# Detection of Extreme Political Sentiments in Pakistan on Social Media

By

**Hafiza Rabail Mushtaq**

**Fall 2019- MS(IS) 00000318564**

Supervisor

**Dr. Sana Qadir**

**Department of Computing**

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Security (MS IS).

In

School of Electrical Engineering & Computer Science (SEECS) ,

National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(June 2023)

# THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Detection of Political Extremism in Pakistan on Social Media" written by HAFIZA RABAIL MUSHTAQ, (Registration No 00000318564), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: Sana Qadir

Name of Advisor: Dr. Sana Qadir

Date: 20-Jun-2023

HoD/Associate Dean:_

Date:

Signature (Dean/Principal):

Date:

# Approval

It is certified that the contents and form of the thesis entitled "Detection of Political Extremism in Pakistan on Social Media" submitted by HAFIZA RABAIL MUSHTAQ have been found satisfactory for the requirement of the degree

Advisor : Dr. Sana Qadir

Signature: Sana Qadir ________

Date: 20-Jun-2023

Committee Member 1:Dr. Razi Arshad

Signature:

20-Jun-2023

Committee Member 2:Dr. Rabia Irfan

Signature: ______________________

Date: 20-Jun-2023

Signature: ______________________

Date: ______________________

# Dedication

This thesis work is dedicated to all the children who do not have access to quality education. It is especially dedicated to young girls whose future prospects are often limited by their gender.

# Certificate of Originality

I hereby declare that this submission titled "Detection of Political Extremism in Pakistan on Social Media" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: HAFIZA RABAIL MUSHTAQ

Student Signature: _______________

# Certificate of Originality

I hereby declare that this submission is my own work and I am the sole author of all material contained therein. The work presented in this thesis is free from any plagiarism and has not been submitted earlier for any degree or diploma at any other institution. The facts, figures, ideas, and contributions of others used in this research work have been appropriately acknowledged and referenced in the bibliography. I am solely responsible for any errors or omissions presented in the thesis. I affirm that the academic content presented in this thesis is solely the outcome of my personal efforts, apart from the aid extended by others in project design and conception or in matters of style, language, and presentation, which have been duly acknowledged.

Author Name: **Hafiza Rabail Mushtaq**

Signature: ____________________

# Acknowledgments

Praise be to Allah (S.W.A), the Creator and Sustainer of the Universe, Who possesses the sole authority to bestow honor and dishonor according to His will. Indeed, no action can take place without His permission. From the moment of my enrollment at NUST until my departure, Allah was the constant source of blessings, guiding me toward success and providing me with opportunities. I am deeply grateful for His countless bounties throughout my research journey.

**Hafiza Rabail Mushtaq**

# Contents

# List of Tables

# List of Figures

# Abstract

The proliferation of political extremism on social media has adverse effects on not only the individuals who are targeted but also on society at large. It causes great damage to the hosting platform as well where such content is being shared. Even though notable research work has been done on sentiment analysis and classification in both academia and industry, an effective and robust tool to detect and classify political extremism on various social media platforms is still a challenge. Previous research work had largely focused on detecting general hate speech on social media via binary classification. But, considering the diverse nature of extremism, binary classification does not suffice the purpose. In this research, we have studied existing solutions and after finding their limitations, we have developed a multi-class and multi-lingual model that detects and distinguishes between neutral, moderate, and strong political extremist content. For training our model, we collected a data set of around 10,000 tweets from prominent political parties and politicians in Pakistan. We used the latest pre-trained BERT model and machine learning classifiers like Support Vector Machine, Random Forest, Naıve-Bayes, and Stochastic Gradient Descent to analyze and detect different classes of extremism. The highest accuracy we achieved is 89% in binary classification and 86% in multi-class classification using the Term Frequency-Inverse Document Frequency word embedding and SVM classifier. It is hoped that the results of this thesis will provide researchers and organizations with a viable solution to detect and classify extreme political sentiments.

# Chapter 1

# Introduction

The rise of online social networking platforms like Facebook and Twitter in the last two decades has resulted in an exponential increase in the amount of user-generated content available online. According to online statistics, Twitter has around 353.90 million users as of April 2023. It has roughly 237.8 million daily users which is 77.4% more than its daily active users in 2019 [1]. These values show a continuous upward trajectory which means more and more people are using these platforms to network with other people. As a result, there is a large amount of data available on this platform that can be used for various research purposes.

Owing to the popularity of these social networking sites, any information shared on these platforms can be disseminated to millions of people within seconds. As a result, there has been a significant increase in both the constructive sharing of ideas and the widespread dissemination of harmful and damaging content online. Although most of the potentially harmful incidents that have taken place on the internet like trolling or bullying have predated its existence, the reach and influence of the internet gives these incidents unprecedented power, especially in today's world of global communication.

As a result, the incidences of aggression because of heated conversations happening online has not remained a minor nuisance. Instead, it leads to criminal activities that affect a huge number of people. Therefore, it is critical that we take preventive measures to reduce or eliminate extremism, abusive behavior, and aggression.

## 1.1 Impact of Information Sharing through Social Media

In this digital aIt provides a place where people can express their thoughts freely, ge, sharing information on different social networking platforms has become a norm and most users share personal opinions, ideas, and thoughts publicly. This online data is a gold mine for analysts and researchers as they dig out valuable information which can be used for strategic decision-making [2].

On these platforms, people review others' opinions and openly show their agreement and disagreement. This results in a lot of positive information sharing and really helps people stay up-to-date on what is happening around the world. However, this excessive information sharing sometimes also leads to an unpleasant, uncivil, and hostile environment. This is especially true when the conversation is about sensitive topics such as religion, ethnicity, political affiliations, etc.

Twitter is one of the most favored and engaging social networking platforms today. It provides a place where people can express their thoughts freely, talk about their interests, and express their feelings, beliefs, and opinions [3]. On Twitter, people not only share their thoughts but also like to hear about other people's opinions and philosophies. Hence, they do follow other people and organizations on Twitter which they find interesting.

This social networking site is particularly famous for sharing political news, opinions, and thoughts. As a result, this platform is actively used by politicians across the globe where they connect with their potential voters and followers. Similarly, political enthusiasts and nationalists also use these platforms to stay on top of the political news. But unfortunately, these social media platforms are often misused to promote polarized beliefs related to political issues. People seek to share their views with a wide audience, encouraging their followers to share those beliefs. In response, nations are becoming more and more polarized. This does not stop there, some individuals may even take action to exhibit their commitment to the stance of the political party they follow. According to history, such situations often lead to alarming situations. In a recent study analysis by the Center for Strategic and International Studies [4], it was stated that right-wing extremists in the United States had been involved in 267 attacks and 97 deaths since 2015. While far-left extremists were involved in 66 attacks and 19 casualties.

If we talk about Pakistan in particular, it is becoming more and more polarized with time. In simple terms, polarization is following an ideology to an extreme level and staying focused on it even when it is contaminated [5]. Online statistics prove that most of these extreme sentiments come from social media. Users consume too much information online which is full of polarized beliefs. This leads to extremism in not only their conversations but also in their actions.

## 1.2 Use Cases of Sentiment Analysis

The process of extracting emotions from the written text of users is referred to as sentiment analysis. This can be done by extracting the unstructured information present in the text data and designing a model to detect and analyze the knowledge present in the data.

Currently, many organizations are making use of sentiment analysis models to better understand their consumer behavior towards a product and to analyze purchasing patterns. But sentiment analysis can be used for analyzing deeper societal issues as well such as detecting racism, sexism, hate speech, and other social behaviors. Identifying and analyzing these sentiments can help researchers uncover many different layers to the root cause of these behaviors, the frequency of their usage under different conditions, and the different triggering factors that impact these sentiments.

Research [6] has explicitly linked this increasingly aggressive public behavior to the widely available extremist content over the internet. Therefore, it is crucial for governments and enabling platforms to come up with solutions that can monitor and better regulate social media content because of its often negative impact on society.

## 1.3 Benefits of Sentiment Analysis

The information gained from the scientific experiments of sentiment analysis can be used by governments to enhance their policies, improvise laws, and amend important rulings. For example, identifying public reactions to traffic rules violations can help ministers amend traffic laws and improve the city's traffic conditions. Recent research contributions in social psychology also point to the multi-faceted nature of prejudice, from overt behaviors such as explicit and direct discrimination to covert behaviors such as implicit and indirect discrimination [7]. Therefore, the results from sentiment analysis experiments can be used by psychologists and psychiatrists to better understand human behavior, the causes and triggers behind them, and also their consequences on broader society.

## 1.4 Definitions

In general, extremism refers to a speech type that is considered by most people to be far outside acceptable, mainstream attitudes, and societal ethics [8]. While extremist content refers to social media content that is considered by most people to be far outside acceptable, mainstream attitudes, and societal ethics [9]. Kim et al [10] argue that definitions of uncivil or extremist content, as well as its online implications and consequences, often vary according to different studies.

Extremism is a rising issue across the world. Political extremism in particular has a much stronger impact on societies. It helps promote radicalization and triggers violent behavior in societies. The extreme sentiments used by leading politicians in their tweets suggest that they use these social channels to fulfill their personal agendas. The result is that this in-civil and abusive behavior has become a norm and a part of our daily interactions [11].

It is surprising that scholars have not used a more nuanced definition of extremism to better distinguish between different types of extremist content. As that can enable researchers to better comprehend the varying dynamics of extremism, which may diverge across different manifestations. This will also help them investigate radicalization processes and will allow them to examine whether individuals progress from moderately extreme to strongly extreme and under what conditions.

## 1.5 Problem Statement

As discussed above, the prevalence, effects, and spread of extremist content over social media have become an alarming situation. Digital media offer individuals an outlet to talk through and share their opinions about social and political issues. These discussions often involve language that contributes to creating a destructive and toxic environment. It can lead to various social problems like promoting trolling behavior, spreading hate and disinformation, and normalizing abusive language.

Political extremism has an even stronger impact on nations. It can greatly impact political campaigns, upcoming elections, and even the potential aftermath of the election. Online statistics [12] suggest that major social and political events trigger a shift in language toward negativity, incivility, and abusiveness. There are different existing solutions provided and proposed by both researchers and industries to detect different types of sentiments present in the data that is shared online by the masses. Most solutions detect and classify sentiments in binary classification [13] only but because of the varied nature of extreme sentiments and language complexities, it is not possible to detect the correct intensity of sentiments in textual data using binary classification. Some solutions are also available for multi-label classification [14] but they are using English language data only and are limited to a particular type of extremism. To the best of our knowledge, no existing solutions detect multi-lingual and multi-label extreme political sentiments.

In this research work, we have proposed a solution to detect extreme political sentiments in Twitter data of leading Pakistani politicians and political parties using machine learning classification algorithms and the latest word embedding techniques. As discussed earlier, the Twitter platform has become one of the most popular social networking sites among politicians where they share political views and news which are mostly in local languages. Therefore, there is a dire need for an efficient solution to detect multi-lingual political Twitter data with extreme sentiments and categorize it into multi-class labels.

## 1.6 Research Objectives

This research work proposed an effective and robust solution to detect extreme political sentiments. The main objectives of this research include:

- Collecting tweets data from leading political parties and politicians of Pakistan
- Training a machine learning based model to detect and classify multi-lingual and multi-class extreme political sentiments
- Generation of a robust and effective model to detect political extremism to be used by organizations and researchers

## 1.7 Thesis Motivation

So far, we have discussed the implications of extremist language on social media including how it is used to engage and radicalize those within online communities. One possible strategy for coping with online aggression is to review and approve user-generated content manually. However, the rate and volume of new user-generated content on the web have made manual methods of moderation and intervention virtually impossible. As a result, the possibility of semi-automatic solutions to detect and identify such uncivil behavior has become important and it has attracted significant attention from researchers in the past few years. All these factors contributed to the motivation behind this research work.

## 1.8 Thesis Organization

This thesis is organized into seven chapters including this introductory chapter. Chapter 2 provides a detailed literature review by explaining the existing techniques for data collection, data annotation, data pre-processing, and different machine learning algorithms used for sentiment analysis. We also discussed their limitations and weaknesses. Chapter 3 gives an overview of the tools and techniques used for the implemented research methodology. In Chapter 4, we have discussed the setup for data collection, data translation, data annotation, and data pre-processing. In Chapter 5, we have discussed the complete setup of model training. We also highlighted the tools and libraries used to train machine learning algorithms. We continue in Chapter 6 by explaining model evaluation and results. This chapter also compares the performance of our model with existing research work. Finally, in Chapter 7, we conclude this research study and highlight possible future work that could be carried out based on the findings from this research work.

# Chapter 2

# Literature Review

In this chapter, all of the important research work primarily related to this study is discussed. The main focus was put on covering recent research studies. We also discussed the advantages and disadvantages of each of those studies

## 2.1 Approaches for Sentiment Analysis

There are various approaches that have been used for sentiment analysis on text data. The approach that works best depends on the nature and type of the data as well as the platform to be used for sentiment analysis.

Most of the previous research works have used either lexicon-based analysis or machine-learning algorithms for detecting and analyzing sentiment. In machine learning techniques, the data is usually classified after converting the text data into vector format [15]. On the contrary, lexicon-based approaches classify the text data by making use of a dictionary lookup method. During this text classification process, it calculates document or sentence-level sentiment polarity with the help of lexicon databases for processing text data. Some of the well-known lexical databases are WordNet, Treebanks, SentiWordNets, etc.

### 2.1.1 Machine Learning Approach

Machine learning is the most classical approach for sentiment analysis. With this technique, each word from text data is represented in the form of vectors. This data is then analyzed using different machine-learning algorithms such as Support Vector Machine (SVM), Random Forest (RF), Naive-Bayes (NB), Stochastic Gradient Descent (SGD), etc.

### 2.1.2 Lexicon Based Approach

The lexicon-based approach establishes the sentiment of a word by using lexical databases. In this approach, a score is obtained for each word in the sentences of text data. This score is then annotated by using the features available in the lexical database. It determines the polarity of a text based on a set of words. Each of those words is annotated with a weight that is extracted to determine the overall sentiment of the text. But it is critical to pre-process the text data before assigning weights to the words for optimal results.

Earlier approaches to sentiment analysis mostly determined the sentiment of a word based on lexical dictionary-based analysis. But in today's digital world, the content shared by most users contains slang, and abbreviation, and is incorrect grammatically. This is where Deep Neural Networks in Natural Language Processing (NLP) help as more advanced models can be built using neural networks.

## 2.2 Word Embedding

To train a machine learning model, the text data first needs to be transformed into vector representations. There are various methods to convert text data into vectors, some of the most common methods include Bag of Words (BoW) [16], Term Frequency-Inverse Document Frequency (TFIDF) [17], Word2Vec [18], etc. All of these methods are known as traditional word embedding techniques now.

The more advanced word embedding techniques include BERT [19], mBERT [20], GloVe [21], ELMo [22], etc. These modern words embedding methods depend on a neural network architecture instead of basic n-gram methods.

### 2.2.1 Term Frequency-Inverse Document Frequency

For this research work, we have primarily used TF-IDF and BERT for word embeddings. TF-IDF comprises two metrics, namely Term Frequency and Inverse Document Frequency. It works on a statistical measure of finding word relevance in a single document or several documents which is referred to as a corpus.

The Term Frequency (TF) determines the frequency of words in a particular document. In simpler terms, it determines the number of times a particular word is present in a document.

The Inverse Document Frequency (IDF) calculates the rarity of words in a document. TF and IDF are calculated as:

TF = (Number of repetitions of a word in a document) / (Total number of words in a document)

IDF = Log [(Total number of documents in the corpus) / (Number of documents containing the word)]

IDF is given more importance over TF because even though TF determines the most commonly occurring words, the IDF score focused on the rarely used words in a document or a corpus that may hold significant contextual information [23].

### 2.2.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT model makes use of transformers, which is an attention mechanism that learns contextual relations between words present in text data. Transformers consist of two separate operations, an encoder, and a decoder. The encoder reads the text input while the decoder produces the result/prediction of the task.

Unlike the traditional directional models, which read the text sequentially i.e. from left to right and right to left, the transformer encoder reads the complete text data simultaneously. Therefore, it is named bidirectional, although it would be more appropriate to call it non-directional. This characteristic enables the BERT model to learn a word's context from all of its surroundings i.e. from both the right and the left of the word [24].

## 2.3 Related Work

Previous work in the related field has demonstrated both the challenges and potential of developing a classification system that differentiates between moderate and strong extremist content in general. To the best of our knowledge, there is no prior research that has focused specifically on detecting political extremism. Also, most previous research works on sentiment analysis have focused only on the binary classification of data.

Some existing research on multi-class classification has concentrated on differentiating between different targets of extremism instead of understanding the differences in their intensity. It is challenging to perform both tasks simultaneously; differentiating between different classes of extremism (such as neutral, moderate, and strong) while also performing target-relevant training. Moreover, classifying tweets based on their content instead of the target causes further challenges as there is not much difference between classes, moderate and strong because these extremist tweets mostly use similar grammatical structures, keywords, and non-verbal features. Nevertheless, previous research is still relevant to this discussion as most of them have used data from social networking sites for sentiment analysis. These research studies are described one by one below:

### 2.3.1 Studies Based on Detection of Islamophobic Content/Hate Speech

Bertie et al [25] train a model to classify between weak Islamophobic, strong Islamophobic, and not Islamophobic content on Twitter. They trained their model using a data set containing 109,488 tweets that were originated by far-right Twitter accounts in 2017. The Twitter data was in the English language only. The authors observed that in the data they collected, Weak Islamophobic tweets were prevalent adding up to 36,963 compared to Strong Islamophobic tweets which were 14,895 in number.

The authors different machine-learning classifiers to test their model including NB, RF, Logistic Regression (LR), Decision Tree, SVM, and also a deep learning classifier. Their main input feature was a Global Vectors for Words Representations (gloVe) word embedding model which was trained using a newly collected data set of 140 million tweets. In their results, they mention that this gloVe word embedding outperforms a basic word embedding model by 5.9%. They also state that a one-against-one SVM classifier outperformed a deep learning algorithm unexpectedly.

From their experiments, the highest accuracy they achieved is 72.17% using an SVM model while the balanced accuracy was 83%. Their model works well overall but struggles to distinguish between Weak Islamophobic speech from both Strong Islamophobic and Not Islamophobic content. In the future, they want to improve their model's performance by increasing the training data size. They also want to engineer additional input features to further analyze their model's behavior.

Khan et al [26] extracted their data set from an online crowd-sourced database called Hatebase [27]. The dataset was collected mainly from India, but they did not select any particular region from within India to collect their data. Their data set consisted of 8,438 English-language tweets and 8,790 Hindi-language tweets. Then, they classified their data set into multi-label classifications which include Islamophobic, Neither about Islam nor Islamophobic, and About Islam but not Islamophobic.

Since their data was multi-lingual, they used Google NMT to detect and translate the non-English data to English to make their system language-agnostic. For their experiment, they used three distinct word embedding models including BERT, Word2Vec, and GloVe. They also used the traditional n-gram methods namely TF-IDF and BoW for word embeddings. Then, they used different machine-learning classifiers with each of the above word embedding techniques. The traditional n-gram embeddings were tested with machine learning classifiers SVM and RF. Word2Vec and GloVe word embeddings were tested with deep neural networks algorithms like Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM). While the BERT embeddings were implemented within the BERT model.

All of the models they created were tested on English data only. During the experiments, some of the sentences remain unchanged after going through Google NMT. So, the model's performance highly depends on Google translator API.

The two highest-performing classifiers were BERT and SVM. The SVM classifier provided the highest accuracy on English as well as translated English data. While mBERT gave the highest accuracy for untranslated Hindi data. As mBERT is trained on multi-lingual data, it performs well with untranslated Hindi data as compared to other classifiers they used in their experiment. Their overall best-performing model is BERT with 10-fold cross-validation accuracy of 96.21% for English data, 94.66% for translated English Data, and 94.17% for Hindi data. While their most poor-performing classifier was LSTM.

In the future, the authors want to experiment with more text classification models like Projection Attention Neural Network for Document Classification On-Device (PRADO) and Projection-Based Modelling in Quasi-Recurrent Neural Networks (pQRNN) which is an extension of the PRADO model. They also want to add more languages in their data set to support further research on Islamophobia as well as contribute to identifying other types of hate speech on social networking platforms.

Shervin et al [28] examine methods to detect hate speech in social media content. They also want to differentiate hate speech from general profanity on social media. The authors wish to develop a lexical baseline for this experiment by executing supervised classification methods on a publicly available and annotated data set for a similar purpose. The data set used is available online on a website called CrowdFlower [29]. Data was in the English language only and had a total of 14,509 tweets.

To detect hate speech, the authors classified and annotated the data set into three categories which are Hate, Offensive, and OK. For feature extraction, the model used character n-grams, word skip grams, and word n-grams. For the machine-learning experiment, the authors applied their data to a linear SVM classifier to determine the model's performance.

In the analysis report, the authors state that distinguishing between hate speech and generally offensive content is the most difficult task in this experiment. As hate speech is confused with offensive content. The authors mentioned that the non-offensive class (OK class) achieved the best results, as most of the samples from the data set were classified correctly for this class by their model in the testing phase.

The best result the authors got was an accuracy of 78% using SVM. In the future, the authors want to analyze the performance of meta-learning and classifier ensembles for this task. Moreover, the authors also want to pursue a careful analysis of the most important and informative features of each class in their data set.

### 2.3.2 Studies Based on Detection of Aggression/Toxicity

Kumar et al [30] presented a report and conclusions from a shared task on Aggression Identification which was hosted as part of the workshop at COLING 2018 on Trolling, Aggression, and Cyberbullying (TRAC - 1). In this exercise, researchers were assigned to develop a machine learning classifier that can detect and classify between non-aggressive, covertly aggressive, and overtly aggressive text data.

For the experiment, researchers were given a data set of 15,000 Facebook comments and posts. The data set they were provided was multi-lingual and had tweets in both English language as well as Hindi language. A total of 30 teams presented their test runs for this task and 20 teams finally submitted their system descriptions that were a part of the workshop proceedings.

From all these submissions, the best model could achieve a weighted F1 score of 0.64 for both English and Hindi data. On unseen data, the scores were 0.60 for English data and 0.50 for Hindi data. The results from this experiment demonstrate that aggression identification is not an easy feat because of its varied nature and language complexities. The authors also mention that when the features are selected carefully, then the machine learning classifiers like SVM, LR, and RF perform much better than deep neural networks.

In the analysis report, the authors mention that there was not much difference in the neural network-based systems' performance as compared to other approaches. The highest-performing team achieved their results (F1 score = 0.64) using LSTM. But in conclusion, the authors mentioned that the low performance of their model might be because of some inconsistencies in the annotation of data provided to them as some participants raised this issue. They suggest that for optimal results in sentiment analysis, data should be annotated by multiple human annotators and should be cross-checked as part of quality assurance.

Jao et al [31] collected a new large-scale data set in the Brazilian Portuguese language. The text data was annotated as either non-toxic or toxic (binary classification) and then was also classified for different types of toxic language (multi-class classification). The authors demonstrated the method for data collection and data annotation where they tried to choose candidates from different demographic regions. They collected a data set of 21,000 tweets using the predefined hashtags and keywords that are highly likely to be used in toxic comments and also the tweets that mention the most influential Twitter users.

First, the authors restrict their experiment to the data set with binary classification only. Bag-of-Words (BoW) was used to demonstrate examples and an AutoML model to develop the baseline machine learning model. Using this combination of BoW+AutoML, the authors achieved a maximum of 74% macro F1-score.

Further, the experiment was done with different variations of the pre-trained BERT model to evaluate their model performance. The BERT variations they used include BERT, mBERT, mBERT-zero-shot, and mBERT-transfer. The model worked well in this case and the highest they achieved is a macro F-1 score of 76% using mBERT.

Then they experimented by classifying their data set into multi-label classifications in which they categorized their text data into multiple levels of toxicity. They used the mBERT model to conduct this experiment. Again, their baseline was a set of BoW+AutoML models which were trained with the help of binary relevance. But in this experiment, the authors only achieved about 20% average precision.

In the end, the authors highlighted that mono-lingual approaches for this experiment still outperform multi-lingual approaches. It was stated that larger data sets are needed to build more reliable models. The authors also mention that there are additional challenges that need to be resolved such as major class imbalance which is naturally there when dealing with real-world data in multi-label classification.

In the future, apart from working on minimizing data imbalance, they want to evaluate if combining classes with high divergences can help build more effective models. They also want to assess the impact of using unlabelled data in their model and see how it performs in semi-supervised techniques.

### 2.3.3 Critical Analysis

All the above research studies significantly contribute to the ongoing scientific research in the domain of sentiment analysis. However, 3 out of 5 above studies have used the data set which is already available online. This helps them to avoid the additional and time-consuming step of collecting data. Since the online available data set is gathered for a similar purpose, it is an ideal and balanced corpus for training and testing a sentiment analysis model. This makes their model less suitable for a practical solution of real-time extremism analysis. The 2 out of 5 above studies have used freshly acquired and annotated data and the performance of their model is impressive but there is still room for improvement in terms of common evaluation metrics such as accuracy and precision. More importantly, to the best of our knowledge, there are no existing solutions that focus on detecting multi-lingual extreme political sentiments. Table 2.1 provides a comparison of the strengths and weaknesses of the above-mentioned studies.

| Previous Research | Advantages | Disadvantages |
|---|---|---|
| Bertie et al | - Large data set<br>- Latest word embedding techniques<br>- Multi-label classification | - Doesn't work on multi-lingual data |
| Hina et al | - Multi-lingual and multi-label data<br>- Multiple word embedding techniques<br>- Good accuracy | - Use an already available online data set |
| Shervin et al | - Multi-label classification | - Use an already online available data set<br>- No latest word embedding techniques<br>- Only one ML algorithm used |
| Jao et al | - Original multi-lingual data set | - Only work for binary classification<br>- Only implemented BERT |
| Kumar et al | - Multi-lingual data | - Online available data set<br>- Low accuracy |

**Table 2.1:** Advantages and Disadvantages of Previous Work

## 2.4 Summary

In this section, we have discussed earlier approaches to performing sentiment analysis. Other than that, we have also highlighted existing research on detecting sentiments in both binary and multi-label classifications. Toward the end of this section, we highlighted the shortcomings of previous research work and discussed the lack of existing solutions to detect extreme political sentiments. In the next chapters, we will discuss the research methodology used during this research.

# Chapter 3

# Research Methodology

This chapter demonstrates the methodology used throughout this research work. We will highlight the steps that were taken to achieve the results. As machine learning approach has been used to detect extreme political sentiments, the steps involved in this experiment include all the typical machine learning processes such as data acquisition, data translation, data annotation, data pre-processing, feature engineering, model training, and model evaluation. This whole process is illustrated in Figure 3.1.

## 3.1 Research Methodology

The research experiment is started with collecting a tweets data set of 10,000 tweets from leading political parties in Pakistan. We used the official Twitter Developer API to collect that data. The tweets data was mostly in Roman Urdu and Urdu script while some tweets were in English as well.

### 3.1.1 Data Acquisition

There are two ways that researchers use to acquire data for research purposes. Either, collect data from one or multiple online available resources and compile it to generate a data set for the research experiment. Otherwise, it is also possible to use an already published data set by

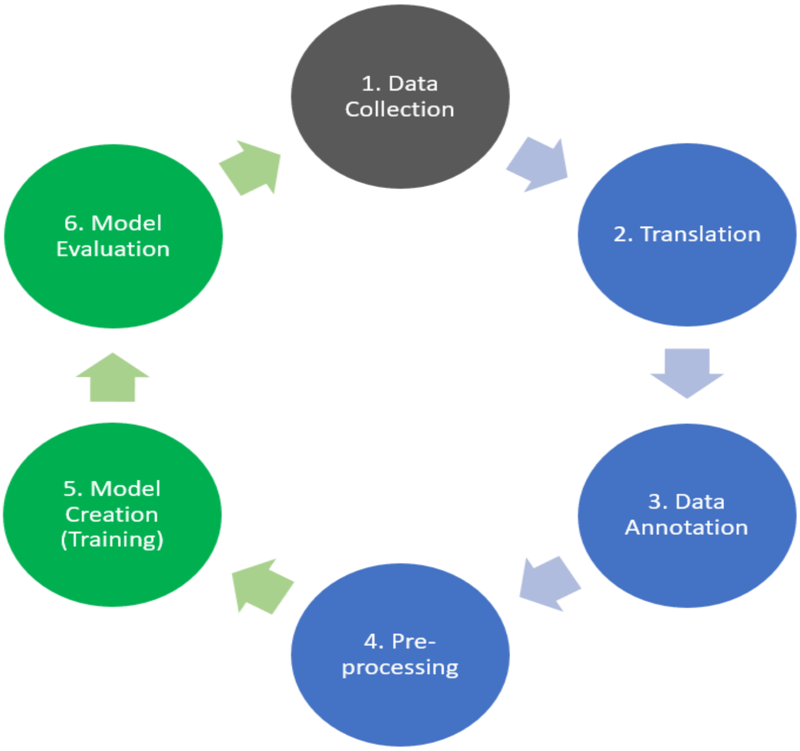


**Figure 3.1:** Research Methodology

other researchers to perform the intended research activities. For this experiment, we collected the required data set from Twitter using the official Twitter Developer API.

### 3.1.2 Data Translation

This step in the machine learning process is optional. It is used when experimenting with textual data that is not in standard language i.e. English. Since most of the word embedding techniques and machine learning algorithms work well with English language data only. Therefore, it is recommended to convert non-English data to English first before using it as input to train machine learning classifiers.

The Twitter data we acquired was mostly in Roman Urdu, some of the tweets were in Urdu script, while a few tweets were in the English language. We used Google NMT [32] to translate text data into English.

### 3.1.3 Data Annotation

Data annotation also known as data labeling is a crucial step in machine learning process especially when working with natural language. The data set can be annotated in binary classification or multi-class classification.

In binary classification, the text data is labeled in two categories only. The simplest example of binary classification is "Yes"/"No" and "Positive"/"Negative". While in multi-class classification, data is annotated in multiple labels to further fine-tune the output results.

For this experiment, the data set was annotated in both binary classifications as well as multi-class classifications.

### 3.1.4 Data Pre-Processing

Data pre-processing is another very important step in machine learning. Data sets often come in raw format and contain a lot of samples. As such, these samples have a lot of different features. But depending on the experiment, only a few of those features are relevant to that particular research work. Moreover, it's really difficult to understand data in its raw format. Therefore, in this pre-processing step, the raw data is cleansed and converted into a state which is understood by the machine learning algorithms easily. Sometimes, researchers use visualization methods such as histograms, word clouds, etc. to inspect the data and its features. The objective of this approach is to explore the data and its features in detail for further pre-processing.

In case of textual data, we need to clean our data extensively so that it has features that are relevant to the intended research work. The usual steps of cleaning text data involve:

- Lower-casing all words present in the text data.
- Removing URLs, emails, and hashtags.
- Removing special and accented characters.
- Removing stop words.

These pre-processing steps help ensure that the text data only contains features and information that are required for the intended experiment. It also helps bring out the right information present within each word of text data. Lastly, it also reduces the data file size, which decreases the computation time. This ultimately helps develop a lightweight and robust machine-learning model.

### 3.1.5 Feature Selection

Feature selection is the procedure to select features automatically or sometimes manually from your data that add the most value to the research output that one is interested in. Because if there are irrelevant features present in the text data, the machine learning model's performance and accuracy will decrease.

Having irrelevant features in the text data also causes several other issues as well such as:

- Over-fitting
- Unnecessary noise in the final model
- High time and space complexities

There are various techniques used to select features from a data set. Some of the frequently used methods include TF-IDF, Chi-Square, BoW, etc.

For this experiment, the TF-IDF vectorizer is used for feature selection of the input text data.

### 3.1.6 Model Training

The next step in a machine learning experiment is model training. In model training, the selected machine learning models are trained with training data to get the expected results. We input the processed data set into a machine learning algorithm, from which it learns the features and patterns. Consequently, the trained model learns from the input data and makes predictions for test data. The machine learning model keeps improving its performance as the training data size is increased. Different machine learning algorithms show different results for the same features and data set. So, before model training, an important step is to choose the right machine learning algorithm for a particular data set.

### 3.1.7 Model Evaluation

It is the final step of the machine learning process. Model evaluation of a machine learning classifier is conducted against the anticipated outcome or against the previously published similar work. During model training, the testing data is separated from the training data so that it is unseen for the machine learning model. If the testing data is same as the training data, it will give inaccurate results since the model is already aware of the data set's features and find the same patterns in it. This will lead to a faulty high level of accuracy.

So, to avoid misleading results, the model should be evaluated on unseen testing data. This way, the machine learning model will give justifiable accuracy and performance. In case, model's results do not meet the evaluation metrics, it is tuned. This fine-tuning of the model is done by choosing the right parameters of the selected algorithm. The best way to do that is to test different parameters on the model and select the best ones for model training. This step can help increase the performance of the model.

## 3.2 Tools and Technology

To conduct this research experiment, different tools and technologies have been used. Twitter data was used to train machine learning algorithms for detecting extreme political sentiments. The table below lists all the tools and technologies used in this research work.

| Tools / Technology | Usage in Implementation |
|---|---|
| Python | Python programming was used for writing different scripts for data acquisition, data translation, pre-processing, word embeddings, model training, and evaluation. |
| Numpy | This Python library is used to perform all mathematical operations required in research. |
| Pandas | This Python library is used to do all data processing. |
| NLTK | It is a suite of libraries used for natural language processing. |
| Scikit-learn | This library was used to perform most of machine learning implementations. |
| Tensorflow | It is a Python library to facilitate deep learning applications. We used it to train our BERT model. |
| Tweepy | It is a Twitter API for directly accessing Twitter data. |
| Tqdm | We used this library to show a progress bar for our BERT training model. |
| TF-IDF | It is a handy algorithm for text classification. We used it for word embedding. |
| Jupyter | It is a web-based interactive platform that we used for performing all our Python programming. |
| Twitter | It is a social media site. We used this platform to collect our data. |
| Google NMT | It is a Google API that is used to translate all non-English data to English. |

**Table 3.1:** Tools and Technologies

## 3.3 Summary

In this section, we describe the complete research methodology of conducting sentiment analysis using machine learning. All of the tools and technologies used for this research work were also highlighted. Additionally, we have also discussed how each of those tools and technologies were used within this experiment.

# Chapter 4

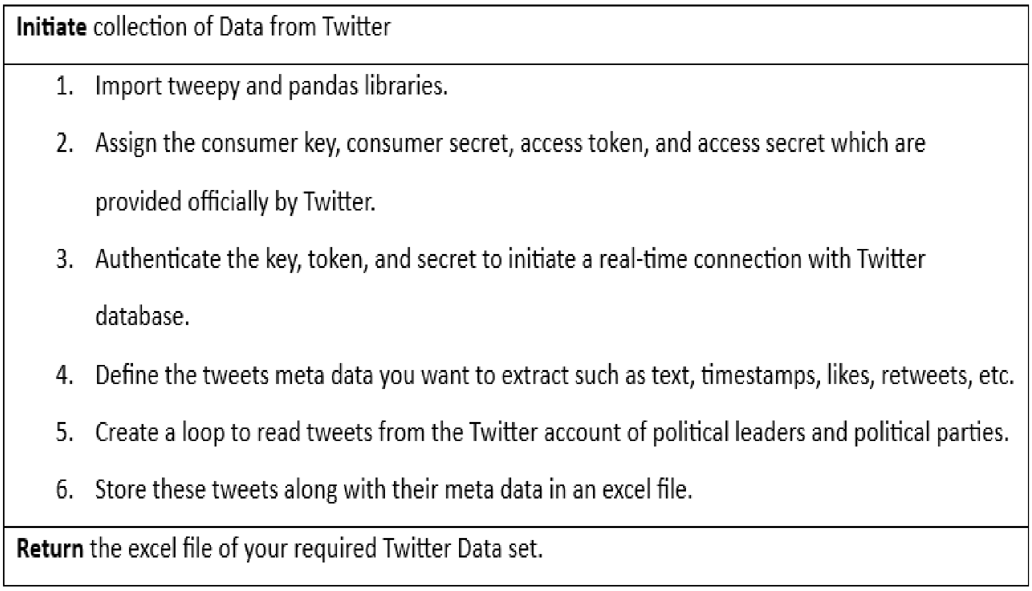
# Data Acquisition and Pre-processing

In this chapter, we discuss the step-by-step process through which our data set was created and processed, so it can be used as input to train machine learning classifiers. We have also discussed the in-depth processes of data collection from Twitter, Data Translation, Data Annotation, and Data Pre-processing. Additionally, the tools and techniques that were used during these processes are also highlighted.

## 4.1 Data Acquisition

Twitter is one of the top social networking sites to share views and opinions and has huge numbers of active users. It makes its Tweets data available to researchers across the world to facilitate ongoing research in different domains. For example, a lot of researchers worked on detecting hate speech on Twitter and analyzed its short and long-term impact on societies. Similarly, many researchers also worked on analyzing racism on this social networking site, they are interested in finding various triggers that cause racism, different origins of racism, how it varies over time, etc.

In this research work, Twitter data is used to detect extreme political sentiments from the tweets originated by the top Pakistani politicians and political parties. This data can be extracted from Twitter with the help of Twitter Official Developer API [33]. Python language was used to extract the Tweets data. We used Tweepy [34] and Pandas [35] libraries to collect the data set that was needed. In Figure 4.1, the algorithm of the data collection code has been highlighted.



**Initiate** collection of Data from Twitter

1. Import tweepy and pandas libraries.
2. Assign the consumer key, consumer secret, access token, and access secret which are provided officially by Twitter.
3. Authenticate the key, token, and secret to initiate a real-time connection with Twitter database.
4. Define the tweets meta data you want to extract such as text, timestamps, likes, retweets, etc.
5. Create a loop to read tweets from the Twitter account of political leaders and political parties.
6. Store these tweets along with their meta data in an excel file.

**Return** the excel file of your required Twitter Data set.

**Figure 4.1:** Algorithm for Data Collection from Twitter

We collected this tweets data in March 2022 and retrieved equal amounts of tweets from each political party and its politicians so that each party is represented equally in the data set and the results are not misleading.

After retrieving the data, all the metadata from the tweets was removed such as Twitter ID, user ID, geo-location, timestamps, etc. This is to make sure that the identity of the users who tweeted the post is not shared with annotators and the public audience.

## 4.2 Data Translation

Since the data was multi-lingual, it was translated into the English language to make this model language agnostic. For translation, Google Neural Network Machine Translator (NMT) [32] was used with the help of Google API. By default, when the translation of text data is requested, the

NMT detects the language and translates it. If the requested language translation pair is not supported by the NMT model, the phrase-based machine translation can be used to translate non-English data to English. In this research work, text data in both Roman Urdu and Urdu script were easily detected and translated by NMT.

An abstract view of the data translation process can be seen below:

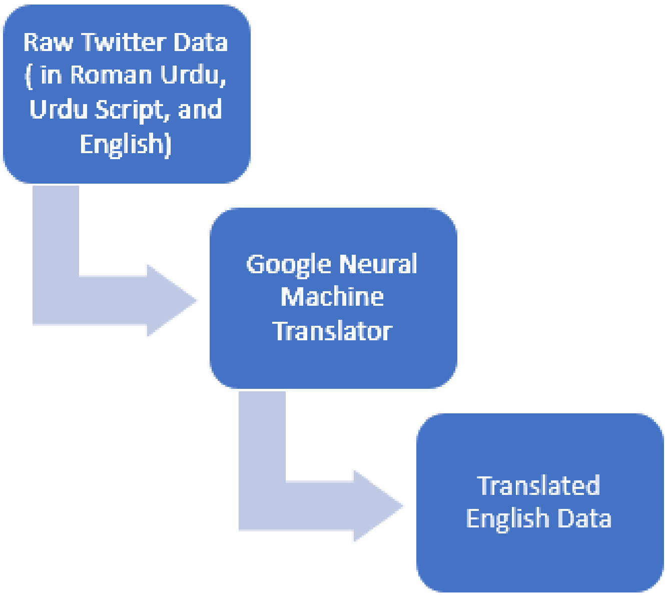

Raw Twitter Data ( in Roman Urdu, Urdu Script, and English)
Google Neural Machine Translator
Translated English Data

**Figure 4.2:** Data Translation Process Using Google NMT

## 4.3 Data Annotation

The data classification task involves distinguishing between neutral, moderate, and strong extremist content from the tweets data. Drawing on Merriam-Webster's [8] definition of extremism, we define strong extremist content as:

"Speech that contains direct accusations, offensive name-calling, and extreme mocking against particular individuals."

This can vary from accusing other politicians to be murdering people, to aiding in terrorism, money laundering, and looting. An example of strong extremist content is calling an individual to be funding terrorist activities and hiding/aiding its activities. Another example of strong extremist content is directly accusing the opposition party of murdering people. We define moderate extremist content as:

"Speech that contains moderately offensive name-calling, and trolling. And speech that contains direct accusations of poor leadership and corruption against a particular opposition party."

An example of moderate extremist content is accusing a particular party of corruption and calling it different names. Another example of moderate extremism is name-calling the politicians from the opposition party and mocking their actions publicly.

The word cloud library [36] from Python was used to visualize the most common words in our text data of 10,000 tweets. Word clouds are useful for envisioning common words from large text data. They are really useful when you need to:

- Identify the prominent themes or topics in a large data set.
- Understand the overall sentiment of the data set under process.
- Visualize the most commonly used words in a visually appealing manner.
- Inspect the patterns and trends from the textual data.

This library was used to visualize the most common words and get an idea of the prominent themes in the tweets data that was collected. Below, you can see the word cloud of that Twitter data set:

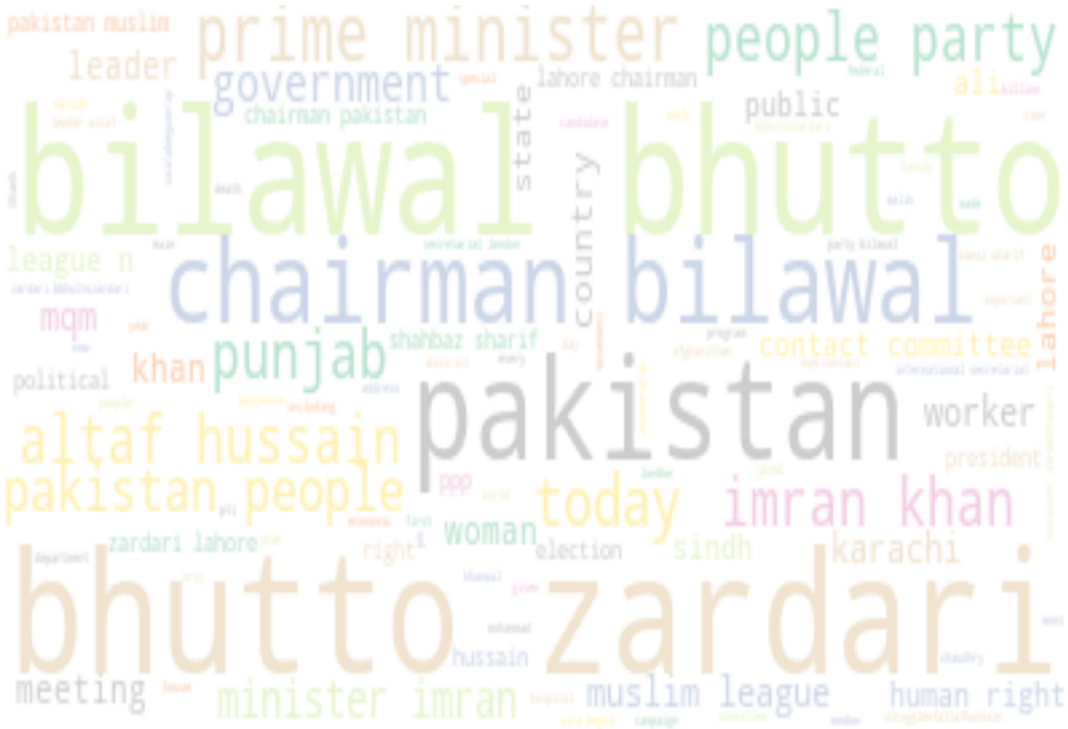
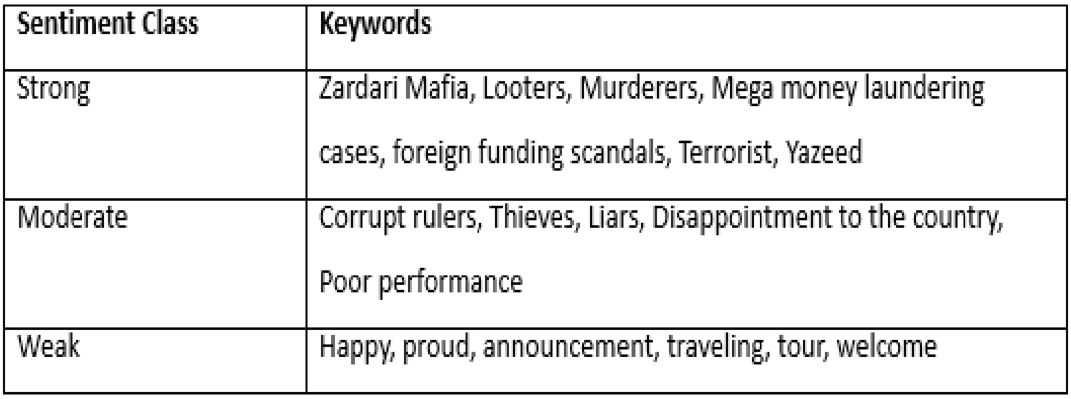


**Figure 4.3:** Word Cloud

Subtle extremist content is often more challenging to observe and may only partly depicts societal negativity. Strong extremism, however, can be easily spotted. Therefore, it is a little challenging to discriminate between these two classes of extremism. But, we shortlisted a few keywords that helped us annotate the data uniformly. An overview of the keywords that were used to classify each extremism class can be seen below:



| Sentiment Class | Keywords |
|---|---|
| Strong | Zardari Mafia, Looters, Murderers, Mega money laundering cases, foreign funding scandals, Terrorist, Yazeed |
| Moderate | Corrupt rulers, Thieves, Liars, Disappointment to the country, Poor performance |
| Weak | Happy, proud, announcement, traveling, tour, welcome |

**Figure 4.4:** Keywords Used for Annotation

For annotation, the data set was classified for both binary and multi-class classifications. First, the data was annotated for binary classification in which tweets data was classified into two classes named "Extremist" and "Not-extremist" classes.

Then, the data set was annotated for multi-class classification. The multi-class labels include neutral, moderate, and strongly extremist content. Below is an overview of the total number of tweet numbers that belong to each category in both binary and multi-class classification.

**Table 4.1:** Number of Tweets Corresponding to Binary Classification

| ***Category*** | ***Tweets*** |
|---|---|
| Not-extremist | 8,136 |
| Extremist | 1,878 |
| Total | 10,014 |

**Table 4.2:** Number of Tweets Corresponding to Multi-class Classification

| ***Category*** | ***Tweets*** |
|---|---|
| Neutral | 8,136 |
| Moderate | 1,511 |
| Strong | 367 |
| Total | 10,014 |

To avoid biases, we got the data annotated by 2 separate annotators without providing any tweet id or username attached to the tweets data. The annotation was carried out based on the above-mentioned keywords, definitions, and guidelines. In the graph below, it can be seen that neutral tweets dominate the data set, followed by moderately extreme. While strongly extreme tweets make up only a small part of this data set.

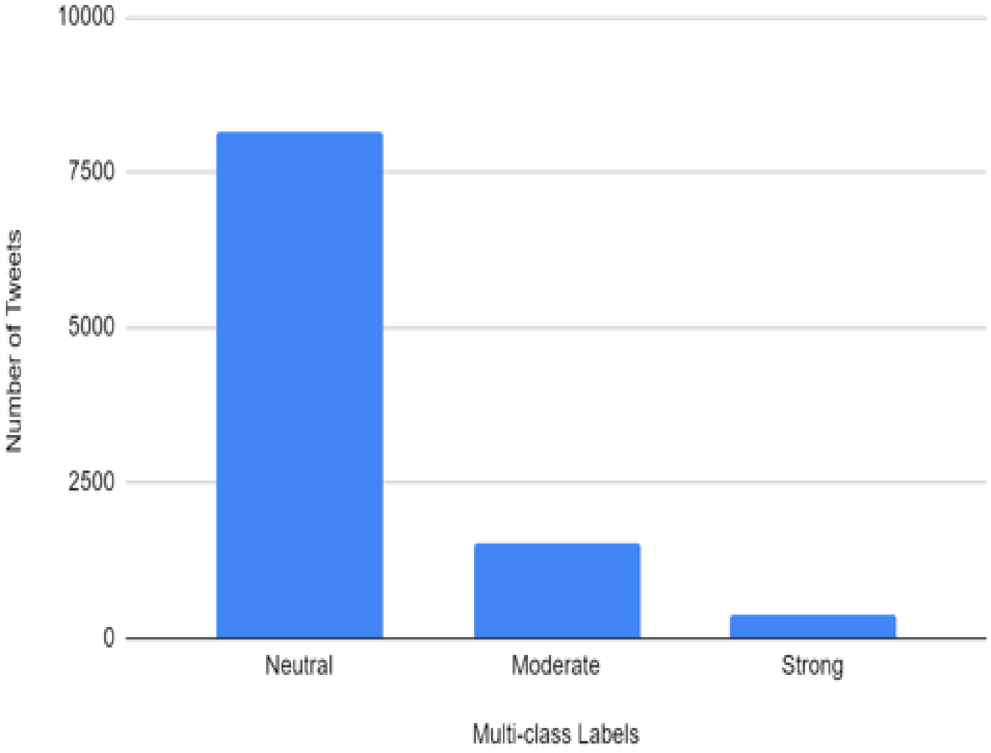
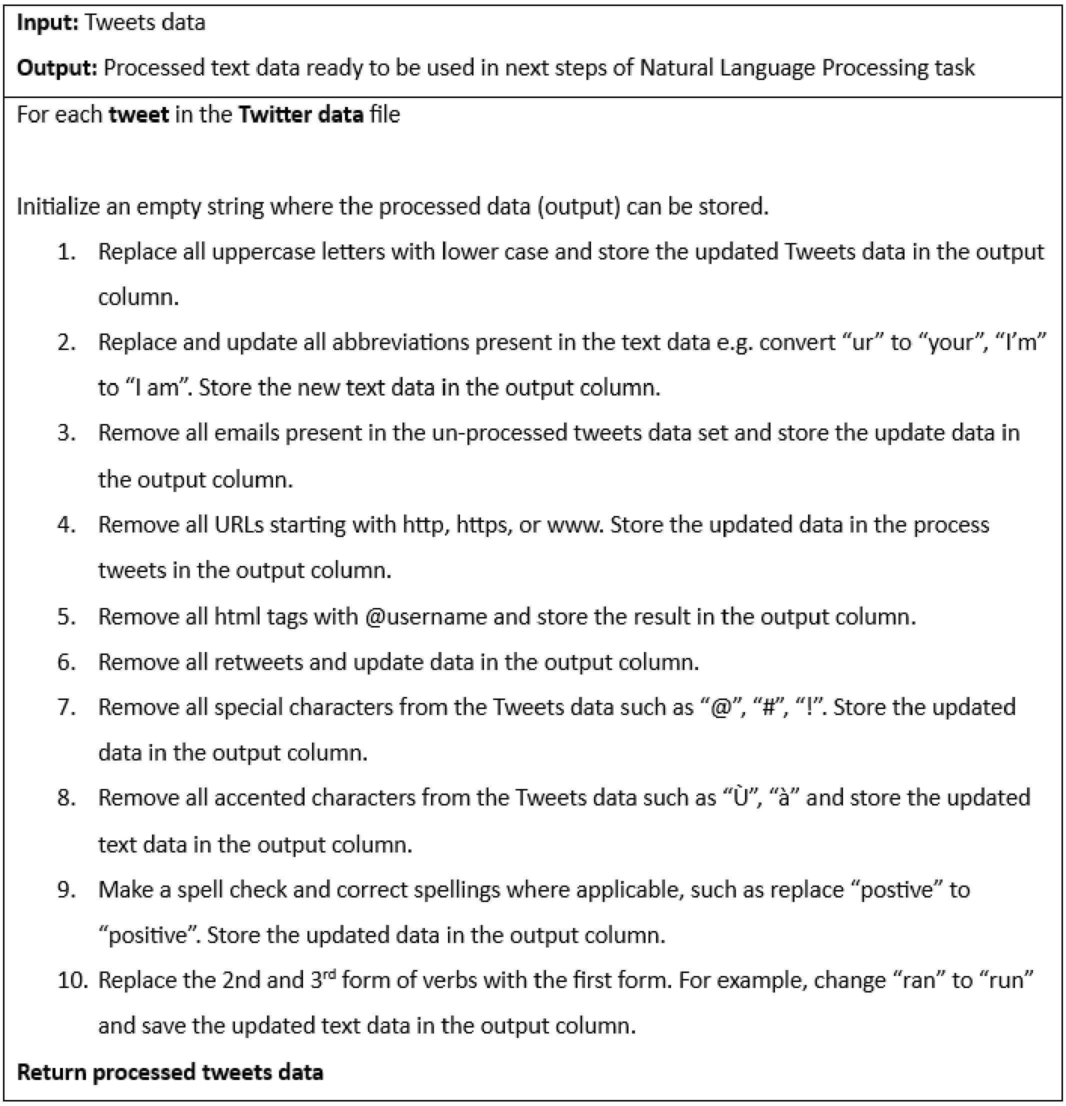


**Figure 4.5:** Data Set Overview in Multi-label Classification

## 4.4 Data Pre-processing

By taking into account the previous research work done on this, below is how the algorithm looks like for data pre-processing.



**Input:** Tweets data

**Output:** Processed text data ready to be used in next steps of Natural Language Processing task

For each **tweet** in the **Twitter data** file

Initialize an empty string where the processed data (output) can be stored.

1. Replace all uppercase letters with lower case and store the updated Tweets data in the output column.
2. Replace and update all abbreviations present in the text data e.g. convert “ur” to “your”, “I’m” to “I am”. Store the new text data in the output column.
3. Remove all emails present in the un-processed tweets data set and store the update data in the output column.
4. Remove all URLs starting with http, https, or www. Store the updated data in the process tweets in the output column.
5. Remove all html tags with @username and store the result in the output column.
6. Remove all retweets and update data in the output column.
7. Remove all special characters from the Tweets data such as “@”, “#”, “!”. Store the updated data in the output column.
8. Remove all accented characters from the Tweets data such as “Ù”, “à” and store the updated text data in the output column.
9. Make a spell check and correct spellings where applicable, such as replace “postive” to “positive”. Store the updated data in the output column.
10. Replace the 2nd and $3^{rd}$ form of verbs with the first form. For example, change “ran” to “run” and save the updated text data in the output column.

**Return processed tweets data**

**Figure 4.6:** Algorithm for Pre-processing of Twitter Data

In the first step of data pre-processing, all the abbreviations in the text data were replaced. For example, "ur" is converted to "your", and "I'm" is converted to "I am". This is to make sure that each word has a clear meaning which can be understood in the later steps. In the next step, all uppercase letters were updated to lowercase letters. As in further NLP, the data set does not need to be case-sensitive.

After that, all the emails present in the Twitter data were removed, as emails do not deliver any useful information and act as noise in the NLP. After removing emails, all the URLs were removed from our Twitter data as the information from URLs is insignificant. This is done using the command *ps.remove_urls(x)*.

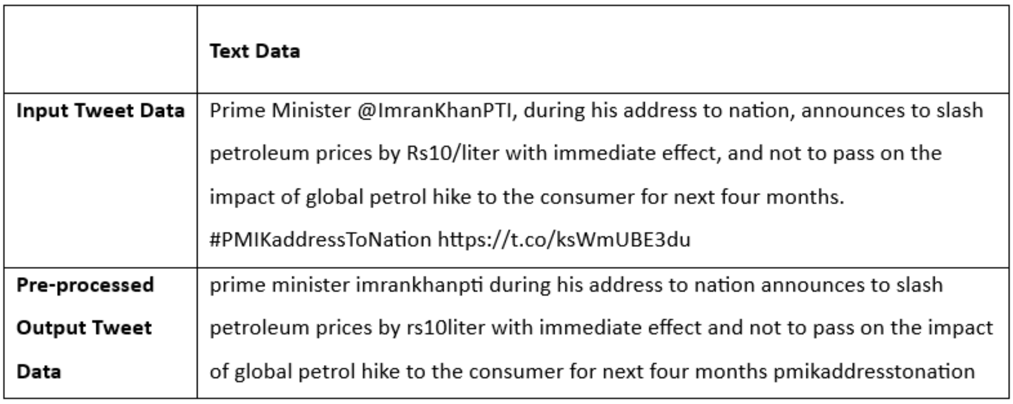
In the next step, all HTML tags were removed. People or organizations that are tagged in the tweets data do not contribute any meaningful information to this sentiment analysis experiment. After removing tags, the retweets information was removed as that information is also not needed for sentiment analysis of data.

Next, all the special characters were removed from the Twitter data. Special characters include "#", "@", "!", etc. These characters are also not needed in further NLP. After removing special characters, we removed all the accented characters as well. Accented characters include à, è, ì, ò, ù, À, È, Ì, Ò, Ù, etc.

In the next step, a quick spelling check was done on the tweets data. The spelling of all the misspelled words was corrected so that each word is interpreted correctly in NLP.

Lastly, all the verbs were converted into the first form from their 2nd or 3rd form. This also helps ensure that the data is perfectly readable in further NLP.

Below is an example of the input tweet data, and the output tweet data after all the pre-processing steps.



| | **Text Data** |
|---|---|
| **Input Tweet Data** | Prime Minister @ImranKhanPTI, during his address to nation, announces to slash petroleum prices by Rs10/liter with immediate effect, and not to pass on the impact of global petrol hike to the consumer for next four months. #PMIKaddressToNation https://t.co/ksWmUBE3du |
| **Pre-processed Output Tweet Data** | prime minister imrankhanpti during his address to nation announces to slash petroleum prices by rs10liter with immediate effect and not to pass on the impact of global petrol hike to the consumer for next four months pmikaddresstonation |

**Figure 4.7:** An Example of Output Tweet Data After Pre-processing

## 4.5 Summary

In this chapter, the step-by-step process of collecting and preparing the Twitter data set has been discussed. The pre-processing is done so that the text set is ready to be used as input to train machine learning algorithms. In the next chapter, further steps in natural language processing and training the sentiment analysis model are discussed.

# Chapter 5

# Feature Selection and Model Training

In this chapter, the complete methodology to train machine learning classifiers is discussed. The tools and libraries that were used during this experiment are also highlighted.

## 5.1 Feature Selection

So far, data collection, data annotation (in which each tweet is appended with its respective sentiment), data translation, and data pre-processing have been discussed. Different machine learning algorithms that are more frequently used by researchers for linguistic data analysis are also explored.

Now, the Twitter data set which is labeled with the sentiment of each tweet will be used by word embedding models like TF-IDF and BERT. Later, this data will be used as an input to train the machine learning classifiers. In the end, we will measure the accuracy, precision, and reliability of the newly-trained machine learning algorithms with testing data.

The traditional n-gram embeddings were tested using four different machine learning classifiers which include SVM [37], RF [38], NB [39], and Stochastic Gradient Descent (SGD) [40]. While the BERT embedding is tested within the pre-trained BERT model [41].

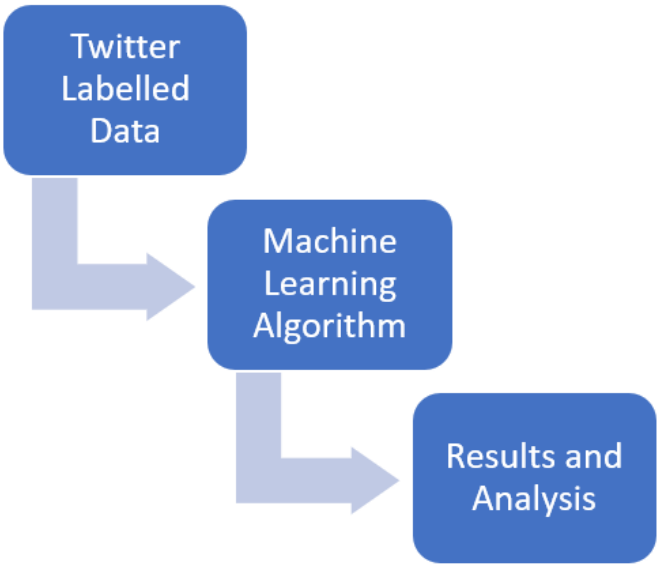


**Figure 5.1:** An Abstract View of Sentiment Analysis Using Machine Learning

An abstract view of the model that detects extreme political sentiments using TF-IDF and machine learning classifiers can be seen in Figure 5.1.

## 5.2 Model Training

Now, the machine learning classifiers and a pre-trained BERT model will be trained on the processed data. The step-by-step process of model training is discussed below:

### 5.2.1 Implementation Using Python

Python language was used to train and test the machine-learning models for sentiment analysis. Since Python is an open-source library, it is interpretive and allows access to hundreds of standard libraries and resources available online which are required for application development.

Python is also very feasible for scripting and language processing because of its dynamic typing and sophisticated syntax [42].

Python language is particularly popular and suitable for sentiment analysis of text data. It has several libraries such as Natural Language Toolkit (NLTK) [43] and sci-kit-learn [44] which make it easy to process text data. NLTK is a popular library for text classification. It includes several pre-trained machine-learning classifiers that can be used for sentiment analysis. Similarly, sci-kit-learn is also extensively used for sentiment analysis. It supports many algorithms for text classification including SVM and LR [45].

Apart from that, for word embedding models like BERT, TensorFlow [46] is a popular library to perform sentiment analysis. It is an open-source library which is developed by Google to primarily facilitate deep learning applications. It accepts data in the form of multi-dimensional arrays of higher dimensions which are called tensors. This is especially suitable when handling larger data sets. With TensorFlow, various analyses can be easily performed such as word embeddings, image classification, audio and video recognition, etc. Moreover, with TensorFlow, it is possible to use word embedding models like BERT with just a few lines of code [47].

For this experiment, Python 3.9 and NLTK (Natural Language Toolkit) were used to process the linguistic data that was gathered from Twitter.

For the programming interface, Jupyter Notebook [48] was used which is a web-based interactive computing platform to develop open-source applications. This online platform is used to combine live codes, narrative text, equations, visualization, etc.

## 5.3 Experiment

For the experiment, 10 different models were created using different word embeddings and machine-learning classifier combinations. However, the main architecture of the sentiment analysis model stays the same and is represented below in Figure 5.2.

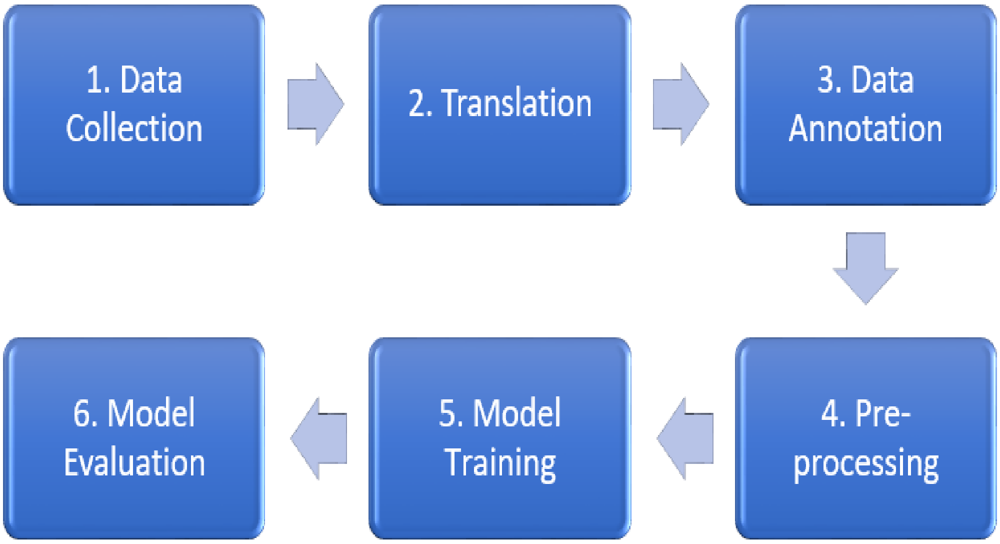


**Figure 5.2:** The Main Architecture of Political Extremism Detecting Model

The model architecture showcases different steps of our model. Input is provided in the form of text data frames of labeled tweets. Then, the NMT translator detects the text language and translates it into English. The translated data is then annotated in binary format and then in multi-class format which includes 3 categories. This annotated data is then processed to eliminate noise and refine meaning within the tweets. The next step in the model architecture is word embedding [49] in which different word embedding techniques were used to calculate the vector representations of words preset in the text data. The vectorized data is then passed to classifiers for training and testing.

The different Python libraries were used in the training and testing phase of our model architecture. These libraries include Scikit-learn, NLTK, TensorFlow, BERT tokenizer, Numpy, Pandas, Tqdm, etc. Based on word embeddings and machine learning classifiers, this experiment was divided into 10 tasks.

In task 1, TF-IDF word embedding having maximum feature of 5,000 is used with the SVM classifier for binary classification of data. In task 2, the same combination of word embedding and machine learning classifier is repeated for the multi-class classification of data. For both these experiments, the kernel is set to linear SVC [50] and the max iterations are set to 10,000. 80% of the data is used for training the model and the random state of the kernel is set to "0". The results of this model are analyzed using classification reports from sci-kit learn libraries [51].

In task 3, the TF-IDF vectorizer is used with the SGD classifier for the binary classification of data. In task 4, the same word embedding and classifier combination is repeated for the multi-class classification of data. The TF-IDF embedding is used with max features of 5,000. The random state is "0" and the training data size is 80%. In both these tests, the max iterations are set to 5,000.

In task 5, the TF-IDF vectorizer is used with the NB classifier for the binary classification of data. In task 6, the same experiment is repeated for the multi-class classification of data. In both task 5 and task 6, the kernel is set to multinomial [52]. The TF-IDF word embedding is used with a maximum of 5,000 features. The random state is "0" and the training data size is 80%.

In task 7, the TF-IDF word embedding is used again with the RF classifier for the binary classification of data. In task 8, the same steps are repeated as in task 7 but used the multi-class classification of data. In these 2 tasks, 200 trees are used to train the model. TF-IDF word embedding is used with a maximum of 5,000 features. Again, the random state is "0", and the training size is 80%. An abstract view of these 8 tasks can be seen in Figure 5.3.

Finally, in task 9, BERT embedding is used within the BERT model. The BERT experiment is repeated for the multi-class classification of data in task 10. In these 2 tasks, a BERT tokenizer is used to tokenize the data so that it can correspond to BERT's vocabulary. The pre-trained BERT model is fine-tuned with the inputs. For that, the data is split into batch sizes and shuffled for better results. In the last layer, the softmax activation function [53] was added.

An abstract overview of tasks 9 and 10 can be seen in Figure 5.4.

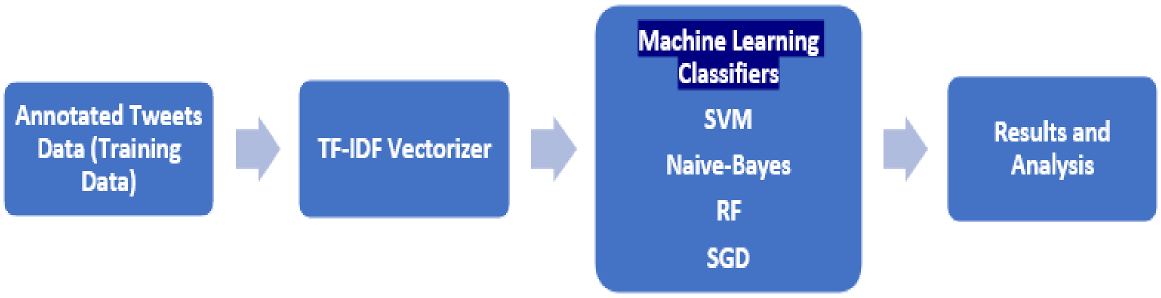

Annotated Tweets Data (Training Data)
TF-IDF Vectorizer
Machine Learning Classifiers
SVM
Naive-Bayes
RF
SGD
Results and Analysis

**Figure 5.3:** An Abstract Overview of Machine Learning Training Model

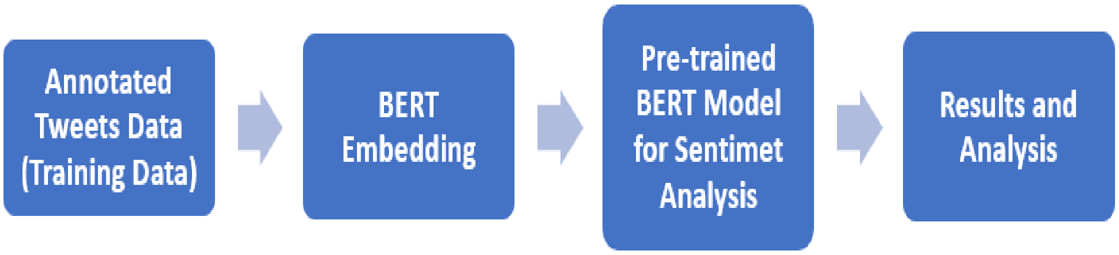

Annotated Tweets Data (Training Data)
BERT Embedding
Pre-trained BERT Model for Sentimet Analysis
Results and Analysis

**Figure 5.4:** An Abstract Overview of BERT Training Model

For all 10 experiments, 80% of the data was used to train the models for each classifier, while

20% data was used for testing the models. To determine the efficacy of these models, accuracy, precision, recall, and F1 score were calculated using the classification report and confusion matrix from the sci-kit-learn library.

## 5.4 Summary

In this chapter, all the steps involved in model training are discussed. The algorithm used for word embeddings and model training is also described. The different Python libraries used in this research experiment are also highlighted.

# Chapter 6

# Model Evaluation and Discussion

This chapter is about the trained model's evaluation and testing. The evaluation of the model is performed using different standard evaluation metrics. Toward the end of this chapter, the results obtained from these experiments are discussed. Lastly, the results are compared with existing research work.

## 6.1 Evaluation Metrics

For sentiment analysis, most research works have used the following evaluation metrics:

- Accuracy
- Precision
- Recall
- F1 score

The above metrics with their formulas are described below. But before we go into descriptions, let us understand a few abbreviations.

- True Positive = TP

- True Negative = TN
- False Positive = FP
- False Negative = FN

If the above terms are described in relevance to this research experiment - "Positive" means that a tweet contains extreme political sentiments while "Negative" means that a tweet does not contain extreme political sentiments.

### 6.1.1 True Positive

A prediction is true and positive when the predicted value is equivalent to the actual value. In this experiment, when the tweets with extreme political sentiments are labeled correctly, it is called true positive.

### 6.1.2 True Negative

A prediction is considered true negative when the predicted value is the same as the actual value. In this experiment, the true negative is a tweet that does not contain extreme political sentiments and the model labeled it correctly.

### 6.1.3 False Positive

A false positive is when the predicted value does not match with the actual value. For example, when a "Negative" (neutral tweet) is labeled as a "Positive" (tweet with extremist content)

### 6.1.4 False Negative

False Negative is when the predicted value does not match with the actual value. For example, when the actual value is "Positive", and the model labeled it as "Negative".

### 6.1.5 Accuracy

Accuracy is defined as the number of correct predictions divided by the total predictions. Its formulas can be seen below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1.1}$$

### 6.1.6 Precision

Precision is defined as the percentage of correct positive predictions. Its formula is given below:

$$Precision = \frac{TP}{TP + FP} \tag{6.1.2}$$

### 6.1.7 Recall

Recall is the percentage of actual positive values that were classified correctly. The formula for that is:

$$Recall = \frac{TP}{TP + FN} \tag{6.1.3}$$

### 6.1.8 F-1 Score

The F-1 score is computed by taking the harmonic mean of a classifier's precision and recall. The formula for that is given below:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.1.4}$$

### 6.1.9 Confusion Matrix

A confusion matrix is a popular measure used to visualize classification problems. Both binary classification and multi-class classification can be visualized through this matrix. It is a c*c matrix where c is the number of classes. The generated matrix compares the predicted values of a machine learning model to their actual values.

## 6.2 Results and Evaluation

The tables below depict the performance of the models concerning different word embeddings and classifiers we have used.

Table 6.1 shows the results for the binary classification of data. While Table 6.2 shows the performance of the models for multi-class classification.

**Table 6.1:** Performance of Classifiers for Binary Classification

| ***Classifiers*** | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1 score*** |
|---|---|---|---|---|
| SVM | 89% | 84% | 80% | 82% |
| Naïve-Bayes | 87% | 83% | 73% | 77% |
| SGD | 88% | 83% | 80% | 82% |
| Random Forest | 86% | 85% | 67% | 71% |
| BERT | 86% | 82% | 72% | 79% |

**Table 6.2:** Performance of Classifiers for Multi-Class Classification

| ***Classifiers*** | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1 score*** |
|---|---|---|---|---|
| SVM | 86% | 69% | 56% | 59% |
| Naïve-Bayes | 85% | 83% | 45% | 48% |
| SGD | 86% | 71% | 53% | 56% |
| Random Forest | 83% | 74% | 41% | 43% |
| BERT | 84% | 78% | 78% | 69% |

### 6.2.1 Results for Binary Classification

The overall performance of these models is better with binary classification compared to the multi-class classification of data. This is in line with the initial statement based on existing research that it is harder to categorize extremism because of its varied nature. The highest accuracy achieved for binary classification is 89% using TF-IDF word embedding and SVM classifier. The second highest-performing model in binary classification is TF-IDF word embedding with SGD classifier, in which an accuracy of 88% and a precision of 83% is achieved.

After SVM and SGD, the third best model for sentiment analysis in the case of binary classification of data is NB. An accuracy of 87% and precision of 83% was achieved using TF-IDF word embedding with the NB classifier.

The last two models in terms of performance for binary classification of data are pre-trained BERT with BERT embedding and RF classifier with TF-IDF word embedding. An accuracy of 86% was achieved using both these models, however, the precision is 85% using RF compared to 82% using BERT.

Below are some of the snippets of the performance of these models for binary classification visualized through the confusion matrix.

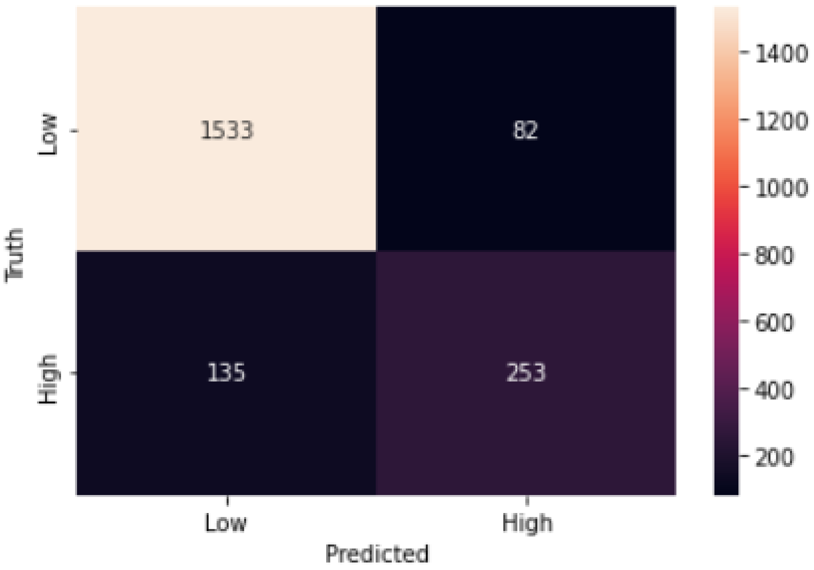


**Figure 6.1:** SVM-TFIDF in Binary Classification

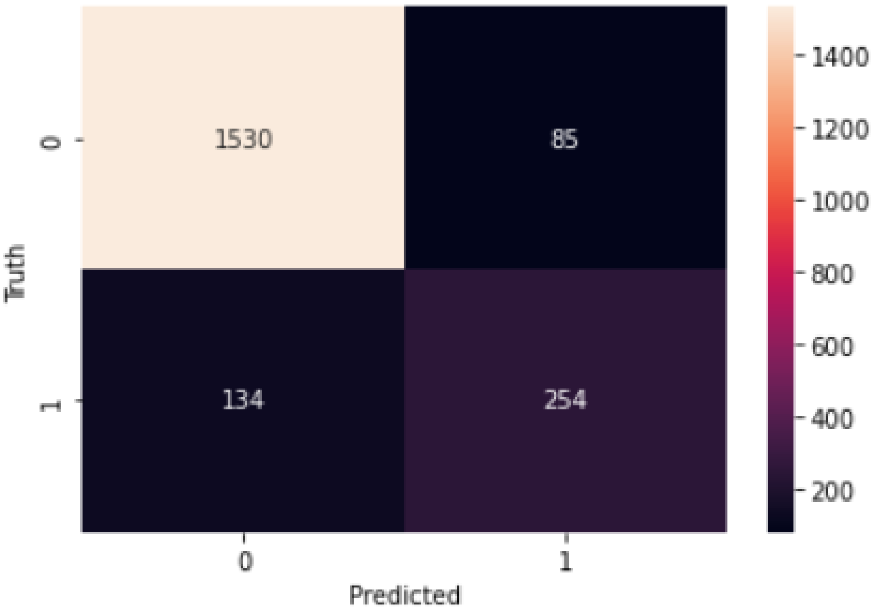


**Figure 6.2:** SGD-TFIDF in Binary Classification

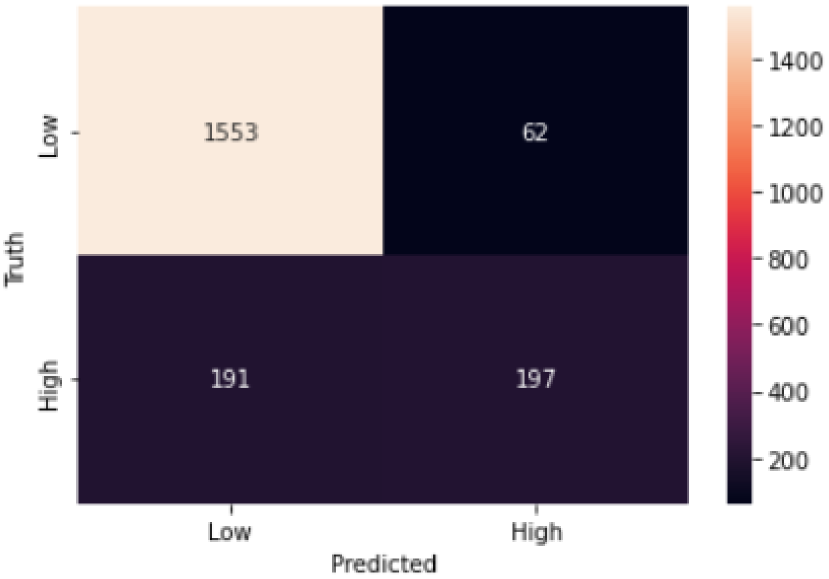


**Figure 6.3:** NB-TFIDF in Binary Classification

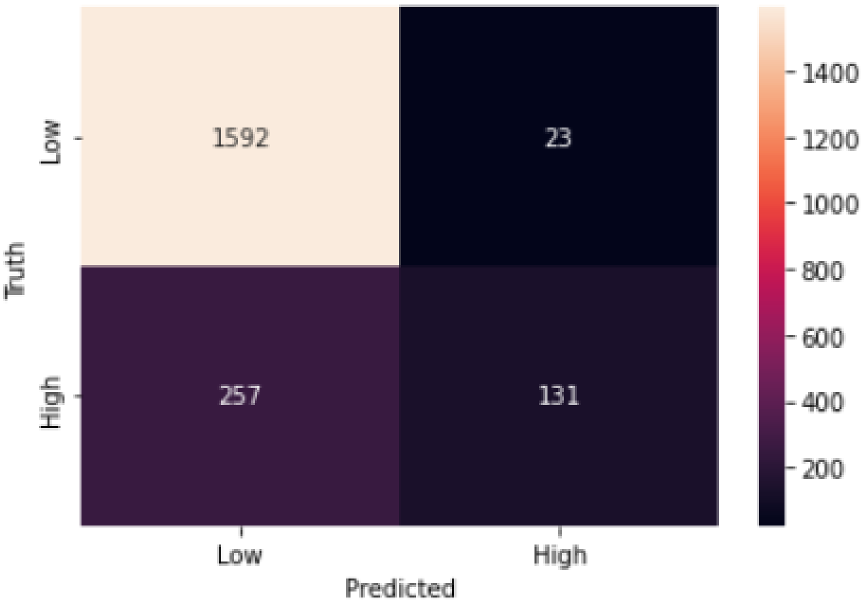


**Figure 6.4:** RF-TFIDF in Binary Classification

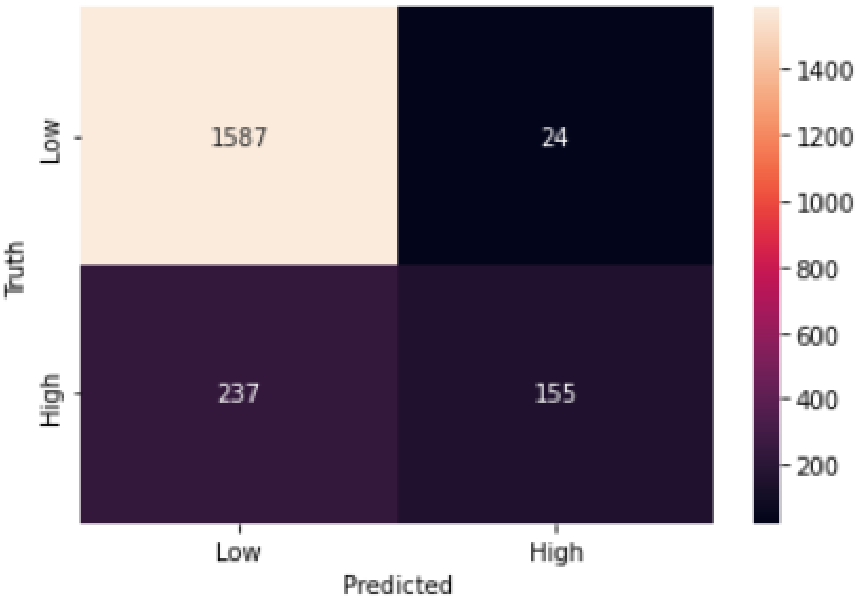


**Figure 6.5:** BERT in Binary Classification

### 6.2.2 Results for Multi-Class Classification

The results for multi-class classification are quite satisfactory as well with the highest accuracy being achieved is 86% using both linear SVM and SGD. However, precision varies for both these top-performing classifiers, 69% precision was achieved using SVM while 71% precision was achieved using SGD classifier for multi-class classification of data.

Similarly, like binary classification, NB is the third-best multi-class classification classifier. An accuracy of 85% and a precision of 83% were achieved using NB.

After NB, the next best model for multi-class classification is the pre-trained BERT model. An accuracy of 84% and a precision of 78% were achieved using BERT. The last model in terms of performance is RF with TF-IDF word embedding in which an accuracy of 83% and precision of 73% was achieved.

Below are some of the snippets of the performance of these models for multi-class classification visualized through the confusion matrix for different experiments that were conducted using varying word embeddings and classifier combinations.

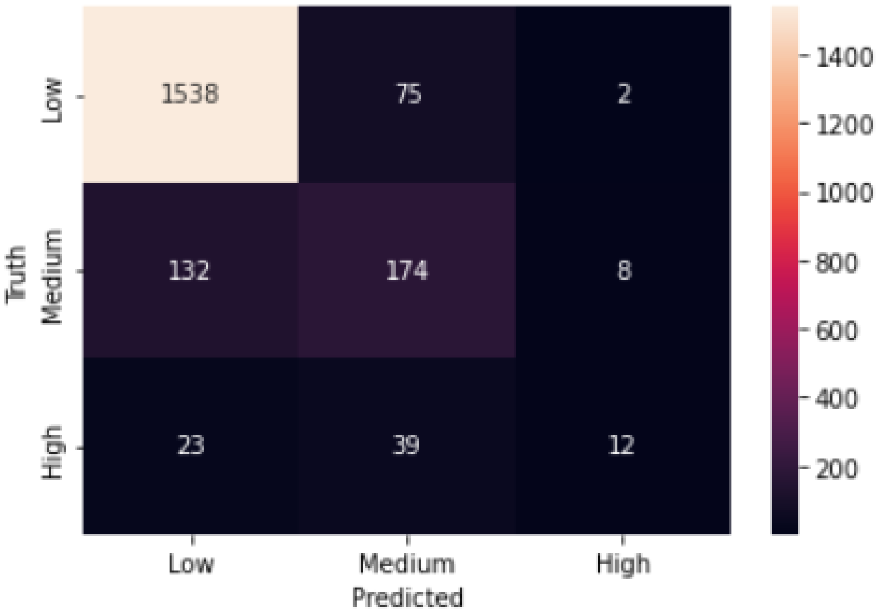


**Figure 6.6:** SVM-TFIDF in Multi-class Classification

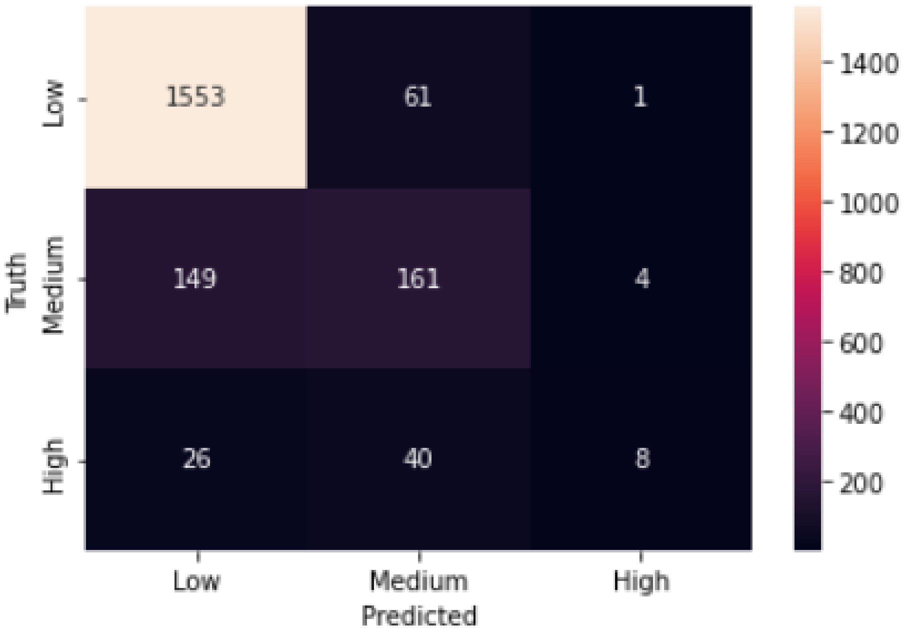


**Figure 6.7:** SGD-TFIDF in Multi-class Classification

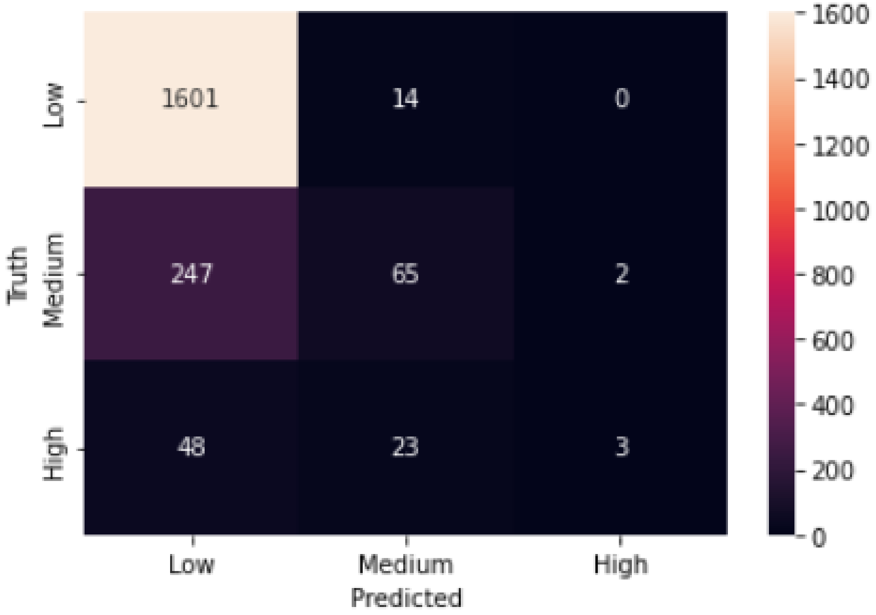


**Figure 6.8:** RF-TFIDF in Multi-class Classification

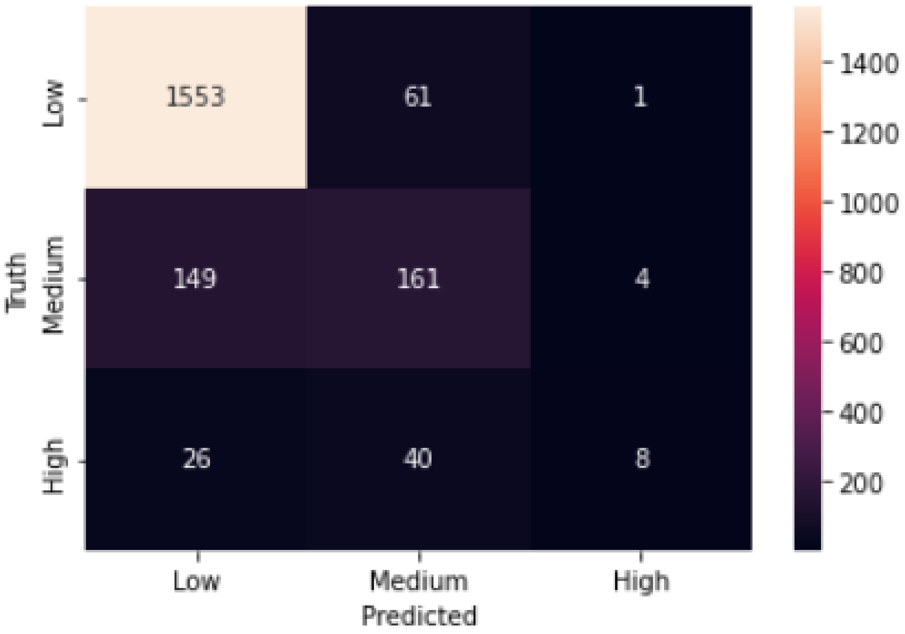


**Figure 6.9:** SGD-TFIDF in Multi-class Classification

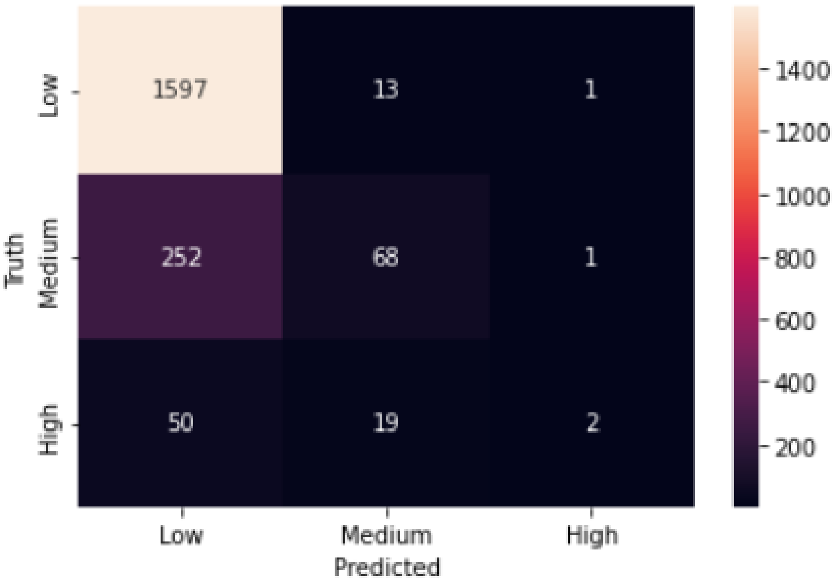
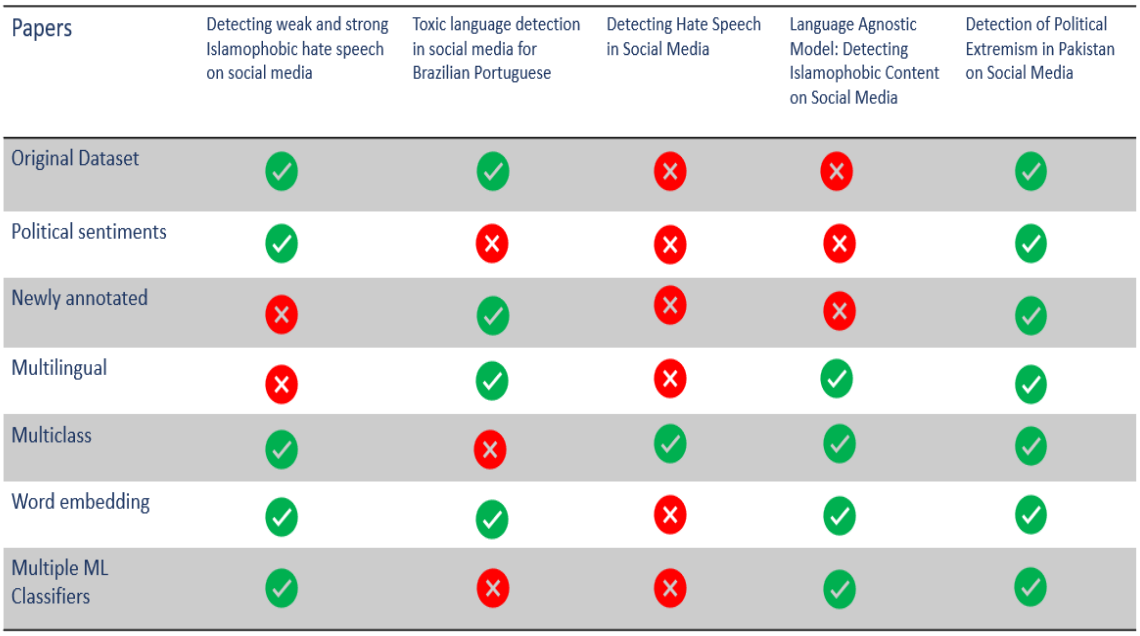


**Figure 6.10:** BERT in Multi-class Classification

## 6.3 Comparison With Existing Work

Recently, there has been substantial work on sentiment analysis. Many researchers have proposed different solutions for sentiment analysis using both binary classification and multi-class classification. When it comes to sentiment analysis, the selection of the data set, its annotations, and pre-processing hold critical importance for achieving expected results. Some of the research works that are closely related to our research topic are listed below. A comparison of their performance with our model is also highlighted in Figure 6.11.

Researchers have proposed different solutions for sentiment analysis. But most of their solutions focused on binary classification which do not contribute significantly to the practical solutions required for sentiment analysis because of the varied nature of sentiments. There are also some dynamic approaches to detect and classify sentiments in multi-class but none of these solutions detect extreme political sentiments. Moreover, most of the existing research works focus on English language data only which is not ideal since social media platforms have data in almost every language. Therefore, a robust solution is needed that detects and translates multi-lingual



| Papers | Detecting weak and strong Islamophobic hate speech on social media | Toxic language detection in social media for Brazilian Portuguese | Detecting Hate Speech in Social Media | Language Agnostic Model: Detecting Islamophobic Content on Social Media | Detection of Political Extremism in Pakistan on Social Media |
|---|---|---|---|---|---|
| Original Dataset | ✓ | ✓ | ✗ | ✗ | ✓ |
| Political sentiments | ✓ | ✗ | ✗ | ✗ | ✓ |
| Newly annotated | ✗ | ✓ | ✗ | ✗ | ✓ |
| Multilingual | ✗ | ✓ | ✗ | ✓ | ✓ |
| Multiclass | ✓ | ✗ | ✓ | ✓ | ✓ |
| Word embedding | ✓ | ✓ | ✗ | ✓ | ✓ |
| Multiple ML Classifiers | ✓ | ✗ | ✗ | ✓ | ✓ |

**Figure 6.11:** Comparison With Existing Work

data so that the feature extraction and model training process can be standardized.

The overall performance of this sentiment analysis model is quite satisfactory compared to the previous researchers in similar domains. The data set used for this research work is not only freshly acquired but also both multi-lingual and multi-class.

The highest-performing classifier for this extremism detecting model is SVM in both binary as well as multi-class classification. The highest accuracy of 89% is achieved using SVM in binary classification. While the lowest accuracy of 83% is achieved with the RF classifier.

Initially, it was expected that a pre-trained BERT model would outperform all other machine learning classifiers because of its impressive learning abilities and excessive pre-training but instead, SVM outperformed BERT by 3% in binary classification and by 2% in multi-class classification. The results from this research work contribute to the ongoing discussion [54] on this topic.

## 6.4 Summary

In this chapter, the performance of our proposed solution to detect extreme political sentiments is discussed in detail. Chronologically, firstly the evaluation metrics are highlighted, and then our final results are discussed in detail. Lastly, the performance of this proposed solution is compared with existing research works that are closely related to this topic.

# Chapter 7

# Conclusion and Future Work

## 7.1 Conclusion

The main contribution of this paper is to develop quantitative methods to detect extreme political sentiments on social media. In this model, a fairly simple and unique solution is presented to detect political extremism on social media. The algorithms that were used for noise cancellation and pre-processing of text data obtained from Twitter are also demonstrated. During that process, the input raw tweets are refined and processed to get more accurate data for training machine learning classifiers. Data pre-processing also helps reduce the size of the textual data set which is important to make a lightweight sentiment analysis model.

To perform core sentiment analysis, the NLTK toolkit was used which enabled the processing of natural language more efficiently. The word embedding process is initialized with the word tokenization process in which each word present in the Twitter data is tokenized with the help of different word embedding techniques. In the next step, this tokenized data is used to train different machine learning algorithms as well as a pre-trained BERT model. The data was split into 80:20 ratios for training and testing of this sentiment analysis model. In the final step, these newly trained classifiers were tested with the testing data. The results that are achieved from these machine learning classifiers and BERT (as shown in Table 6.1 and Table 6.2) indicate that this is a viable solution to detect extremism. While more work needs to be done in order to make better distinctions between different types of extremism, the results reported in this experiment are promising.

Moreover, the simplicity of this model makes it easy to use the method and findings from these experiments in detecting other forms of sentiments as well such as racism, misogyny, ethnic discrimination, etc.

## 7.2 Future Work

For future work on detecting political extremism, we want to expand the source of the political data. For that, we want to collect data from different mediums used by politicians for information sharing across the globe and detect their sentiments. Moreover, we plan to enhance the performance of the model by further increasing the data set size and using additional input features.

We also want to build an app or a browser extension of our model which can detect political extremism in real-time. Additionally, we want to experiment with additional word embedding models such as gloVe [55], PRADO [56], and more machine learning classifiers such as LSTM [57] to learn whether these classifiers can perform better than the one we have used in this thesis.

# Bibliography

[1] An overview of twitter users statistics, https://www.bankmycell.com/blog/how-many-users-does-twitter-have.

[2] Eman MG Younis. Sentiment analysis and text mining for social media microblogs using open source tools: an empirical study. *International Journal of Computer Applications*, 112(5), 2015.

[3] I Hemalatha, GP Saradhi Varma, and A Govardhan. Preprocessing the informal text for efficient sentiment analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 1(2):58–61, 2012.

[4] Social media and political extremism in the united states, https://onlinewilder.vcu.edu/blog/political-extremism/.

[5] Political extremism and pakistan, https://www.pakistantoday.com.pk/2023/01/08/political-extremism-and-pakistan/.

[6] Dhiraj Murthy. Evaluating platform accountability: terrorist content on youtube. *American behavioral scientist*, 65(6):800–824, 2021.

[7] Kevin L Nadal, Katie E Griffin, Sahran Hamit, Jayleen Leon, Michael Tobio, and David P Rivera. Subtle and overt forms of islamophobia: Microaggressions toward muslim americans. 2012.

[8] Dictionary by merriam-webster: America's most trusted online dictionary, https://www.merriam-webster.com/.

[9] Andrej Sotlar. Some problems with a definition and perception of extremism within a society. *Policing in central and Eastern Europe: Dilemmas of contemporary criminal justice*, pages 703–707, 2004.

[10] Jin Woo Kim, Andrew Guess, Brendan Nyhan, and Jason Reifler. The distorting prism of social media: How self-selection and exposure to incivility fuel online comment toxicity. *Journal of Communication*, 71(6):922–946, 2021.

[11] Tim Phillips and Philip Smith. Everyday incivility: Towards a benchmark. *The Sociological Review*, 51(1):85–108, 2003.

[12] Reddit (2021), reddit policy for the privacy and security of is available at, https://www.reddithelp.com/hc/en-us/categories/360003246511-privacy-security.

[13] Shakeel Ahmad, Muhammad Zubair Asghar, Fahad M Alotaibi, and Irfanullah Awan. Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Human-centric Computing and Information Sciences*, 9:1–23, 2019.

[14] Mayur Gaikwad, Swati Ahirrao, Shraddha Phansalkar, Ketan Kotecha, Shalli Rani, et al. Multi-ideology, multiclass online extremism dataset, and its evaluation using machine learning. *Computational Intelligence and Neuroscience*, 2023, 2023.

[15] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

[16] Wisam A Qader, Musa M Ameen, and Bilal I Ahmed. An overview of bag of words; importance, implementation, applications, and challenges. In *2019 international engineering conference (IEC)*, pages 200–204. IEEE, 2019.

[17] Shahzad Qaiser and Ramsha Ali. Text mining: use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1):25–29, 2018.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[19] Bert 101 - state of the art nlp model explained, https://huggingface.co/blog/bert-101: :text=what%20is%20bert%3f,model%20for%20natural%20language%20processing.

[20] Bert multi-lingual base model (cased), https://huggingface.co/bert-base-multilingual-cased.

[21] Glove: Global vectors for word representation, https://nlp.stanford.edu/projects/glove/.

[22] Allennlp - elmo — allen institute for ai, https://allenai.org/allennlp/software/elmo.

[23] A guide on word embeddings in nlp is available at, https://www.turing.com/kb/guide-on-word-embeddings-in-nlp.

[24] Bert explained: State of the art language model for nlp, https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

[25] Bertie Vidgen and Taha Yasseri. Detecting weak and strong islamophobic hate speech on social media. *Journal of Information Technology & Politics*, 17(1):66–78, 2020.

[26] Heena Khan and Joshua L Phillips. Language agnostic model: Detecting islamophobic content on social media. In *Proceedings of the 2021 ACM Southeast conference*, pages 229–233, 2021.

[27] Crowd sources online database - crowdflower, https://hatebase.org/.

[28] Shervin Malmasi and Marcos Zampieri. Detecting hate speech in social media. *arXiv preprint arXiv:1712.06427*, 2017.

[29] Online data sets - crowdflower, https://data.world/crowdflower/.

[30] Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression, and cyberbullying (TRAC-2018)*, pages 1–11, 2018.

[31] Joao A Leite, Diego F Silva, Kalina Bontcheva, and Carolina Scarton. Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. *arXiv preprint arXiv:2010.04543*, 2020.

[32] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[33] Academic research access - advance your research objectives with public data on nearly any topic, https://developer.twitter.com/en/products/twitter-api/academic-research.

[34] Tweepy - an easy-to-use python library for accessing the twitter api, https://www.tweepy.org/.

[35] pandas - python data analysis library, https://pypi.org/project/pandas/.

[36] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. Word cloud explorer: Text analytics based on word clouds. In *2014 47th Hawaii international conference on system sciences*, pages 1833–1842. IEEE, 2014.

[37] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[38] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[39] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[40] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[43] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[44] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[45] Open source nlp models for sentiment analysis, https://pub.towardsai.net/16-open-source-nlp-models-for-sentiment-analysis-one-rises-on-top-b5867e247116.

[46] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *Osdi*, volume 16, pages 265–283. Savannah, GA, USA, 2016.

[47] Sentiment analysis with tensorflow hub, https://medium.com/codex/sentiment-analysis-with-tensorflow-hub-678c30ac79a2.

[48] Jupyter notebook for open-source programming, https://jupyter.org/.

[49] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*, 2019.

[50] Lubor Ladicky and Philip Torr. Locally linear support vector machines. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 985–992, 2011.

[51] Oliver Kramer and Oliver Kramer. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53, 2016.

[52] Shuo Xu, Yan Li, and Zheng Wang. Bayesian multinomial naïve bayes classifier to text classification. In *Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017 11*, pages 347–352. Springer, 2017.

[53] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. arxiv 2018. *arXiv preprint arXiv:1811.03378*, 2018.

[54] Yasmen Wahba, Nazim Madhavji, and John Steinbacher. A comparison of svm against pre-trained language models (plms) for text classification tasks. In *Machine Learning, Optimization, and Data Science: 8th International Workshop, LOD 2022, Certosa di Pontignano, Italy, September 19–22, 2022, Revised Selected Papers, Part II*, pages 304–313. Springer, 2023.

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

[56] Prabhu Kaliamoorthi, Sujith Ravi, and Zornitsa Kozareva. Prado: Projection attention networks for document classification on-device. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5012–5021, 2019.

[57] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

## Certificate for Plagiarism

It is certified that PhD/M.Phil/MS Thesis Titled "Detection of Political Extremism in Pakistan on Social Media" by HAFIZA RABAIL MUSHTAQ has been examined by us. We undertake the follows:

a. Thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.

c. There is no fabrication of data or results which have been compiled/analyzed.

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC plagiarism Policy and instructions issued from time to time.

**Name & Signature of Supervisor**

Dr. Sana Qadir

Signature : Sana Qadir