "deeply bidirectional." Bidirectional indicates that during the training phase, BERT learns information from both the left and right sides of a token's context.

As the first step after EDA, the dataset was split into Train (80%) – Test (20%). In order to vectorize the text, a multilingual BERT based model (paraphrase-multilingual mpnet-base-v2) was used for converting the texts into numeric representations. These numeric representations of the texts were then used for building the models. This vectorization technique uses vectors to encode whole sentences and their semantic content. This aids in the model's understanding of the text's context, intent, and other nuances.

3.6 Predictive Modeling

As part of predictive modeling, various binary classification families of models were tested. Classification is a technique for categorizing data into a set number of classes. The main goal of a classification problem is to determine which category/class a new data set belongs to. In this study, the classes are hate speech vs non-hate speech. Different classification algorithms starting from linear models like logistic regression and progressing to neural network models were tested. The modeling techniques that were used cover a wide spectrum of classification algorithms. All these classification models were tuned using Grid Search CV to find the best hyperparameters for each model. Performances of all these models were checked and the best performing model was identified. The following classification techniques were explored and tested on the data:

- **Logistic Regression**: Logistic regression is a supervised classification technique. When the data is linearly separable and needs to be interpreted, it works well. The main issue arises when the data contains a significant degree of overlap between the classes. Because the weights are multiplicative rather than additive, the interpretation is more challenging.

- **Random Forest**: Random forests are a classification method that uses an ensemble of decision tree learning methods. Individual models make predictions that are unrelated to one another. The final output class is the majority class of all the individual trees' output classes. It solves the problem of overfitting. Since the bootstrap sample is randomly selected, any combination of rows and variables might cause bias in the tree and results.

- **K-Nearest Neighbors**: Neighbor-based classification is a type of lazy learning as it does not attempt to build a general internal model and instead simply stores instances of the training data. Classification is determined by a simple majority vote of each point's k nearest neighbors. This algorithm is simple to implement, resistant to noisy training data, and effective with large amounts of training data.

- **Gradient Boosting**: In Gradient Boosting the new trees are being trained to decrease the errors of preceding models. It constructs the model stage by stage. When a decision tree is used as the weak learner, the resulting algorithm is known as gradient boosted trees, and it typically surpasses random forest.

- **Support Vector Machines**: A separating hyperplane technically defines SVM as a classifier. SVM creates a separator that is utilized to divide distinct classes in multidimensional space by minimizing error. The goal of the model is to determine the optimum separator for dividing the dataset into n classes. It has a high accuracy compared to other classification models like logistic regression and decision trees, but it uses a lot of memory and is difficult to modify.

- **Neural Network**: Neural networks are a set of algorithms that recognize patterns and are roughly fashioned after the human brain. They aid in the grouping of data that is not labeled based on similarities between sample inputs and when they have a dataset that is labeled to train on, they can easily categorize the data.

3.7 Model Evaluation

Following are the main metrics that are associated with classification problems to determine a model's performance:

- Accuracy: Accuracy indicates the total number of correct predictions
  Accuracy = Number of correct predictions/ Number of all predictions

- Confusion Matrix: Although a confusion matrix is not an appropriate metric for evaluating a model, it does provide information about the predictions. The confusion matrix shows beyond accuracy by displaying the right and wrong classifications. False-positive is also called type I error. False-negative is also called type II error.

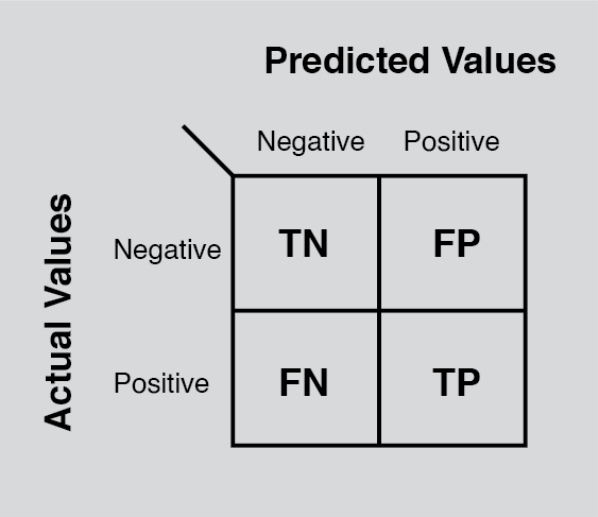

Figure 3.2 Confusion Matrix

- Precision and Recall: Precision determines the goodness of the model when the prediction is positive. Positive forecasts are the emphasis of precision. It shows how many of the positive predictions came true. Precision = TP/ TP + FP

  Recall determines how good the model is at accurately predicting positive classes. Actual positive classes are the basis of recall. It shows how many of the positive classes the model can accurately predict. Recall = TP/ TP + FN

Since precision and recall have a trade-off, we can't aim to maximize both. Increasing precision decreases recall and vice versa. Depending on the objective, we can focus on maximizing precision or recall.

- F1 Score: F1 Score is the weighted average of precision and recall. Since it accounts for both false positives and false negatives, f1-score is a highly relevant indicator than accuracy for situations with the uneven class distribution. F1 score = 2 *(Precision * Recall)/ (Precision + Recall). The best value for an F1 score is 1 and the worst is 0.

- Sensitivity and specificity:
  Sensitivity, also known as the true positive rate (TPR), is the same as recall. As a result, it calculates the percentage of positive classes that are accurately predicted to be positive.
  Specificity is comparable to sensitivity, except it is only concerned with the negative class. It calculates the percentage of negative classes that are correctly predicted as negative.

The following flow chart explains the research methodology that was followed for this research work:

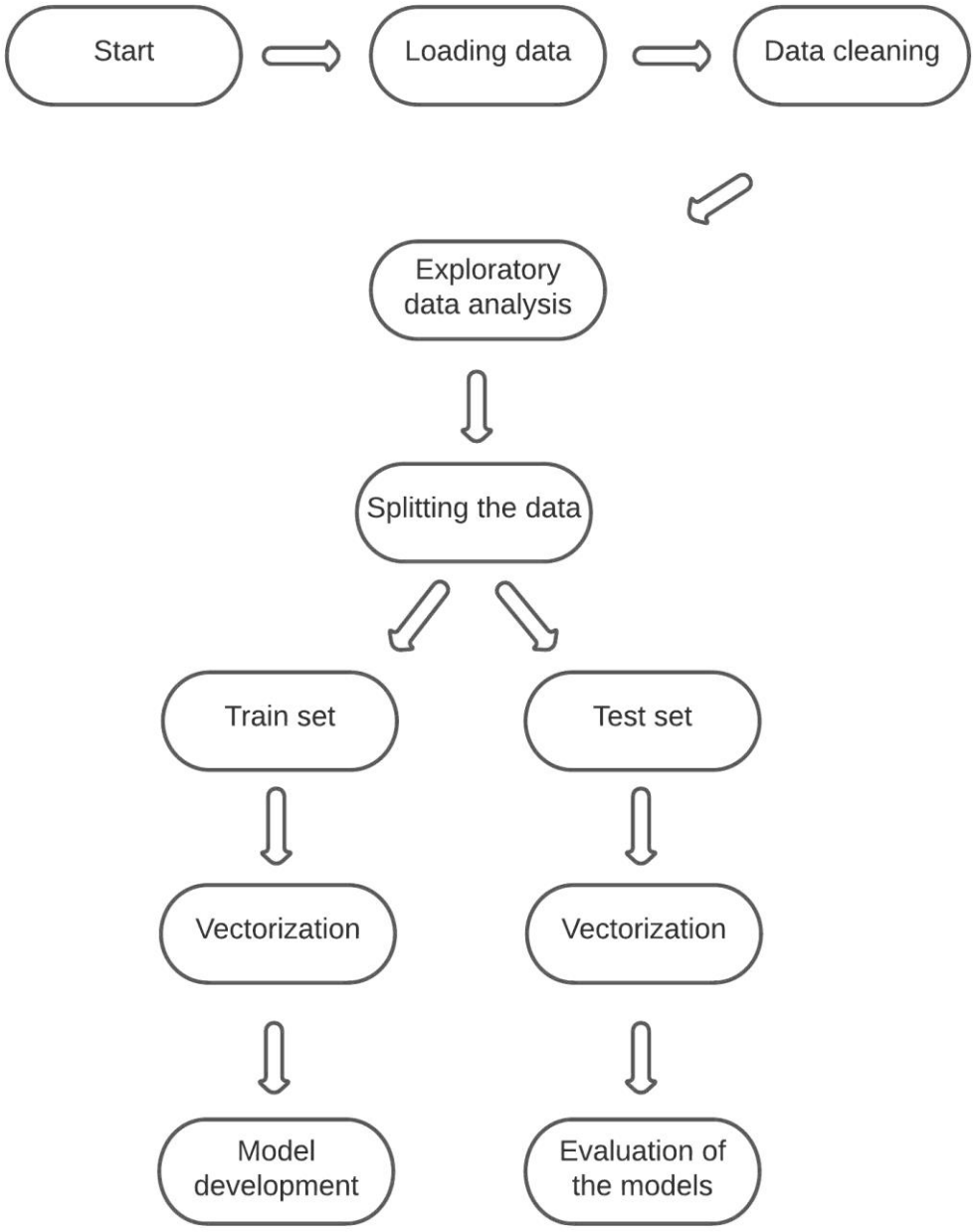

Figure 3.3 Research Methodology Flow Chart

3.8 Required Resources

This research work required the following hardware and software resources throughout the study execution.

Software Requirements:

Python will be used to build machine learning models

- Language: Python 3.8+
- Python Libraries for machine learning: Pandas and NumPy for data processing, Matplotlib and Seaborn for data visualization, scikit-learn, sentence_tranformers library for data pre-processing, predictive modeling & model evaluation

Hardware Requirements:

A laptop with the following specifications will be used:

- Memory: 12 GB
- Operating System. Windows 10: 64-bit
- Processor: Intel Core i7 10th Gen Processor

3.9 Summary

After doing an extensive background study, problem analysis and literature review, the proposed methodology described in this chapter was used for predicting hate speech. The research methodology includes data from various established and reliable sources, data cleaning steps, EDA, text vectorization technique and predictive modeling algorithms along with model evaluation metrics to identify the best performing model for detecting hate speech. In the next chapter, the analysis of this research methodology will be explained in detail.

CHAPTER 4: ANALYSIS

4.1 Introduction

This chapter will describe EDA, balancing of hate proportion, data cleaning, vectorization and hyperparameter tuning done for all the models.

4.2 Exploratory Data Analysis

Since there are only two columns and one of which is text, exploratory data analysis becomes limited. The following charts explain the data (English data which was used for training the models):

- **Label distribution**

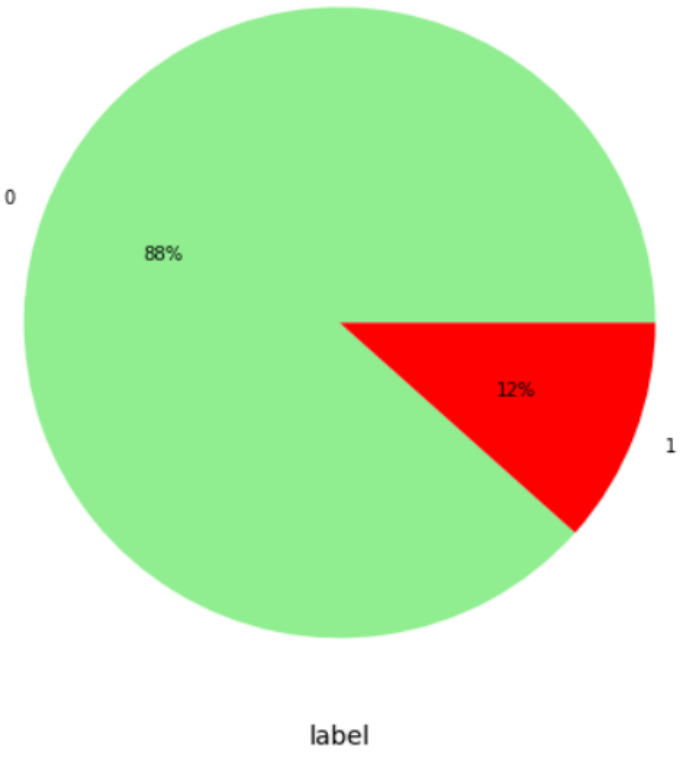

Figure 4.2 Label Distribution Chart

We see that in the above data 12% of the data is hate and 88% is non-hate

- **Distribution of number of words**

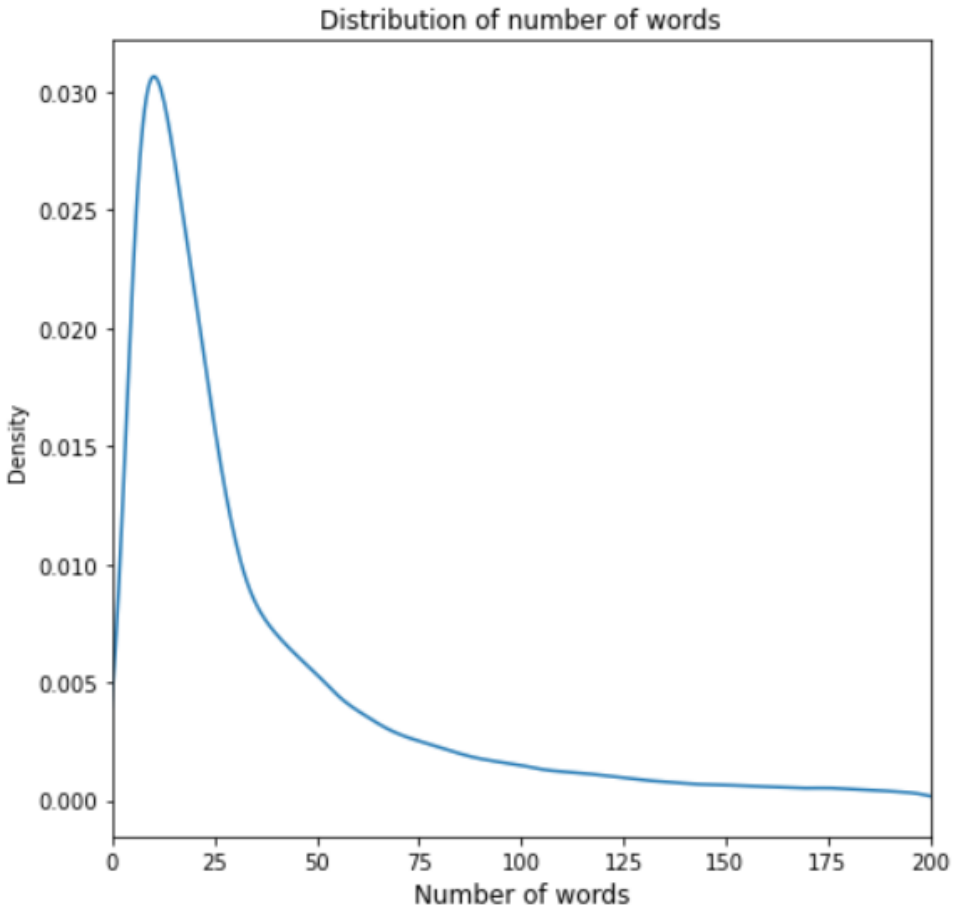

Figure 4.3 Distribution of number of words

We see that the majority of the texts have less than 25 words.

- **Distribution of the number of characters**

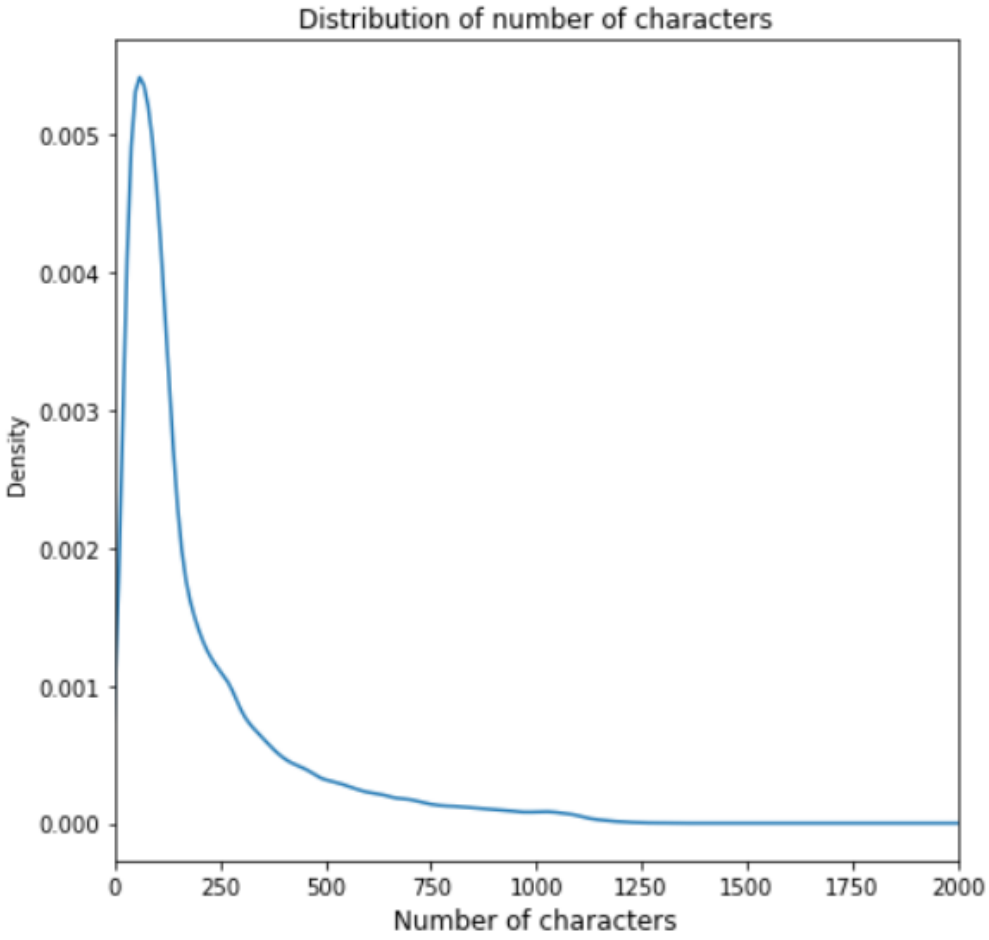

Figure 4.4 Distribution of number of characters

We see that the majority of the texts have less than 250 characters.

- **Hate proportion table**

label	male	female	straight	lgbtq	hindu	christian	muslim	jew	latino	white	black	asian
non-hate	13.3%	6.1%	0.3%	0.9%	0.2%	0.5%	0.9%	0.8%	0.1%	0.6%	0.3%	0.1%
hate	11.1%	14.6%	0.6%	7.4%	0.2%	0.8%	4.5%	4.2%	0.2%	2.5%	2.7%	0.7%
hate_proportion	0.83	2.40	1.95	8.35	1.38	1.52	4.92	5.10	2.92	3.96	7.73	8.92

Table 4.1 Hate Proportion

Table 4.1 compares the mentions of specific groups in hate vs non-hate sentences. The groups are divided according to gender, sexual orientation, religion and race.

E.g.: We see that of all the sentences containing hate, 14.6% of the sentences have mentions of females while of all the sentences containing hate, 6.1% of the sentences have mentions of females. The hate proportion here is calculated as 14.6 / 6.1 = 2.40

We see that:

- The ratio of hate proportion for female to male is 2.40:0.83, which clearly shows that there is more hate towards females compared to males in the data

- The ratio of hate proportion for LGBTQ to straight is 8.35:1.95, which clearly shows that there is more hate towards LGBTQ compared to straights in the data

- The ratio of hate proportion for Jews (5.10) and Muslim (4.92) is significantly more compared to Christians (1.52) and Hindus (1.38) in the data. This clearly shows that there is more hate towards Muslims and Jews compared to Christians and Hindus in the data

- The ratio of hate proportion for blacks (7.73) and Asians (8.92) is significantly more compared to whites (3.96) and Latinos (2.92) in the data. This clearly shows that there is more hate towards blacks, and Asians compared to Whites and Latinos in the data

4.3 Balancing the hate proportion

From the above table 4.1, we see that the hate proportion is not equally distributed amongst different target populations. In order to build models that are robust, we need data that is more balanced. Hence, built counterfactuals for balancing the data. Counterfactuals are statements that swap out one group for another while keeping the intention of the sentence the same. E.g.: “Men are brave” becomes “Women are brave” or vice-versa. Another example could be “White people are hardworking” becomes “Black people are hardworking” or vice-versa.

Now after augmenting the data, we see that the data is comparatively more balanced (refer table 4.2)

label	male	female	straight	lgbtq	hindu	christian	muslim	jew	latino	white	black	asian
non-hate	16.3%	14.9%	1.6%	1.0%	1.4%	1.9%	2.0%	1.9%	1.5%	1.6%	1.4%	1.4%
hate	15.9%	15.2%	7.3%	5.2%	3.9%	4.7%	5.0%	6.3%	4.7%	6.8%	6.1%	5.9%
hate_proportion	0.98	1.02	4.58	5.30	2.73	2.43	2.51	3.26	3.07	4.20	4.25	4.26

Table 4.2 Hate Proportion on data with counterfactuals

4.4 Data Cleaning

All the files used were CSV files which were read into python. To use the data for modeling, it needed to be cleaned. Cleaning the text data included converting text to lower case, removal of URLs, HTML tags, emojis, extra spaces and punctuations, tags, etc. A function was created called ‘text_cleanup’ which takes in text as an input and returns the cleaned version of the text. Here is an example of a text which is cleaned by the text_cleanup function:

```
text = "@sam When twitter rappers DM me their     TRASH links!!!! http://t.co/3587nMR4AUQg 😂😒"
text_cleanup(text)
```

```
'when twitter rappers dm me their trash links'
```

Figure 4.1 Text Clean-up Function

After cleaning the data, duplicates and missing values were checked in the data. The text column was unique and the datasets had no missing values.

4.5 Hyperparameter Tuning

In order to find the best parameters for the models, hyperparameter tuning was done for each of them. The parameters used for tuning and the best parameters for each of the models are as follows:

- **Logistic Regression**:

  The parameters that were used for tuning are as follows:

  1. 'class_weight': ['balanced', None]

     The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n_samples / (n_classes * np.bincount(y)).

  2. 'penalty': ['l1', 'l2', 'elasticnet']

     'l2': adds a L2 penalty term and it is the default choice

     'l1': adds a L1 penalty term

     'elasticnet': adds both L1 and L2 penalty terms

  3. 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]

     The inverse of regularization strength; must be a positive float. Smaller values specify stronger regularization.

  After doing a grid search the best parameters came out to be:

  {'class_weight': None,

  'penalty': 'l2'

  'C': 100}

- **Random Forest Classifier**

  The parameters that were used for tuning were:

  1. 'max_features': [0.05, 0.10, 0.25]
     The number of features to consider when looking for the best split

  2. 'min_samples_leaf': [5, 13, 21, 35]

  The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches.

  After doing a grid search the best parameters came out to be:
  {'max_features': 0.25,
  'min_samples_leaf': 5}

- **Gradient Boosting**

  The parameters that were used for tuning were:

  1. 'max_depth': [5, 25, 50, 100]
     The maximum depth of the tree.

  2. 'n_estimators': [250, 500, 1000]
     The number of trees in the forest.

  After doing a grid search the best parameters came out to be:
  {'max_depth': 5,
  'n_estimators': 1000}

- **K-Nearest Neighbors**

  The parameters that were used for tuning were:

  1. 'n_neighbors': [3, 7, 15, 25]
     Number of neighbors to use.

  2. 'weights': ['uniform','distance']
     'uniform': uniform weights. All points in each neighborhood are weighted equally.
     'distance': weight points by the inverse of their distance. In this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.

  After doing a grid search the best parameters came out to be:
  {'n_neighbors': 7,
  'weights': 'distance'}

- **Support Vector Machine**

  The parameters that were used for tuning were:

  1. 'C': [0.001, 0.01, 0.1, 1, 10]
     Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.

  2. 'kernel': ['linear', 'poly', 'rbf', 'sigmoid']
     Specifies the kernel type to be used in the algorithm. If none is given, 'rbf' will be used.

  3. 'class_weight': ['balanced', None]

The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n_samples / (n_classes * np.bincount(y))

After doing a grid search the best parameters came out to be:

{'C': 10,
'class_weight': None,
'kernel': 'rbf'}

- **Neural Network (MLP Classifier)**

  The parameters that were used for tuning were:

  1. 'hidden_layer_sizes' : [(512,128,32,8), (256,84,28,9), (192,48,12)
     The ith element represents the number of neurons in the ith hidden layer.

  2. 'alpha': [0.0001, 0.001, 0.01, 0.1]
     L2 penalty (regularization term) parameter.

  3. 'activation': ['tanh', 'relu']
     ‘tanh’: the hyperbolic tan function, returns f(x) = tanh(x).\
     ‘relu’: the rectified linear unit function, returns f(x) = max(0, x)

  After doing a grid search the best parameters came out to be:

  {'hidden_layer_sizes': (512, 128, 32, 8),
  'alpha': 0.001,
  'activation': 'relu'}

4.6 Summary

This chapter describes the process of cleaning text data. It also includes EDA which includes: label distribution, distribution of number of words, distribution of number of characters and hate proportion table. From label distribution chart we saw that 12% of the data is hate and 88% of the data is non-hate. Distribution of number of words chart and distribution of number of characters chart showed that majority of the texts contain less than 25 words and less than 250 characters. The hate proportion table showed that there was an imbalance in hate and non-hate proportions for different groups. Hence, the data was balanced by building counterfactuals. This chapter also explains the hyperparameters used for tuning the models. For each model, the best hyperparameters found were highlighted.

CHAPTER 5: RESULTS AND DISCUSSIONS

5.1 Introduction

This chapter will describe the results of the modeling exercise and conclude on the best-performing model. Built several models like Logistic Regression, Random Forest, Gradient Boosting, K-Nearest Neighbors, Support Vector Machines and Neural Network. The best performing model which was MLP Classifier was later tested on data in different languages.

5.2 Results/Evaluation

In order to compare the performances of the models, F1-score was used as it is a more stable metric to use when the data is imbalanced. Following table 5.1 shows the F1 scores of train and test sets for different models:

Model	Train Set (F1 Score)	Test Set (F1 Score)
Logistic Regression	0.78	0.77
Random Forests	0.98	0.75
Gradient Boosting	1.00	0.87
K-Nearest Neighbors	1.00	0.87
Support Vector Machines	1.00	0.95
Neural Networks (MLP Classifier)	0.98	0.92

Table 5.1 F1-Scores for train and test sets for each modeling technique

- Logistic Regression:

  F1-score score for the train set is 0.78 and for the test set is 0.77
- Random Forest Classifier:

  F1-score score for the train set is 0.98 and for the test set is 0.75
- Gradient Boosting:

  The F1-score score for the train set is 1.00 and for the test set is 0.87
- K-Nearest Neighbors:

  The F1-score score for the train set is 1.00 and for the test set is 0.87

- Support Vector Machines:
  The F1-score score for the train set is 1.00 and for the test set is 0.95
- Neural Network (MLP Classifier):
  The F1-score score for the train set is 0.98 and for the test set is 0.92

We see that SVM has the highest F1-score on the test set but it is not flexible because SVMs do not predict a probability but instead, it predicts a class directly and that rigidity does not help to reach a very high precision or recall by adjusting the threshold. Hence, Neural Network (MLP Classifier) is the best model with the F1-score of 0.92 on the test set.

Next, tested the Neural Network (MLP Classifier) on the multi-lingual datasets. Following table 5.2 shows the results:

Data Language	F1 Score
English	0.99
Portuguese	0.99
Arabic	0.96
Hindi	0.96
Greek	0.93
Chinese	0.92
Danish	0.91
Turkish	0.91

Table 5.2 F1-scores for multi-lingual test sets

The Neural Network (MLP Classifier) model performed the best on the English and the Portuguese dataset with an F1-score of 0.99. The model performs fairly well on Chinese, Arabic, Turkish, Hindi, Greek, and Danish data.

5.4 Summary

This chapter mentioned F1-score for each model that was built. For comparing the models F1-score was taken into consideration because F1-score was a better metric to use because our data was unbalanced. The final results showed that Neural Network (MLP Classifier) is the best model for hate speech classification. Hence, used this model for testing on multi-lingual data. The model performed best on the English and Portuguese datasets with F1-scores of 0.99 each. The model performs decently on other languages as well with the lowest F1 score being 0.91 for Turkish.

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary

- The main aim of this research work was to propose a model to predict if a text input, irrespective of language used, consists of hate speech or not. Amongst the several models that were tested, found that a Neural Network algorithm to be the best model to predict hate vs non-hate speech. Before reaching the final product, studied the state-of-the-art approaches of text classification models using BERT-based text vectorization techniques. Data cleaning, EDA and text vectorization were part of the initial process. Built multiple models and determined the best-performing model based on the evaluation metric (F1-Score).

6.2 Contributions

- Since English is the widely used language on social media, the models used by the social media platforms to detect hate speech are only in English. Hence, if a hate speech text is in a language other than English then it goes undetected. The model that is built through this research work will help in detecting multi-lingual hate speech and thereby protecting the targeted groups from hate speech.

- The hate speech detection work that was mentioned in the literature review, detect hate well for marginalized group like women, blacks, LGBTQ groups but it does not identify hate well against non-marginalized groups like men, white people, straight people. In this research work, the data is de-biased by creating counterfactuals that helped in creating balanced hate and non-hate speech data for all the groups.

6.3 Recommendations

- All social media platforms should build models that would detect hate in multiple languages so that no hate goes undetected on social platforms.
- Also, these models should be able to detect hate evenly, whether it's towards marginalized groups or non-marginalized groups.

6.4 Limitations

The model used for vectorization (paraphrase-multilingual-mpnet-base-v2) can vectorize texts in the languages that it is based on. If a language is outside the recognition of this language model, then it will not be able to create vectors for that particular language.

6.5 Future Work

The sentence- "They don't deserve to be called niggers" should not be detected as hate, but since the sentence contains a slur, the model picks it up as hate. If a slur occurs in a sentence, the model detects it as hate even though the overall sentence is not hateful. This is because there are not enough non-hateful sentences with slurs in the data, i.e., sentences that contain slurs are overwhelmingly labeled as hate. Hence, when the model sees a slur in a sentence, it tends to classify it as hate. For future work, if and when better data becomes available that contains non-hateful sentences containing slurs then more robust hate speech detection models could be built.

REFERENCES

Aluru, S.S., Mathew, B., Saha, P. and Mukherjee, A., (2020) Deep Learning Models for Multilingual Hate Speech Detection. [online] Available at: http://arxiv.org/abs/2004.06465.

Badjatiya, P., Gupta, S., Gupta, M. and Varma, V., (2017) Deep learning for hate speech detection in tweets. In: *26th International World Wide Web Conference 2017, WWW 2017 Companion*. International World Wide Web Conferences Steering Committee, pp.759–760.

Davidson, T., Warmsley, D., Macy, M. and Weber, I., (2017) Automated Hate Speech Detection and the Problem of Offensive Language. [online] Available at: http://arxiv.org/abs/1703.04009.

Gaydhani, A., Doma, V., Kendre, S. and Bhagwat, L., (2018) Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach. [online] Available at: http://arxiv.org/abs/1809.08651.

de Gibert, O., Perez, N., García-Pablos, A. and Cuadros, M., (2018) Hate Speech Dataset from a White Supremacy Forum. [online] Available at: http://arxiv.org/abs/1809.04444.

Kamble, S. and Joshi, A., (2018) Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models. [online] Available at: http://arxiv.org/abs/1811.05145.

Kapil', P., Ekbal', A. and Das, D., (n.d.) *Investigating Deep Learning Approaches for Hate Speech Detection in Social Media*.

Lee, Y., Yoon, S. and Jung, K., (2018) Comparative Studies of Detecting Abusive Language on Twitter. [online] Available at: http://arxiv.org/abs/1808.10245.

Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Goyal, P. and Mukherjee, A., (2020) HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. [online] Available at: http://arxiv.org/abs/2012.10289.

Pamungkas, E.W., Basile, V. and Patti, V., (2021) A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, 584, p.102544.

Plaza-del-Arco, F.M., Molina-González, M.D., Ureña-López, L.A. and Martín-Valdivia, M.T., (2021) Comparing pre-trained language models for Spanish hate speech detection. *Expert Systems with Applications*, 166, p.114120.

Ranasinghe, T. and Zampieri, M., (2020) Multilingual Offensive Language Identification with Cross-lingual Embeddings. [online] Available at: http://arxiv.org/abs/2010.05324.

Vijayaraghavan, P., Larochelle, H. and Roy, D., (2021) Interpretable Multi-Modal Hate Speech Detection. [online] Available at: http://arxiv.org/abs/2103.01616.

Waseem, Z. and Hovy, D., (n.d.) *Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter*. [online] Available at: http://github.com/zeerakw/hatespeech.

Zhang, Z. and Luo, L., (2018) Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. [online] Available at: http://arxiv.org/abs/1803.03662.

Agarwal, S. and Chowdary, C.R., (2021) Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. *Expert Systems with Applications*, 185.

Ahmed, H.A., Shiva Prasad, P.S., Ahmed, S.R., Uday, A. and Akhil, S., (n.d.) Deep Learning Based Fusion Approach for Hate Speech Detection. *JOURNAL OF RESOURCE MANAGEMENT AND TECHNOLOGY*, [online] 12, p.2021. Available at: www.jrmat.com.

Ali, M.Z., Ehsan-Ul-Haq, Rauf, S., Javed, K. and Hussain, S., (2021) Improving Hate Speech Detection of Urdu Tweets Using Sentiment Analysis. *IEEE Access*, 9, pp.84296–84305.

Budi Herwanto, G., Maulida Ningtyas, A., Gede Mujiyatna, I., Nyoman, I., Trisna, P. and Eka Nugraha, K., (2021) Hate Speech Detection in Indonesian Twitter using Contextual Embedding Approach. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems*, 152, pp.177–188.

Cinelli, M., Pelicon, A., Mozetič, I., Quattrociocchi, W., Novak, P.K. and Zollo, F., (2021) Online Hate: Behavioural Dynamics and Relationship with Misinformation. [online] Available at: http://arxiv.org/abs/2105.14005.

Das, M., Saha, P., Dutt, R., Goyal, P., Mukherjee, A. and Mathew, B., (2021) You too Brutus! Trapping Hateful Users in Social Media: Challenges, Solutions & Insights. [online] Available at: http://arxiv.org/abs/2108.00524.

Dhanya, L.K. and Balakrishnan, K., (2021) Hate speech Detection in Asian Languages:A Survey. Institute of Electrical and Electronics Engineers (IEEE), pp.1–5.

Jiang, A. and Zubiaga, A., (2021) Cross-lingual Capsule Network for Hate Speech Detection in Social Media. [online] Available at: http://arxiv.org/abs/2108.03089.

Mansourifar, H., Alsagheer, D., Fathi, R., Shi, W., Ni, L. and Huang, Y., (2021) Hate Speech Detection in Clubhouse. [online] Available at: http://arxiv.org/abs/2106.13238.

Markov, I. and Daelemans, W., (2021) *Improving Cross-Domain Hate Speech Detection by Reducing the False Positive Rate*. [online] Available at: https://huggingface.co/.

Markov, I., Ljubeši´c, N.L., Fišer, D. and Daelemans, W., (2021) *Exploring Stylometric and Emotion-Based Features for Multilingual Cross-Domain Hate Speech Detection*. [online] Available at: https://simpletransformers.ai/.

Miok, K., Škrlj, B., Zaharie, D. and Robnik-Šikonja, M., (2021) To BAN or Not to BAN: Bayesian Attention Networks for Reliable Hate Speech Detection. *Cognitive Computation*.

Mullah, N.S. and Zainon, W.M.N.W., (2021) Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review. *IEEE Access*.

Parihar, A.S., Thapa, S. and Mishra, S., (2021) Hate Speech Detection Using Natural Language Processing: Applications and Challenges. Institute of Electrical and Electronics Engineers (IEEE), pp.1302–1308.

Perifanos, K. and Goutsos, D., (2021) Multimodal hate speech detection in greek social media. *Multimodal Technologies and Interaction*, 57.

Plaza-Del-Arco, F.M., Molina-Gonzalez, M.D., Urena-Lopez, L.A. and Martin-Valdivia, M.T., (2021) A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9, pp.112478–112489.

Plaza-del-Arco, F.M., Molina-González, M.D., Ureña-López, L.A. and Martín-Valdivia, M.T., (2021) Comparing pre-trained language models for Spanish hate speech detection. *Expert Systems with Applications*, 166.

Pronoza, E., Panicheva, P., Koltsova, O. and Rosso, P., (2021) Detecting ethnicity-targeted hate speech in Russian social media texts. *Information Processing and Management*, 586.

Qureshi, K.A. and Sabih, M., (2021) Un-Compromised Credibility: Social Media Based Multi-Class Hate Speech Classification for Text. *IEEE Access*, 9, pp.109465–109477.

Rajput, G., punn, N.S., Sonbhadra, S.K. and Agarwal, S., (2021) Hate speech detection using static BERT embeddings. [online] Available at: http://arxiv.org/abs/2106.15537.

Sultan, D., Mussiraliyeva, S., Toktarova, A., Nurtas, M., Iztayev, Z., Zhaidakbaeva, L., Shaimerdenova, L., Akhmetova, O. and Omarov, B., (2021) Cyberbullying and Hate Speech Detection on Kazakh-Language Social Networks. Institute of Electrical and Electronics Engineers (IEEE), pp.197–201.

Vitiugin, F., Senarath, Y. and Purohit, H., (2021) Efficient Detection of Multilingual Hate Speech by Using Interactive Attention Network with Minimal Human Feedback. In: *ACM International Conference Proceeding Series*. Association for Computing Machinery, pp.130–138.

Zampieri, N., Illina, I. and Fohr, D., (2021) Improving Automatic Hate Speech Detection with Multiword Expression Features. [online] Available at: http://arxiv.org/abs/2106.00237.

Zhou, X., Yang, Y., Fan, X., Ren, G., Song, Y., Diao, Y., Yang, L. and Lin, H., (n.d.) *Hate Speech Detection based on Sentiment Knowledge Sharing*. [online] Available at: https://github.com/1783696285/SKS.

APPENDIX A: RESEARCH PLAN

The above image shows the plan followed for this research work.

APPENDIX B: RESEARCH PROPOSAL

CROSS-LINGUAL HATE SPEECH DETECTION

Samrudhi Keluskar

Liverpool John Moores University – Master of Science in Data Science

Research Proposal

August 2021

Abstract

On social media, racist and sexist remarks are common forms of hate speech. Hate speech consists of a very wide range of expressions that promotes or justifies hatred, violence, and discrimination against an individual or group of individuals for a variety of reasons. This research proposal is based on detecting hate speech in different languages. BERT, a text embedding method for different languages is clearly explained in this proposal. Algorithms like Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, Neural Network which will be used to classify the text into hate vs non-hate are summarized in this report. Various classification metrics to evaluate the models are thoroughly described so that the best-performing model can be easily identified. The purpose of this study is to identify hate speech on social media and internet platforms so that measures can be taken to prevent the spread of hatred. The identification and segmentation of hate speech on social media platforms will contribute to the reduction of hatred and violence directed at specific groups.

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUC-ROC…… Area Under the Curve-Receiver Operating Characteristic Curve

BERT…… Bidirectional Encoder Representations from Transformers

BiLSTM…… Bidirectional Long Short-term Memories

CNN…….. Convolutional Neural Networks

CV……. Cross-Validation

FN……. False Negative

FP…….. False Positive

LSTM…… Long Short-Term Memories

SVM…….. Support Vector Machines

TF-IDF……..… Term Frequency-Inverse Document Frequency

TN……. True Negative

TN……. True Positive

TPR……... True Positive Rate

1. Background

Hate speech refers to any type of expression that is intended to humiliate, criticize, or instigate hatred towards a group or a class of people based on race, religion, skin color, sexual identity, gender identity, ethnicity, disability, or national origin. In recent years, there has been a boom in interest in detecting abusive language, hate speech, cyberbullying, and trolling (Schmidt and Wiegand, 2017). Social media platforms have been under increasing pressure to address these issues. Because of the similarities between these subtasks, scholars have grouped them under the umbrella titles "abusive language," "harmful speech," and "hate speech."

Hate speech is protected in the United States under the First Amendment's free speech provisions, but that is disputed in the legal body and in relation to speech codes on university campuses. Hate speech is a felony in several countries, including the United Kingdom, Canada, and France. Hate speech is defined as remarks directed at minorities in a manner that may end in violence or social disorder. Individuals who are guilty of employing hate speech must pay hefty penalties and may even face prison time. These regulations apply to online sites, prompting online sites to implement anti-hate speech policies of their own. In response to criticism that they are not adequately helping in preventing hate speech on their platforms, Facebook and Twitter have implemented policies prohibiting the usage of their websites for attacking people based on race, ethnicity, gender, or sexual orientation, as well as violent threats against others.

In extreme circumstances, this could include language that terrorizes or promotes violence, but if we only consider this definition then we would leave out a significant amount of hate speech. Importantly, the definition here excludes most of the occurrences of objectionable language because individuals frequently use expressions that are very offensive to specific individuals but are used in fundamentally many ways. For instance, some African Americans frequently use the term n*gga in a common online conversation (Warner and Hirschberg 2012). When quoting rap lyrics, rappers use phrases like h*e and b*tch, and young video game players commonly used homophobic slurs like f*g. On social media, this kind of language is common (Wang et al. 2014). As a result, boundary conditions are essential for any hate speech detection system.

The purpose of this research work is to build a classification model that can detect hate speech on different online platforms. These platforms can then take actions to prevent the spread of hatred on their platforms. Overall, this can help reduce violence and hate in society.

2. Related Research

It has become crucial for all social media and internet platforms to keep their sites clean. Several researchers are working towards detecting hate speech on the internet. Following few works are done in this field.

(Davidson et al., 2017) have mentioned the use of a crowd-sourced hate speech set of terms to gather tweets that contain hate speech keywords. A sample of these tweets was labeled into three classes: hate speech, offensive language, and neither using crowdsourcing. They first stemmed the tweets and then created unigram, bigram and trigram which were then processed through TF-IDF. They had trained multiple different models and found that Logistic Regression and Linear SVM tended to perform notably better than other models. The top-performing model had an overall precision of 0.91, recall of 0.90, and F1 score of 0.90. They discovered that racist and homophobic tweets were more likely to be labeled as hate speech, whereas sexist tweets were frequently classified as offensive. It was difficult to classify tweets without explicit hate keywords.

(Gaydhani et al., 2018) presented a method for categorizing tweets on Twitter into three categories: hateful, offensive, and clean. They experimented with n-grams as features and passed their term frequency-inverse document frequency (TF-IDF) values to several machine learning models using a Twitter dataset. They looked at three popular text classification machine learning algorithms: Logistic Regression, Naive Bayes, and Support Vector Machines. They attained 95.6 percent accuracy on test data after fine-tuning the logistic model that produced the best results. 4.8 percent of the offensive tweets were misclassified as hateful, according to the findings. More instances of offensive language that does not contain hateful terms can be obtained to remedy this problem.

(Aluru et al., 2020) used 16 different sources to undertake a large-scale analysis of multilingual hate speech in 9 languages. They discovered that basic models like LASER embedding with logistic regression perform best in low-resource settings, while BERT-based models perform better in high-resource settings. When it comes to zero-shot classification, languages like Italian and Portuguese perform well. Their proposed architecture could be useful for languages with limited resources. These models could also be used as a starting point for future multilingual hate speech detection projects.

The task to classify a tweet as racist, sexist, or neither is difficult due to the intricacy of natural language constructs. To deal with this complexity, (Badjatiya et al., 2017) conducted extensive experiments with several deep learning architectures to learn semantic word embeddings. They investigated using deep learning techniques to the task of detecting hate speech. They looked at char n-grams, word Term FrequencyInverse Document Frequency (TF-IDF) values, Bag of Words Vectors (BoWV) over Global Vectors for Word Representation (GloVe), and task-specific embeddings learned with FastText, CNNs, and LSTMs. The researchers found that deep learning methods outperform state-of-the-art char/word n-gram methods by 18 F1 points on a benchmark dataset of 16K annotated tweets.

(Zhang and Luo, 2018) developed Deep Neural Network structures as feature extractors, which are good at apprehending hate speech semantics. Their methods surpass the highest performing method by up to 5 percentage points in macro-average F1 and 8 percentage points in the more difficult situation of recognizing hostile material. For evaluation, they used the conventional Precision, Recall, and F1 metrics.

(Waseem and Hovy, 2016) offered a list of critical race theory-based criteria, which they utilized to label a public corpus of over 16k tweets. For each tweet and the user description, they gather unigrams, bigrams, trigrams, and four grams. They ran a grid search over all relevant feature set combinations to find that character n-grams outperformed word n-grams by at least 5 F1-points. To assess the influence of various features on prediction performance and quantify their expressiveness, they utilized a logistic regression classifier and 10-fold cross-validation. They studied the impact of several non-linguistic factors in combination with character n-grams in

detecting hate speech. They also provide a lexicon based on the most relevant words found in their data.

Annotating big sets of data is exceedingly difficult due to the context-dependent nature of online aggressiveness. Previously researched datasets in abusive language detection were too small to build deep learning models efficiently. Hate Speech on Twitter, a considerably larger and more reliable dataset, was recently released. However, the full potential of this dataset has yet to be discovered. (Lee et al., 2018) conducted the first comparison research of several models on Hate and Abusive Speech on Twitter in this paper and highlighted the possibilities of incorporating more characteristics and context data to improve the model. Experiments demonstrate that bidirectional GRU networks with Latent Topic Clustering modules, trained on word-level features, are the best accurate model, scoring F1 of 0.805.

(de Gibert et al., 2018) described hate speech data made up of thousands of words that were manually classified as containing or not containing hate speech. The sentences were taken from Stormfront, a white supremacist discussion board. To complete the manual labeling work, a custom annotation tool was created, which allows annotators to choose whether to examine the context of a statement before labeling it. Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Recurrent Neural Networks with Long Short-term Memories were among the algorithms tested (LSTM). Individual accuracy is shown in the results for hate and no-hate, as well as overall accuracy. In addition, the report includes a thorough qualitative and quantitative analysis of the produced dataset, as well as multiple baseline tests with various categorization models.

(Mathew et al., 2020) introduced HateXplain, a benchmark hate speech dataset addressing several elements of the topic in this research. Every post in their dataset is explained in three ways: the basic, frequently used three-class classification (hate, offensive, or neither), the target group (the set of people who are hate speech victims in the post), and the reasons, i.e., the segment of the post on which their labeling decision (as hate, offensive or normal) is based. They used current state-of-the-art models and discovered that models which work well in classification do not do well on explainability criteria such as model plausibility and faithfulness. CNN-GRU, BiRNN,

and BERT are some of the models they've utilized. Accuracy, macro F1-score, and AUC-ROC score were among the parameters they reported

(Ranasinghe and Zampieri, 2020) used English data to create predictions in languages with fewer resources by using cross-lingual embeddings and transferred learning. They made predictions based on similar data in Spanish, Bengali and Hindi reporting F1 macro values of 0.84 for Bengali, 0.85 for Hindi, and 0.75 for Spanish. Finally, they demonstrated that their method outperforms the best systems in recently shared tasks in these three languages, demonstrating the durability of cross-lingual transfer learning for this task.

(Vijayaraghavan et al., 2021)proposed a deep learning model in this paper that can (a) identify hate speech by efficiently capturing the text semantics as well as the socio-cultural context in which a certain hatred expression is generated, and (b) deliver explainable insights into our model's decisions. They employed baseline deep learning models as well as traditional models. Deep learning models clearly outperform classical models, according to their findings. They demonstrated that their model outperforms existing state-of-the-art hate speech classification algorithms by conducting a thorough review of alternative modeling strategies. Finally, they demonstrate the significance of social and cultural environment factors in identifying clusters linked with various types of hate.

(Plaza-del-Arco et al., 2021) tackled the problem of detecting hate speech in Spanish on social media in this research and provided a greater grasp of the possibilities of new machine learning approaches. Emojis, mentions, hashtags, elongated words, phrases, and URLs occur in the content, making tokenization challenging. They evaluated Deep Learning's performance to that of newly transfer learning-based pre-trained language models, as well as conventional machine learning models. Their key offering is the use of multilingual and monolingual pre-trained language models like BERT, XLM, and BETO to obtain promising outcomes in Spanish.

(Pamungkas et al., 2021) investigated hate speech identification in data-scarce languages by using zero-shot learning to transfer knowledge from a data-rich language, English. They tested a variety of neural architectures and suggested different models that use various cross-lingual language

embeddings to transfer knowledge between languages. They also used information from a cross-lingual lexicon of abusive expressions to assess the influence of additional knowledge in their experiment. Their joint-learning models outperformed others in most languages, according to the findings. An easy strategy that utilizes machine translation and a pre-trained English language model, on the other hand, provides a reliable result. Multilingual BERT, on the other hand, did not perform well in cross-lingual hate speech detection.

(Kamble and Joshi, 2018) investigated hate speech identification in English-Hindi code-mixed tweets in this paper. They employed three common deep-learning models to identify hate speech (CNN-1D, LSTM, and BiLSTM) and empirically verified their usefulness. Their models were not only good at obtaining the semantics of hate speech but also its context. They also proved the effectiveness of domain-specific word embeddings in bringing intrinsic value to the code-mixed environment. Their research uses a benchmark dataset to demonstrate how deep learning models can improve on well-known statistical classifiers work.

(Kapil' et al., 2020) suggested deep learning algorithms for detecting several sorts of hate utterances online in this research, which used diverse embeddings. They used CNN, LSTM, BiLSTM, and CharacterCNN to create 13 deep learning models. Identifying hate speech from a huge proportion of text, particularly tweets with insufficient contextual information, presents a number of practical difficulties. Their studies on three publicly accessible datasets from various domains reveal that accuracy and F1-score have improved significantly. For all three datasets, LSTM and BiLSTM based on recurrent neural networks scored best.

Summary:

Most of the data used in the above research papers come from Twitter. (de Gibert et al., 2018) used data from Stormfront, a white supremacist discussion board. (Aluru et al., 2020) used 16 different sources to undertake a huge-scale investigation of multilingual hate speech in nine different languages. (Mansourifar et al., 2021) have used data from a voice-based social media platform called Clubhouse. (Das et al., 2021) made use of data from Gab, an American social networking service that is well known for its far-right user base.

Emojis, mentions, hashtags, elongated words, phrases, and URLs occurred in the data, making tokenization challenging. Hence, the data was cleaned so that text embedding becomes easy. Traditional vectorization approaches such as Bag of Words, Word2Vec, Global Vectors for Word Representation (GloVe) and FastText were used by the researchers as the base vectorization methods. Some papers show the use of stemming the tweets and then creating unigram, bigram and trigram which are then processed through TF-IDF. Multilingual BERT, XLM, BETO and ELMo are some of the most advanced text embedding techniques that were explained in the above research papers. (Aluru et al., 2020) have used LASER text embedding method. LASER offers multilingual sentence representations for performing various NLP tasks. It supports over 90 languages written in 28 different alphabets. Latent Semantic Analysis (LSA) has also been used as a dimensionality reduction algorithm and has been shown to be very effective in dimensional classification problems. Neural networks with static BERT embeddings remarkably improve the model's overall performance according to (Rajput et al., 2021)

Commonly tested models for text classification in the above research papers are Logistic Regression, Random Forest, Support Vector Machines, Gradient Boosting and Deep Neural Network. The performances of these classification models were evaluated using metrics like accuracy, precision, recall and F1 score.

3. Aim and Objectives

The key aim of this research work is to propose a model to predict if a text input consists of hate speech or not. The goal of this research study is to recognize hate speech on social media sites such as Twitter so that action can be taken to prevent the spread of hate.

The following research objectives are framed based on the aim of this study:

- To investigate the state-of-the-art approaches to the text classification models using BERT based text embedding techniques
- To preprocess the dataset for cleaning, removing duplicates, and sentence embedding

- To propose multiple text classification models for detecting hate speech
- To evaluate and compare the performances of the implemented algorithms in order to determine the best text classification model

4. Significance of the Study

The research is contributing to detecting hate speech in different languages using predictive classification models. This research can be used to identify hate speech which then can help in implementing strict measures to prevent the spread of hatred. Knowing which users are the root cause of spreading hatred aids in blocking such accounts. It is critical to put a stop to online hate because it has the potential to incite violence. Social media and internet platforms are the beneficiaries of this research. This study will help these platforms detect hate speech and take appropriate measures to stop hate on their platform. The detection and removal of hate text from social media platforms will aid in reducing hate and violence against certain groups.

5. Scope of the Study

Due to the time restrictions, the scope of the research study will be restricted as below:

- The hate speech detection models for this study will be built using the data in English, Hindi, Portuguese, Chinese, Spanish, Danish, Arabic, Turkish, and Greek
- Text vectorization will be done using BERT
- An only small set of machine learning algorithms will be used like Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, Neural Network
- Hyperparameters will be tuned only a few models like Random Forest and Neural Network

6. Structure of the Study

There would be six main chapters in this research thesis. The next five chapters' topic matter is described in this section.

Chapter 2: Consists of literature review of the latest papers in the hate speech domain.

Chapter 3: Consists of research methodology to be used for building prediction models to predict hate/non-hate speech. This chapter describes in detail steps like data collection and data cleaning, EDA, vectorization, predictive modeling techniques, evaluation metrics and required resources.

Chapter 4: Consists of analysis of the data after preprocessing, EDA, analysis of hate proportion, and hyperparameter tuning analysis.

Chapter 5: Consists of model results and discussion of the results. This chapter mentions the best performing model and the test data languages on which the model's performance is outstanding.

Chapter 6: Consists of the final conclusion of the research work. This chapter also describes the limitations and contribution of this research work. Provides suggestions for future work.

7. Research Methodology

The main objective of this study is to detect hate speech in different languages. Before building the final text classification model, there are several steps that need to be considered. Each step is further explained in detail. An overview of the methodology that would be followed for this study is as follows:

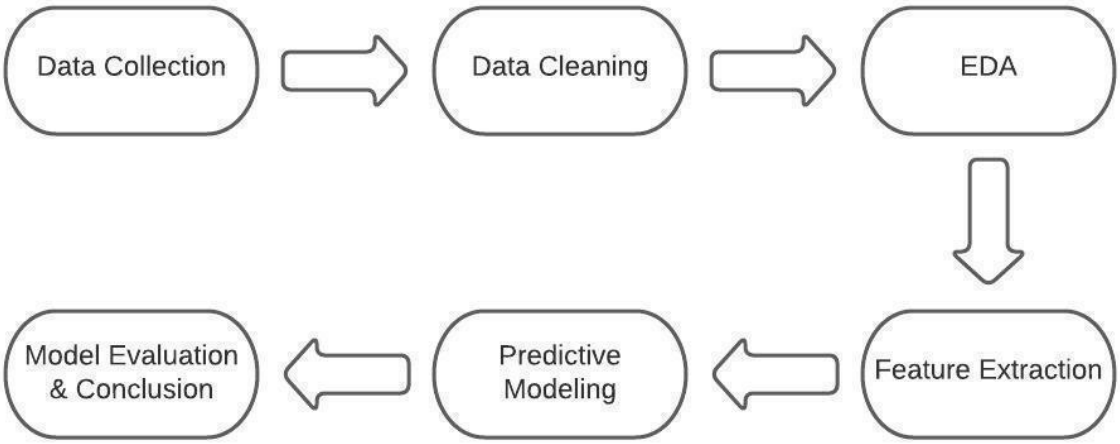

Figure 1: Research Methodology Overview

1. Data Collection and Description:

The datasets for this study come from several sources because it is a multilingual hate speech detection study. The datasets include hate/offensive speech texts in English, Hindi, Portuguese, Chinese, Spanish, Danish, Arabic, Turkish, and Greek. Since English is the most widely used language on social media, there are more datasets in that language. These datasets contain labeled hate/offensive speech texts.

2. Data Cleaning:

Before processing the data through the model, we will have to clean it. This can be done by:

- Converting the text to lower case
- Removing URLs
- Removing HTML tags
- Correcting misspelled words
- Removing unnecessary punctuation, tags, etc.

3. EDA:

Exploratory Data Analysis (EDA) is a method that uses a progressive approach to uncover the most essential and frequently hidden patterns in a data set. The following steps would be a part of this section:

- Duplicate and missing value analysis
- Label distribution

4. Vectorization:

As the first step after EDA, the dataset will be split into Train(65%) – Test(20%) – Validation(15%). Sentence embedding will be used to convert the texts to vectors. Sentence embedding techniques use vectors to encode whole sentences and their semantic content. This aids in the model's understanding of the text's context, intent, and other nuances.

The sentence embedding technique that will be used is called BERT which is short for Bidirectional Encoder Representations from Transformers. BERT is based on the Transformer architecture. BERT has been pre-trained on a large corpus of unlabeled text, which includes the entire Wikipedia (2,500 million words!) and the Book Corpus (800 million words). BERT is a model that is "deeply bidirectional." Bidirectional indicates that during the training phase, BERT learns information from both the left and right sides of a token's context.

5. Predictive Modeling:

As part of predictive modeling, the binary classification family of models will be chosen. Starting with linear models like logistic regression and progressing to neural network models, the following techniques will be explored and tested on the data:

- Logistic Regression: Logistic regression is a supervised classification technique. When the data is linearly separable and needs to be interpreted, it works well. The main issue arises when the data contains a significant degree of overlap between the classes. Because the weights are multiplicative rather than additive, the interpretation is more challenging.

- Decision Trees: If the link between variables and outcomes is nonlinear or if the variables are dependent on one another, logistic regression models fail. When we want the output to

be instinctive and the outcome to be explainable to non-technical people, the decision tree model is the best option. If no linear relationship exists, the decision tree fails, and if not tweaked, it tends to overfit.

- Random Forest: Random forests are a classification method that uses an ensemble of decision tree learning methods. Individual models make predictions that are unrelated to one another. The final output class is the majority class of all the individual trees' output classes. It solves the problem of overfitting. Since the bootstrap sample is randomly selected, any combination of rows and variables might cause bias in the tree and results.

- Gradient Boosting: In Gradient Boosting the new trees are being trained to decrease the errors of preceding models. It constructs the model stage by stage. When a decision tree is used as the weak learner, the resulting algorithm is known as gradient boosted trees, and it typically surpasses random forest.

- SVM: A separating hyperplane technically defines SVM as a classifier. SVM creates a separator that is utilized to divide distinct classes in multidimensional space by minimizing error. The goal of the model is to determine the optimum separator for dividing the dataset into n classes. It has a high accuracy compared to other classification models like logistic regression and decision trees, but it uses a lot of memory and is difficult to modify.

- Neural Network: Neural networks are a set of algorithms that recognize patterns and are roughly fashioned after the human brain. They aid in the grouping of data that is not labeled based on similarities between sample inputs and when they have a dataset that is labeled to train on, they can easily categorize the data.

Finally, Grid Search and Randomized Search CV will be used to modify some hyperparameters to see whether the top-performing model can be improved.

6. Model Evaluation

For classification problems, several measures are available and can be appropriately chosen based on the requirements. Following are few metrics that would be used to determine a model's performance:

- Accuracy: Accuracy indicates the total number of correct predictions
  Accuracy = Number of correct predictions/ Number of all predictions

- Confusion Matrix: Although a confusion matrix is not an appropriate metric for evaluating a model, it does provide information about the predictions. The confusion matrix shows beyond accuracy by displaying the right and wrong classifications.

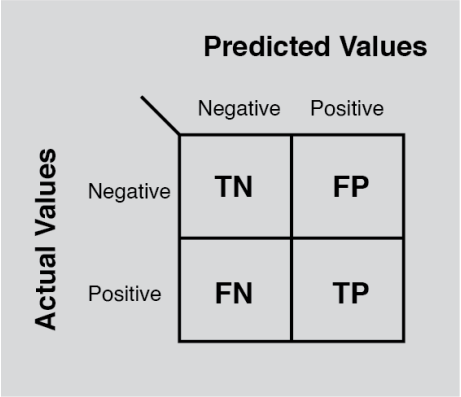

Figure 2: Confusion Matrix

False-positive is also called type I error. False-negative is also called type II error.

- Precision and Recall: Precision determines the goodness of the model when the prediction is positive. Positive forecasts are the emphasis of precision. It shows how many of the positive predictions came true. Precision = TP/ TP + FP
  Recall determines how good the model is at accurately predicting positive classes. Actual positive classes are the basis of recall. It shows how many of the positive classes the model can accurately predict.
  Recall = TP/ TP + FN

Since precision and recall have a trade-off, we can't aim to maximize both. Increasing precision decreases recall and vice versa. Depending on the objective, we can focus on maximizing precision or recall.

- F1 Score: F1 Score is the weighted average of precision and recall. Since it accounts for both false positives and false negatives, f1-score is a highly relevant indicator than accuracy for situations with uneven class distribution.
  F1 score = 2 *(Precision * Recall)/ (Precision + Recall)
  The best value for F1 score is 1 and the worst is 0.

The following flow chart explains the research methodology that will be followed for this study:

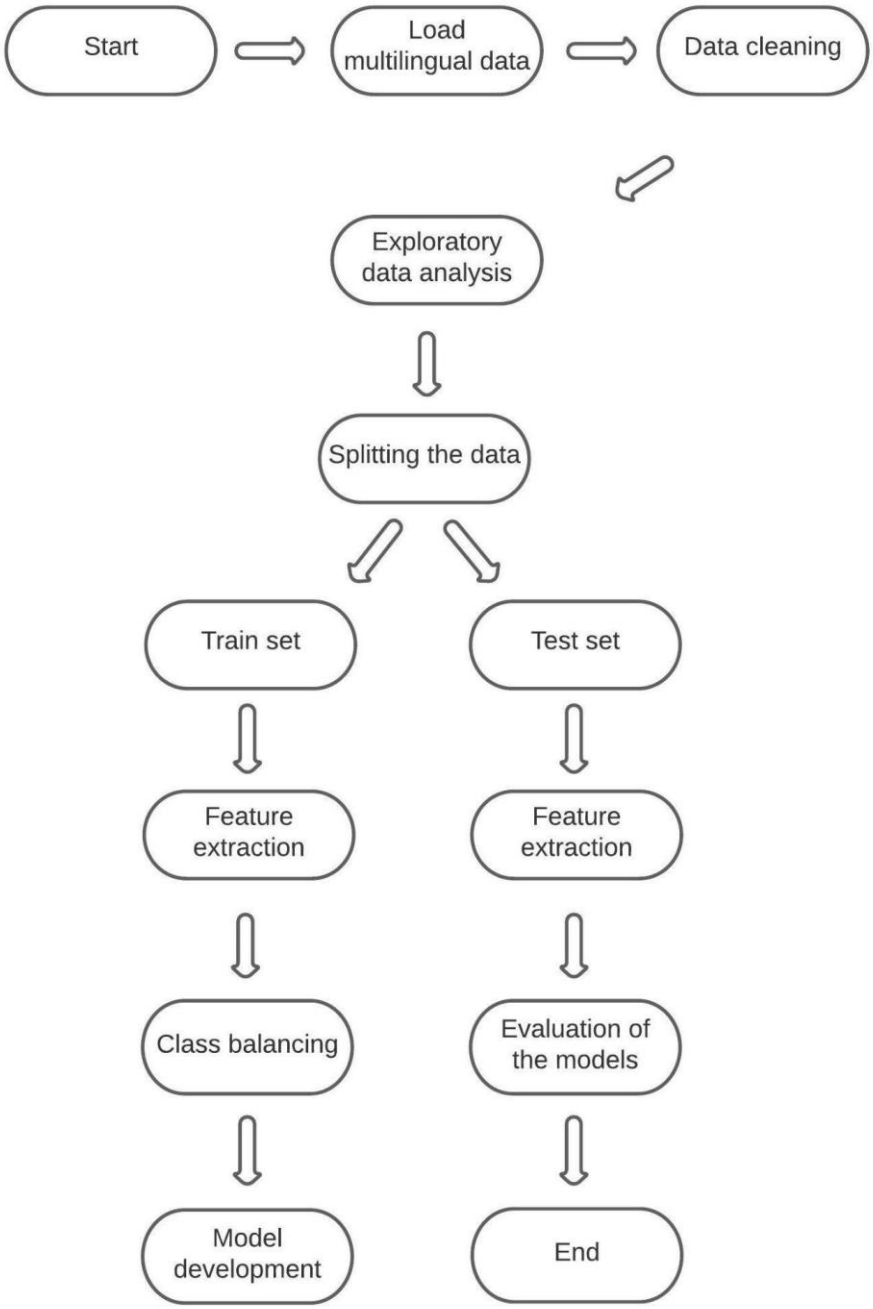

Figure 3: Research Methodology Flow Chart

7. Required Resources

The research will require the following hardware and software resources throughout the study execution.

Software Requirements:

Python will be used to build machine learning models

- Language: Python 3.8+
- Python Libraries for machine learning: Pandas and NumPy for data processing, Matplotlib and Seaborn for data visualization, TensorFlow, scikit-learn and Statsmodel library for data pre-processing, predictive modeling & model evaluation

Hardware Requirements:

A laptop with the following specifications will be used:

- Memory: 12 GB
- Operating System. Windows 10: 64-bit
- Processor: Intel Core i7 10th Gen Processor

8. Summary

After doing an extensive background study, problem analysis and literature review, the proposed methodology described in this chapter will be used for predicting hate speech. The research methodology includes data from various established and reliable sources, data cleaning steps, EDA and vectorization techniques, predictive modeling algorithms along with model evaluation metrics to identify the best performing model for detecting hate speech. Datasets in different languages will be considered for this study as a multilingual hate speech detection model will be built at the end of the study. Various classification models will be tried and tested to find the best hate and non-hate predicting model. The models which will be tested are selected based on the type of the problem, as this is a text classification problem, classification models will be used in

this study. These modeling techniques can also be seen in the literature review chapter. Most research papers have mentioned the use of classification algorithms starting from basic linear models like logistic and advancing to neural network models. Also, the evaluation metrics mentioned in this chapter is used for evaluating classification models and have a brief description in the literature review. In the next chapter, implementation of this research methodology will be explained in detail

8. Research Plan

The chart below explains the timeline of the research plan for this study:

Figure 4: Research Plan Timeline

9. Risk and Contingency Plan

Following are a few risks involved and their solution for this project:

- It might be possible that the data on which the hate speech detection model would be used in the future is in a language that the model does not support. In this case, the classifier can

be retrained using a text embedding model that supports this particular language, and then this retrained model would be useful for detecting hate speech in that language.

- If the local computer is not sufficient for processing large amounts of data, then we would use GPU resources from Google Colab.

10. References

Aluru, S.S., Mathew, B., Saha, P. and Mukherjee, A., (2020) Deep Learning Models for Multilingual Hate Speech Detection. [online] Available at: http://arxiv.org/abs/2004.06465.

Badjatiya, P., Gupta, S., Gupta, M. and Varma, V., (2017) Deep learning for hate speech detection in tweets. In: *26th International World Wide Web Conference 2017, WWW 2017 Companion*. International World Wide Web Conferences Steering Committee, pp.759–760.

Davidson, T., Warmsley, D., Macy, M. and Weber, I., (2017) Automated Hate Speech Detection and the Problem of Offensive Language. [online] Available at: http://arxiv.org/abs/1703.04009.

Gaydhani, A., Doma, V., Kendre, S. and Bhagwat, L., (2018) Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach. [online] Available at: http://arxiv.org/abs/1809.08651.

de Gibert, O., Perez, N., García-Pablos, A. and Cuadros, M., (2018) Hate Speech Dataset from a White Supremacy Forum. [online] Available at: http://arxiv.org/abs/1809.04444.

Kamble, S. and Joshi, A., (2018) Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models. [online] Available at: http://arxiv.org/abs/1811.05145.

Kapil', P., Ekbal', A. and Das, D., (n.d.) *Investigating Deep Learning Approaches for Hate Speech Detection in Social Media*.

Lee, Y., Yoon, S. and Jung, K., (2018) Comparative Studies of Detecting Abusive Language on Twitter. [online] Available at: http://arxiv.org/abs/1808.10245.

Mathew, B., Saha, P., Yimam, S.M., Biemann, C., Goyal, P. and Mukherjee, A., (2020) HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. [online] Available at: http://arxiv.org/abs/2012.10289.

Pamungkas, E.W., Basile, V. and Patti, V., (2021) A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, 584, p.102544.

Plaza-del-Arco, F.M., Molina-González, M.D., Ureña-López, L.A. and Martín-Valdivia, M.T., (2021) Comparing pre-trained language models for Spanish hate speech detection. *Expert Systems with Applications*, 166, p.114120.

Ranasinghe, T. and Zampieri, M., (2020) Multilingual Offensive Language Identification with Cross-lingual Embeddings. [online] Available at: http://arxiv.org/abs/2010.05324.

Vijayaraghavan, P., Larochelle, H. and Roy, D., (2021) Interpretable Multi-Modal Hate Speech Detection. [online] Available at: http://arxiv.org/abs/2103.01616.

Waseem, Z. and Hovy, D., (n.d.) *Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter*. [online] Available at: http://github.com/zeerakw/hatespeech.

Zhang, Z. and Luo, L., (2018) Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. [online] Available at: http://arxiv.org/abs/1803.03662.