1.5em

AUTOMATIC DETECTION OF OFFENSIVE LANGUAGE FOR ROMAN PASHTU

By

Anas Ali Khan

A thesis submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology,
Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in
Electrical (Telecommunication) Engineering

February 2022

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Anas Ali Khan** Registration No. **00000276676**, of Military College of Signals has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have been also incorporated in the said thesis.

Signature: _____

Supervisor: Asst Prof. Dr. Shibli Nisar

Date: _____

Signature (HoD): _____

Date: _____

Signature (Dean): _____

Date: _____

ABSTRACT

Now adays, cyberbullying on social media platforms is at its peak. It's a vital challenge for researchers these days. And hence a tally of research work is done to address this issue in a variety of languages around the Globe. Social media dices are heavily used by people to express their views in their native languages. Besides positive views, people often use abusive or offensive language to express their anger or frustration. Resource rich languages have offensive language detection systems to automatically monitor and block offensive content, however, they are very rare for low resourced languages. This is because of the non-availability of datasets for local languages. This work proposes a model which automatically detects offensive language for a very low resource language i.e., Pashto. The roman Pashto dataset is created by picking 60 thousand comments from different social media and labeling them manually. The proposed model is trained and tested using three different feature extraction approaches i.e., bag-of-words (BoW), term frequency-inverse document frequency (TF-IDF), and sequence integer encoding. Four traditional classifiers and a deep sequence model are used to train on this task. Experimental result shows that random forest classifier works best and give 94.07 The corpus created in this work is made available for the researcher working in this domain.

Keywords — Natural Language Processing, Text Mining, Automation, Deep Learning

DEDICATION

This thesis is dedicated to

MY FAMILY AND TEACHERS

for their love, endless support and encouragement

ACKNOWLEDGEMENTS

I am grateful to Allah, the Almighty, for His mercy and benevolence who has bestowed me with the strength and the passion to complete this thesis. Without his consent I could not have indulged myself in this task.

I am also thankful to my supervisor especially and committee members who have always guided me with their keen and useful counseling in achieving my research objectives.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Motivation

The natality of social media has directly influenced the methods and intent of mass communication [3]. It was initially governed by ethical and social norms before the nativity of social media. Mass communication was initially used for awareness and cultivation of knowledge, effectively. But with the parturition of social media platforms, the intent of mass communication was deeply influenced and now adays social media dices are heavily used by people to express their views in their native languages. People mostly feel safe to use their native language for communication and consider it as a natural flare to use on social media dices for expressing their views. Some people can only speak their native language to communicate and hence they excessively use it for communication on social media platforms. Society has got different pilers and one can classify it broadly in two types, positive and negative. Besides positive views from positive pilers of the society, we can notice abusive or offensive language used on these platforms from the negative pilers to express their anger or frustration about anything on social media. It may include various content like verbal and written content. And people mostly prefer to communicate in written comments using their native languages. So, negative comments from negative users and social media platforms imparts the parturition of cyberbullying. Due to the popularity of social media dices and the increased rate of their nativity such as TikTok, Snack Videos, etc., has a vital role in increasing cyberbullying exponentially in the society. It's pretty much consequential issue for the individuals in the society and needed to be addressed on time, as we can't afford the increasing ratio any further. Therefore, we opted to pursue the initiative with the traditional local language Pashto.

1.2 Contribution

Resource rich languages have offensive language detection systems to automatically monitor and block offensive content, however; they are very rare for low resourced languages. This is because of the non-availability of datasets for local languages. Mostly, resource rich languages like English, French, German and Arabic etc. have datasets easily available on Web and hence, mostly researchers have focused to address the afore mentioned consequential issue of cyberbullying for such languages on priority basis. Here in this study, we'll discuss the work done for such resource rich languages as a

model and will focus on the resource power language, Pashto. So far researchers round the Globe has neglected resource power languages like Pashto, Urdu, Punjabi and many other local languages with power recourses, in this regard; due to which the ratio of cyberbullying has increased drastically in past few years in such languages. This work possesses a model which automatically detects offensive language for a very low resource language i.e., Pashto. The roman Pashto dataset is created by picking 60 thousand comments from different social media platforms and labeling them manually.

1.3 Goals

This work is aimed to highlight the peripheral challenge of cyberbullying in local languages due to the unawareness of the speaking community and almost negligible contributions to counter the issue in such languages. To achieve the afore mentioned milestone, we opted to work for a very popular local language Pashto and hence, proposed a model which is trained and tested using three different feature extraction approaches i.e., bag-of-words (BoW), term frequency-inverse document frequency (TF-IDF), and sequence integer encoding. Four traditional classifiers and a deep sequence model are used to train on this task. Experimental results are very much impressive. The results shows that random forest classifier works best and give 94.07% accuracy on a combination of unigrams, bigrams, and trigrams. The same classifier gives maximum accuracy of 93.90% with TF-IDF. However, the overall highest accuracy of 97.21% is achieved using bidirectional long short-term memory (BLSTM). The corpus created in this work is made available for the researcher working in this domain.

1.4 Observations

As mentioned above, the social media has played a vital role to directly influence the methods and intent of mass communication [4]. Times ago, mass communication was initially the best suite for the awareness and cultivation of knowledge. Initially, it was ruled by ethical and social norms. Nowadays, social media avowed individuals to connect and communicate their perception about anything via platforms like Twitter, Facebook, Instagram, Snapchat, YouTube, and TikTok, etc. [5]. The latest research about social media influencers have revealed that people have minimal tolerance in their emotions and conduct, which imparts aggression to their behavior and content [6]. As a result, people use language that antagonizes the feelings of others. There is no Face-to-Face contact among users, which empowers individuals to share their opinion without any dread. Here comes the parturition of cyberbullying, which is a big challenge for researchers these days. Despite the tenet for content published by various social media platforms, it's quite difficult to encounter the violations, manually [7, 8]. The reason is the huge tally of data on social media platforms. The privilege to users of social media daises for expressing their feelings and opinions in native languages makes it more

difficult to detect the violations. Therefore, hate speech and offensive language detection on social media have become an active area under research nowadays [9].

1.5 Social Media as a key Influencer in Cyberbullying

Social media daises are interactive technologies that provide a central point of communication for people around the world and hence, is a key influencer in the parturition of cyberbullying in community round the Globe. Individuals of various geographic locations, religions, skin colors, and cultures often troll each other using invasive language [10]. People mostly favor and feel contented to use their native language to write their judgment, response, or remarks about online products, videos, and articles [11]. Comments with belligerent language arguments should not be perceptible to other users because it grounds cyberbullying [12]. And there is no mechanism yet implemented on social media platforms to encounter such comments in local languages from the negative users on social media. Therefore, global community is facing a big challenge regarding measures to be taken to control the elevating graph of cyberbullying in local languages.

1.6 National Interests and Benefits

Let us take Pakistani community as an example in the effectives of the afore mentioned social virus. On 28th of August, 2021, The News claimed that an increase of 83% is recorded in cybercrimes in Pakistan in the last three years. The statistics of social media subscribers in Pakistan shows an increase of 24% and having a total stack of around 46 million as per a report published in February 2021. Therefore, Pashto speaking community being the third largest in Pakistan must have contributed considerably in the above facts and figures regarding social media in Pakistan. The mentioned 83% increase is surely backed by the parturition of social virus in shape of cyberbullying on social media platforms. Pashto speakers are pretty much popular on TikTok and other social media platforms these days.

1.7 Research Objective

The above discussion on the facts and figures of our country in social media aspects is imperatively demanding an effective contribution in NLP (Natural Language Processing) for local languages in Pakistan.

Pashto language is one of the most popular and widely spoken local language of the people in K.P.K region of our country, Pakistan. According to a report of UNESCO, there are about 25 million Pashto speakers in Pakistan. And about 60 million speakers around the world. Therefore, the Pashto speaking community is also facing the consequences of uncontrolled and unsupervised parturitions

of social media dices. The community is facing a big challenge these days in cyber grounds. Hence, it is imperative to propose an instinctive system to spot, rest or veto belligerent language before it is published online. In this work, we proposed different models to encounter the afore mentioned issue for roman Pashto comments on social media platforms. This work will provide grounds to address the uncontrolled parturition of cyberbullying in Pashto speaking community.

1.8 Thesis Structure

This work comprises of mainly five chapters. The very first chapter is about the introduction to the topic. It discusses the motivation, contributions, goals, observations, National interests and benefits. The chapter is summarized with discussing the thesis structure. The second chapter presents the literature review. The third chapter is about the methodology we have followed to achieve the milestone, including data collection process, flow diagrams and discussion about the data classification and purification techniques. Fourth chapter includes the deep insight in the proposed models with a discussion on the outcomes of each model along with the comparison of their outcomes. It is summarized through tables and graphs of comparison of the results of each model. The Fifth and the very last chapter is about the conclusion to this marvelous achievement with discussion of future contributions in this regard.

Chapter 2

LITERATURE REVIEW

2.1 Related Work

Researchers so far felt contented to address the highlighted issue due to available resources of the resourceful languages and therefore, have focused on the resourceful languages form the last decade and a handsome amount of work is done so far in this regard. Until now most of the researchers have encountered the aforementioned challenge for several resource rich languages like English, Arabic, German and Indonesian, etc [13–15]. In the past few years, researchers extensively used machine learning techniques for Natural Language Processing (NLP), especially for belligerent language and abhor comments from social media users [16–18]. Reference [19], used N-gram features extraction technique and machine learning models to detect the belligerent language comments from YouTube in Arabic. Similarly, same n-gram approach features and machine learning models to perceive belligerent text from Indonesian social media [20]. Schneider et al. [21], encountered the emanate cyberbullying issue for the German language by using the technique of convolutional networks to detect the antagonistic comments from Twitter. In 2019, G. I. Sigurbergsson and L. Derczynski used LSTM and Logistic regression techniques to utter the challenge of unethical and offensive comments detection for Danish and English languages [18].

All of the above work done has a great significance in the field of NLP. This work has imparted grounds to peruse the same for local languages and so did by us for a very popular and historical local language Pashto. This work is focused on the discussed challenge of cyberbullying for a resource poor language Pashto. Pashto speaking community is the 3rd largest in Pakistan with more than 15% stack and more than 25 million speakers [22]. YouTube is a highly used video website that contains millions of users around the world [23]. People watch content provided by YouTube and mostly share their opinion in comments. Pashto speaking community is also seemed quite active on YouTube. Also, traces of cyberbullying can be sensed from the contents and comments of the Pashto speakers on the platform of YouTube. YouTube, is the maximum trafficked website after Google. YouTube contains Trillions of hours of videos and about 2 billion viewers. Pashto speakers fairly contributes to the viewer's stack. Similarly, other social media platforms like Facebook also provide a hot platform for people around the globe to communicate and share their content [24]. Nowadays,

TikTok is playing a vital role in the elevating graph of cyberbullying, especially, in Pakistan and other countries like India [5]. Pashto speakers are also active on the Facebook and TikTok dices. According to a survey in January 2021, there are about 61.34 million internet users and about 46 million social media subscribers in Pakistan [6]. These numbers increased with rapid rate in past few years. Since Pashto speaking community is the 3rd largest in Pakistan, therefore, they contribute fairly to the numbers of social media subscribers. Cyberbullying in Pashto speaking community is increasing day by day. This is a social disaster for Pashto speaking community and need to be addressed. Pashto is a resource poor language and has multiple dialects [25, 26]. But people mostly use English alphabets rather than Pashto alphabets to express their views. Similarly, in Afghanistan, there are about 19 million Pashto speakers and Pashto is one of the most communicated and official language [27]. Cyberbullying amongst the people of Afghanistan and Pakistan is also at its peak. People of Afghanistan also use mostly roman Pashto comments to express their views. Therefore, considering the utmost requirement for the mitigation of the social disaster of the Pashto speaking community via an increasing ratio of cyberbullying through social media. We took responsibility to initiate the war against the social virus in the form of cyberbullying in Pashto speaking community. We tried our best to formulate an instinctive model to control the virus of cyberbullying in roman Pashto on social media platforms. We achieved very handsome results for each model we have suggested in this study and efficiently encountered the belligerent comments in our dataset.

Chapter 3

METHODOLOGY

The complete proposed model is discussed in this section. Data set collection for Roman Pashto was the first and challenging task for offensive text classification. After data set collection it was preprocessed to convert it in a structured form. All the HTML tags, links, URLs, unnecessary characters, and digits were removed from the data. During preprocessing, some stop words e.g., “is”, “he”, “we” etc. were also removed. Such words are not important because they do not contribute to the classification of offensive text. After preprocessing of the data set, features were extracted using BoW, TF-IDF, and sequential model. Four different machine learning classifiers were used on 700 most frequent features for offensive text classification. A deep learning model was also trained and tested for classification. The complete pipeline of the proposed model is shown in Figure 3.1.

3.1 Dataset Collection and Preprocessing

A key challenge in this research work was the collection of the Roman Pashto dataset because there was no such data set available online. A dataset of more than 60,000 documents was duly compiled by taking the comments in Roman Pashto from social media platforms i.e., YouTube, TikTok, and Facebook etc. Different types of comments were available on these digital platforms under the video section or posts. After the collection of roman Pashto comments, they were manually annotated in terms of offensive and non-offensive documents. We labeled ‘TRUE’ for non- offensive and ‘FALSE’

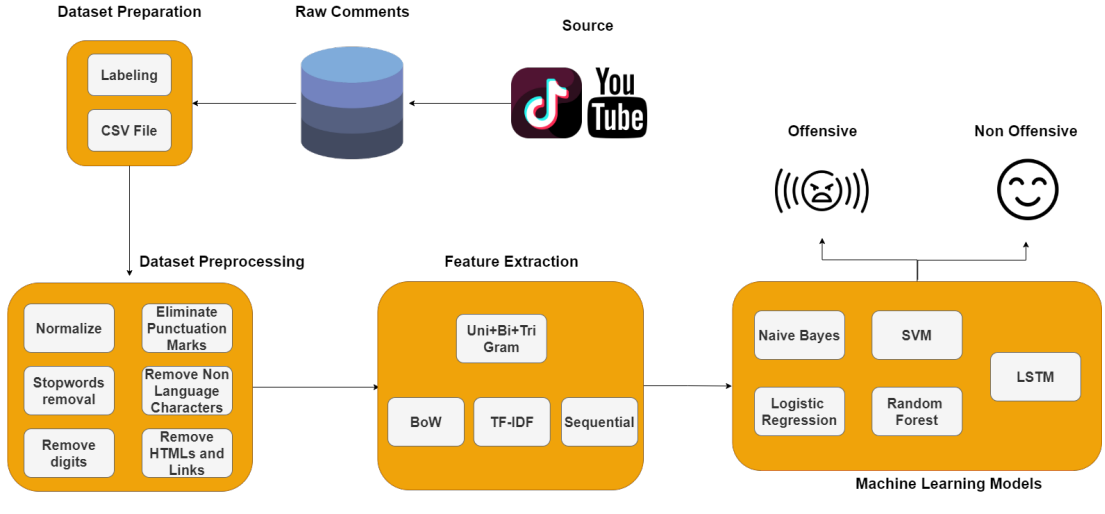

Figure 3.1: Complete pipeline of proposed model

for offensive document. Therefore, the results generated in this work are 100% correct and the accuracy achieved is duly verified, as the annotation was done by reading each document (human approach and understanding). The labeling approach as mentioned was totally based on personal understanding and was done as per the content to which each document was posted. Therefore, the labeling of each document as offensive or not in itself was a challenging task for more than 60,000 comments in the corpus. It is the result of a critical contextual analysis of each document in the corpus.

Dataset was the key to proceed with the solution model to the subject. Because there was no such dataset yet available digitally for roman Pashto. We collected roman Pashto comments from more than 800 videos of YouTube, Facebook and TikTok. We manually collected the data, as there is no such tool available so far in our knowledge to extract commented data from any social media platform. After the collection of more than 60,000 commented data, we had a dataset in raw form consisting of unwanted punctuations, html codes/commands and HTTP links. We applied preprocessing techniques to clear the unwanted content from the dataset. After the successful cleaning of the dataset, we were left with a dataset of about 60000 comments ready to be labeled. We then manually labeled the dataset as 'TRUE' for non-offensive and 'False' for offensive comments. The challenge of labeling the dataset was fulfilled, each comment was analyzed critically to declare it as TRUE or FALSE. Our analysis is so deep and precise that we also considered the contexts of each comment. We had to watch the video content with attention to understand the scenario and theme of the content being presented in that video. Afterwards, we followed the comments on each video. We thoroughly examined each comment of the users and their conversations with each other via comments, by doing such keen and deep analysis of each comment we were able to label the comments properly. Moreover, the labels assigned to each comment were so accurate that it may infer the hidden positivity inside an apparently negative comment. We are so confident about the labeling, because it was done by human effort and understanding, based on keen observations.

Besides the cleaning and labeling of the data, we also addressed the repetitions of same comments. It was done intentionally, so that we could have as unique dataset as we could to enhance the effectiveness and accuracy of our models. Another intention to have as much unique dataset as we can was to prepare a huge dataset with maximum content. This also helped us to achieve the problem of various dialects we have for Pashto language, as people use mostly roman Pashto and communicate in their own dialects. We then applied n-gram technique for feature extraction from the cleaned and labeled dataset we had by then. We took 700 most frequent and important features from our dataset, using 'Bag of Words' And 'TF-IDF' approach. We also applied the sequential data classification

Table 3.1: Examples of designing n-grams from roman Pashto sentence

N-grams	Roman Pashto
Sentence	Aga yaw dala kas dy
Unigram	‘aga’, ‘yaw’, ‘dala’, ‘kas’, ‘dy’
Bigram	‘aga yaw’, ‘yaw dala’, ‘dala kas’, ‘kas dy’
Trigram	‘aga yaw dala’, ‘yaw dala kas’, ‘dala kas dy’

technique and generated the files with same number and features classification. We recorded this data in Microsoft Excel with ‘.csv’ extension. We applied various machine learning classifiers to build models for all the generated files. We recorded the model building time and the percentage of correctly classified instances for each model and did comparison. The comparison tables are mentioned in up-coming section.

3.2 Character N-gram and Word N-gram

Character N-gram and Word N-gram technique were used for sentence tokenization in a sequence of words (Word N-Gram) or tokenizing a word in a sequence of characters (Character N-Gram). It also split the speech into phonemes. Sequenced words were used as features in natural language processing (NLP) while phonemes are used in Speech processing. It assigned probability values to the sequenced words or characters which were used for classification. This model can be used for next item prediction in a sequence and auto completion of sentences. Tokens probabilities were used by classifiers for text or speech classification. N-Gram technique is further extended into the following models. An n-gram model is used to predict the next item in a sequence. It’s widely used model in NLP and one of its uses includes auto completion of sentences. As the title suggests, there are two types of n-gram models and they are differentiated and explained in tables below.

Features played a vital role in classification of textual data through classifiers. Features could be extracted from the data under observation and analysis, through various feature extraction techniques. N-gram technique for features extraction was the most beneficent technique for us that consists of abutting succession of n words or characters. N-gram technique helps to assign probabilities to the characters and words in natural language processing. The assigned probabilities are then used by the classifiers to classify the text.

1. **Uni-gram:** feature with single character or word is known as uni-gram (n=1).
2. **Bi-gram:** feature with two contiguous characters or words is known as bi-gram (n=2).
3. **Tri-gram:** feature with three contiguous characters or words is known as tri-gram (n=3).

4. **Uni+Bi-gram:** Combination of uni+bi gram.

5. **Uni+Tri-gram:** features are combined in uni+tri pattern

6. **Uni+Bi+Tri-gram:** features are combined in uni+bi+tri pattern

The number of n-grams tokens from a sentence can be calculated as follow:

$$\text{N-Grams} = Y - (n - 1) \tag{3.1}$$

Where 'Y' is the total number of words (or characters) in a sentence and n is the number of contiguous words (or character). Table 1 shows a sentence with 6 words. It has 6 uni-grams, 5 bi-grams, and 4 tri-grams. In this work, uni + bi + tri-gram model is used to tokenize all the documents in the dataset. After the application of the n-grams technique on the data set, the resulting tokens act as features for data classification. Methods e.g., Bag of Words or TF-IDF are used for assigning numerical values to extracted features. Where 'Y' is the total number of words (or characters) in a sentence and 'n' is the number of contiguous words (or character). Table 1 shows a sentence with 6 words. It has 6 uni-grams, 5 bi-grams, and 4 tri-grams. In this work, uni + bi + tri-gram model is used to tokenize all the documents in the dataset. After the application of the n-grams technique on the dataset, the resulting tokens act as features for data classification. Methods e.g., Bag of Words or TF-IDF are used for assigning numerical values to extracted features. And hence preparing the dataset for model building through classifiers.

3.3 Feature Extraction and Selection

Using n-gram technique, all the documents in the corpus are tokenized which are used as features for text classification. These features are selected using three frequently used models i.e., BoW, TF-IDF and sequential model.

3.3.1 Bag of Words

To attain the information about the frequencies of each feature in the dataset, Bag of Words approach is the most common technique used to acquire the frequencies for the most important features of the dataset. Therefore, we also used the same technique to extract the information about the features in our dataset. According to this approach, the structure of data is lost as it only retains the frequency related information about the dataset. So, considering the fact as a flare and as per the desired outcomes of our dataset, it proved very much helpful for us. As we were not interested in retaining the structure of the document rather frequencies to produce an efficient classification model. One may

ask that how could we retain the position related information to have the correct structure of each document? But the answer to this question is quite simple that we don't need to keep this information. This approach could be understood more clearly in a way that for instance we put different objects in a bag, so we exactly know what we have inside the bag but we can't retain their position inside the bag. This is done intentionally as we need frequency related information for all the features of our dataset. And also, if we try to retain the position information as well, then we would be having longer chains of data and there will be uniqueness in dataset so the frequency related information would be lost. Therefore, we considered each labeled comment as single document and classified each document into maximum possible features. By doing so we got frequencies of each feature of the very first document in all other documents and so on. We stored this information in excel with '.csv' extension. After retaining all the possible information about each document's features of the whole dataset, we applied the afore mentioned techniques of frequency-based classification and TF-IDF based classification.

3.3.2 TF-IDF

The frequency-based representation achieved above is converted to TF-IDF representation. And this is the most helpful representation of data to achieve better results. In this type of representation, we are actually combining the two parts of representations. TF stands for 'Term Frequency' and IDF stands for 'Inverse Document Frequency'.

The TF of a feature can be calculated by calculating the ratio of the total occurrence of any feature in the document to the size of the document. By doing so, we just normalize that frequency value of that feature (word). Because if a specific document is longer enough then all the words have higher frequency in that document, doesn't mean that those words (features) are important rather it means that due to the higher length of the document, that feature has got good sufficient value.

Therefore, to reduce that difference and to treat longer and shorter documents equally we normalize the frequencies. The word with higher frequency will have higher TF value. This TF value is then multiplied by the IDF value of the same word. And it's calculated by taking log of the total number of documents (comments) we have in our dataset plus 1 and dividing it by the number of documents that includes that particular word for which we are going to calculate the TF-IDF value. The mathematical representation of TF-IDF will be as under;

$$TF = \frac{\text{Count of } t \text{ in } d}{\text{Num of words in } d} \tag{3.2}$$

$$TF = \frac{\text{Count of } d(t)}{N} \tag{3.3}$$

$$IDF(t) = Log\left[\frac{\text{N}}{DF+1}\right] \tag{3.4}$$

$$TF - IDF(t) = TF * IDF \tag{3.5}$$

The greater number of documents having that word will reduce the IDF value and hence that world would have lower significance. This is important to reduce the impact of frequently occurring words in most of the documents and are known as stop words. Till this point we have removed almost all of the stop words, but there can be some domain specific stop words that are still left, but their impact is minimized. So, the frequently occurring words and the stop words are normalized and removed via TF-IDF approach. We are then left with the words that are significant and having higher TF and higher IDF values. There are also some words which are rarely found in the dataset and occurs infrequently, are known as Noise words. Such type of words has also very less or almost no impact on the results. Therefore, TF-IDF is the most frequently used techniques to normalize data.

3.3.3 Sequential Model

Textual data is considered as time-series data such as weather, or financial data. N-grams technique tokenizes the text which is converted in vectors using BoW or TF-IDF model. BoW deals with the frequency of tokens in the document only while TF-IDF deals with words as well as document frequency and assign less weightage to non-significant words. Both these models are helpful for further classification using machine learning algorithms, but they do not keep the sequence of words in the corpus or the order of words occurring in the document. Keeping the sequence or order record for the words in documents is very important while classification. Sequence information helps in the prediction of new words, classification of documents with much better, and improved classification accuracy.

Word Indexed Dictionary

Text cannot be given as direct input to the neural networks because neural networks only accept numerical data. As the text is the sequence of words, if each word has an integer representation, it can be converted into a sequence of numbers. After preprocessing, all documents in the corpus are tokenized in vector forms which are then converted into word indexed dictionary. This dictionary keeps the sequence information of each word.

Input Sequence Padding

Neural networks require the input of the same size and shape for both training and testing data. In the dataset, all the documents are not of the same length. Once all the training and testing data is converted into vectors, every vector is of variable length. We need to have the training and testing sequences of the same size and shape, so we need to pad the input sequences to the maximum length. There are two types of padding: pre-padding and post-padding. In pre padding, all the input sequences which are shorter than the maximum length sequence, are padded with zeros in the beginning. In the case of post padding, the sequences are padded with zeros in the end.

Words Embedding

To present the features of a word, words embedding is used. Word embedding is a form of word representation that allows words that are used in similar ways to have similar representation. In embedding, each word is represented as real valued vector in a predefined vector space. A word's position within the vector space is learned from the text and is based on the words surrounding the word when used in the text. Each word is mapped to a real valued vector of high dimension.

Chapter 4

MACHINE LEARNING CLASSIFIERS

4.1 Naïve Bayes Classifier

As the name depicts, Naïve Bayes classifier operates on the Bayes Theorem with a speculation of uniqueness among predictors. In general, the classifier speculates that the occurrence of a particular feature in a class is not related to any other feature of the same class. For example, a fruit may be considered as banana, if its length is about 6 inches, curved a little bit and yellow in color. Therefore, even if these properties of the banana fruit may depend on each other or may relate to some other properties of the same fruit (Banana), each one contributes independently to its probability and that is why it's known as 'Naïve'.

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)} \tag{4.1}$$

where P(c| x) : Posterior probability of class C w.r.t predictor X

$P(x \mid c)$: Likelihood probability of predictor w.r.t class

$P(c)$: Prior probability of class

$P(x)$: Prior probability of predictor

Using the above formula of Bayes Theorem, the classifier calculates the probabilities of features 'x' in a class 'c'. Building a model based on Naïve Bayes Classifier is quite easy and helpful in natural language processing (NLP), where we have large datasets. It provides simplicity along with highly sophisticated classification methods. Besides some benefits like easy and fast to predict the class of unknown dataset along with multiclass prediction capability with assumption of independence, Naïve Bayes Classifier has excellent performance and requires less training data. Also, it performs well for categoric input variables. But occurrence of categoric variables in training dataset is a must, otherwise it assigns zero frequency to such variables. To avoid zero frequencies, we apply smoothing techniques and usually it's been done through Laplace Estimation. It is also known as bad estimator because assumption of independent predictors is quite impossible in real life and it's done by Naïve Bayes while calculating probabilities.

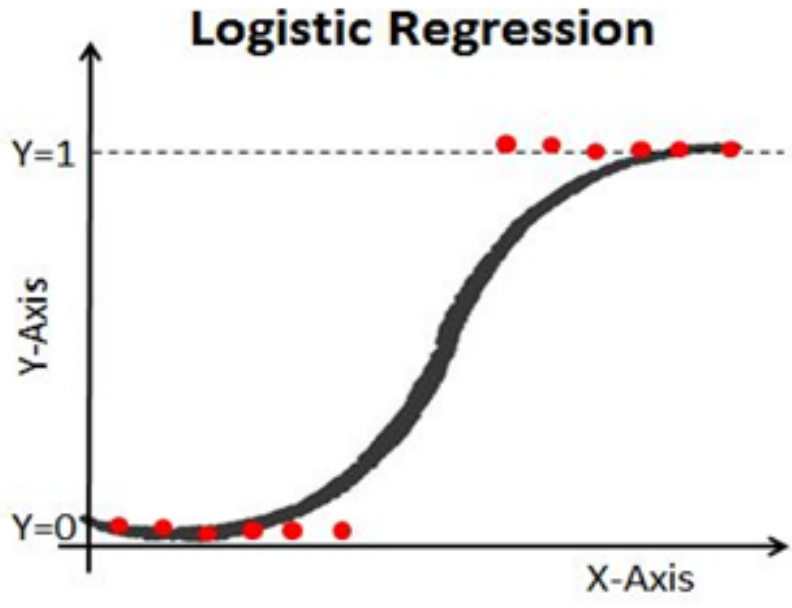

Figure 4.1: Graphical representation of logistic regression

4.2 Logistic Regression

Logistic Regression being the simplest and frequently used Machine Learning Classifier for data classification, it's pretty much easier to implement it as the baseline for any binary classification problem. Logistic Regression has got some basic fundamental concepts for constructive analysis in deep learning. Logistic regression formulizes and gauge the association between one dependent binary variable and independent variables as shown in Figure 4.1. This is very functional regression method for generalizing binary classification problems. Logistic Regression being efficient for spam email detection and Diabetes prophesy with more accurate results than linear regression has got our attention to implement the same algorithm for building an efficient model for the dataset we have to detect belligerent comments labeled as 'FALSE'. By implementing the model.

4.3 Support Vector Machine

Support Vector Machine (SVM), being supervised machine learning algorithm it is widely used these days for classification and regression challenges. It has got influence in mostly segregation problems. To understand the functionality of SVM, we have presented few figures below. The figure 4.2 depicts that each data item is plotted in n-dimensional space (where 'n' represents the number of features). And every data item known as feature of the specific dataset has got an associated value of a particular coordinate. After plotting each data item in n-dimensional space, here comes the classification challenge of the two classes of data. To address this problem of segregation, we need to find a suit-

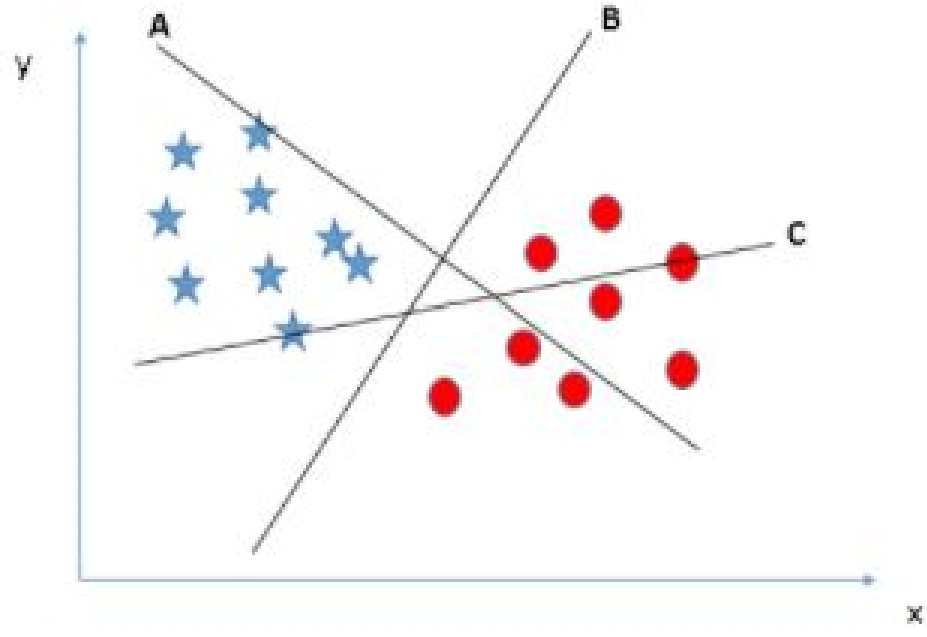

Figure 4.2: Visualization of Support vector machine [1]

able hyper-plane. The hyper-plane identification has its own principles for accurate segregation of the two classes. It should accurately bisect the data items belonging to each class. It should have maximum 'Margin' from each data item of both the classes, as shown in figure **??**. But SVM selects the hyper-plane which completely bisects the two classes prior to maximizing the 'Margin'. Sometimes linear segregation is not possible; therefore, an additional feature is introduced by the SVM to solve the riddle. The feature helps to maximize low dimensional input space to a higher dimensional input space i.e., it converts a non-separable problem to a separable problem and is useful for non-linear segregations. SVM uses 'Kernel trick' to perform such segregations.

$$z = x^2 + y^2$$

Where z is the additional feature that is introduced here via kernel trick. SVM works tremendously with clear margin of separation and high dimensional spaces. It is effective for cases with greater number of dimensions than the number of samples. It works with a subset of training points in the support vector and hence, it is also efficient in memory utilization. Besides all these benefits, SVM has got some limitations as well and they may include its less efficient performance with large datasets due to higher training time requirements. It is also less efficient with noisy datasets containing overlapping target classes. And last but not the least, SVM does not perform direct probability estimation and uses the expensive five-fold cross-validation to calculate these probabilities.

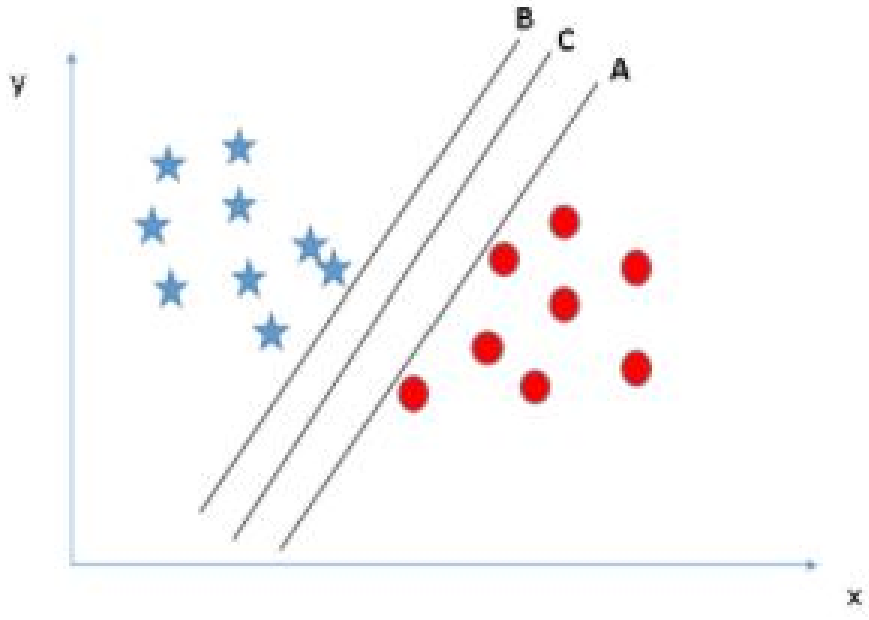

Figure 4.3: General classification hyperplane representation of SVM algorithms [1]

4.4 Random Forest

The most efficient among all the discussed above classifiers in our case is 'Random Forest'. It is a flexible and easy to use algorithm that generates excellent results even without parameter tunning. Its simplicity and diversity are so attention grabbing, that most of the researchers prefer to use it the most for segregation of classes of data and regression challenges.

Random Forest being supervised learning algorithm, builds an ensemble of decision trees usually trained with bagging method. The idea behind bagging technique is that the concatenation of learning models enhances the overall results. Therefore, random forest builds versatile decision trees and link them together in a way to achieve accurate and stable estimation. It is very helpful to measure the relative influence of each feature on estimation. We have different available tools which are provided by SKlearn to help us determine the importance of a feature by just looking to the number of nodes that uses it and reduce impurity across all trees in the forest.

4.5 Deep Learning Classifier

For a huge amount of data in datasets, deep learning models are performing in terms of data classification and prediction. Input data can be of type image, text, or speech, etc. Recurrent neural network (RNN) emerged as efficient learners of sequential data. As the text is the sequence of words and preserving sequence is very important to interpret the actual context of the text, RNN based models are most suitable for text analysis [28, 29]. RNN's are suitable for sequential data because, unlike other

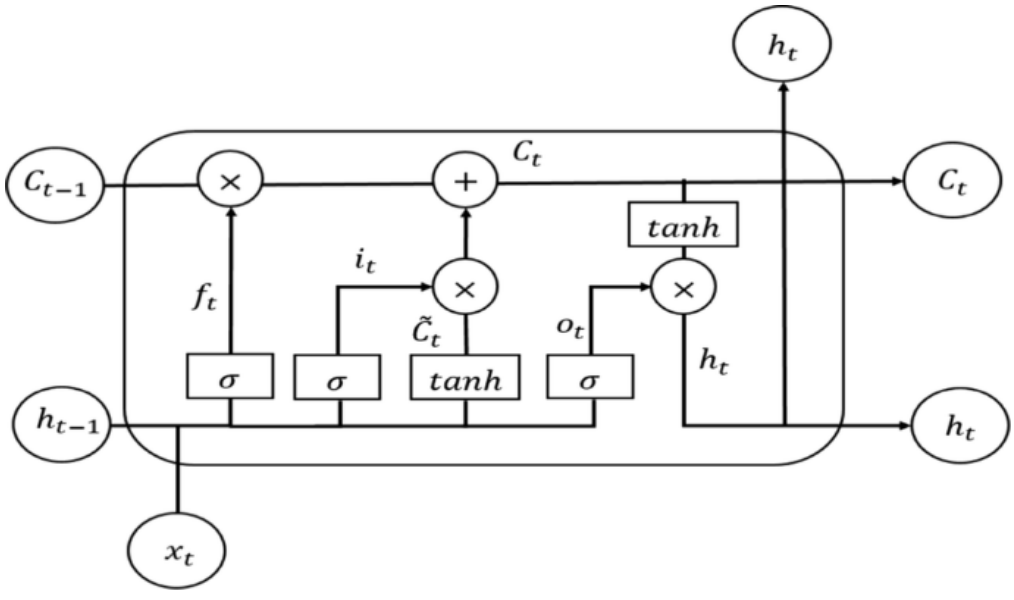

Figure 4.4: LSTM Repeating module structure [2]

neural networks where all the inputs are independent of each other, in RNN inputs are interrelated. Although RNNs are very powerful for learning sequences, they are practically vulnerable to vanishing gradient problem [30]. Vanishing gradient problem means RNN fails to remember the things in long past. It is possible that the sentiment of a document can highly rely on the beginning portion of the text. So, it can lead to the misclassification of the text document as simple RNN cannot remember the long-term dependencies. To handle the vanishing gradient problem, an improved variant of simple RNN is developed which is called LSTM [28].

4.6 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a variant of RNN, a class of deep neural network (DNN). LSTMs are capable of handling long term dependencies [29]. They were introduced by Hochreiter Schmidhuber (1997) and were refined and popularized by many people [30]. A beauty of LSTMs is remembering things for long term which traditional RNN can't handle properly [31]. In LSTMs there is a concept of memory cells that uses gates to keep or throw away information. The gates are named as forget gate, input gate, and output gate. Layers of neural networks such as sigmoid and tanh are used in the cell architecture as shown in Figure. 4.4 [32].

Where Xt is the input of LSTM block at current time stamp. Output of hidden layer ht is calculated as follows

$$f_t = \sigma(w_f\,[h_{t-1}, X_t] + b_f), \tag{4.2}$$

$$i_t = \sigma(w_i\,[h_{t-1}, X_t] + b_i), \tag{4.3}$$

$$O_t = \sigma(w_o\,[h_{t-1}, X_t] + b_o), \tag{4.4}$$

$$C_t = f_t * C_{t-1} + i_t * \tanh\,(w_c\,[h_{t-1}, X_t] + b_c), \tag{4.5}$$

$$h_t = O_t * \tanh\,(C_t), \tag{4.6}$$

where *ft* is forget gate, it is input gate and *Ot* is output gate. *ht-1* is the output of previous hidden state. *Ct-1* and *Ct* are previous and current cell memories. *wf*, *wi* and *wo* are weight matrices for different gates while *bf*, bi and bo are bi as vectors for these gates. σ is the sigmoid activation function which results in either 1 or 0 and tanh is the hyperbolic tangent activation function for neural network layer. The input *Xt* and *ht-1* are given to all the three gates. Forget gate will throw away the unnecessary information while the input gate layer decides which values needs to be updated and kept. Based on this information, old cell state *Ct-1* is updated into new cell state *Ct* by using multiplication and addition operations. Final output is based on updated cell state but will be passed through a sigmoid function which will decide which part of cell state are going to be the part of output.

In this research work, bidirectional LSTM model is used. BLSTM networks uses 2 hidden layers i.e., a backward hidden layer and a forward hidden layer [5]. Each layer works as same LSTM layer, but they bring bidirectional memory for the network. BLSTM network passes information in 2 ways i.e., from past to future and from future to past as shown in **Fig. 4.5**. Thus BLSTM is the proper solution having stronger memory to store all the useful features both previous and future with high precision. Keeping a record of the sequence of words in any document is very important in NLP for better classification, contextual analysis, and future prediction. For this purpose, a sequential model of the data set using words embedding is developed. For data classification, a deep learning-based model using BLSTM (Bidirectional Long Short-Term Memory) and dense layers is used. 64 nodes in the BLSTM layer with 0.2% dropout followed by 64 nodes in the dense layer is used. ReLU (Rectified Linear Unit) activation function is used for dense layer nodes while for the output layer, the Sigmoid activation function is used. The sigmoid activation function is used at the output layer for binary classification. Before BLSTM, a dropout layer has also been introduced in the model to minimize the overfitting problem. It randomly ignores certain neurons in each iteration which may be considered in the next iteration. Table 4.1 shows the summary of the compiled model.

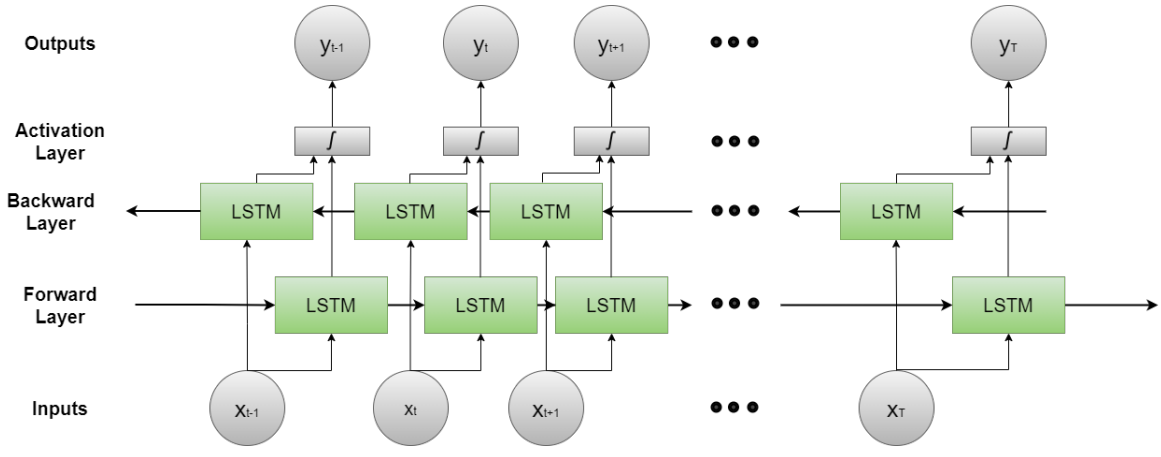

Figure 4.5: General classification hyperplane representation of BLSTM algorithms

Table 4.1: Parameters set for BLSTM

Layer(type)	Output Shape	Param
Embedding	(None,260,64)	1431808
Spatial dropout1d	(None,260,64)	0
Bidirectional	(None,128)	66048
Dense	(None,64)	8256
Dense1	(None,1)	65

Chapter 5

RESULTS DISCUSSION

In this work, four different machine learning algorithms and a deep learning model are used to classify the offensive roman Pashto words. The two main approaches are named BoW and TF-IDF for feature extraction. Both these approaches use N-Gram words but cannot keep the sequence information of words. We have used Tri Gram (uni +bi+ tri) words as features. After pre-processing of the dataset, its features were extracted using count vectorizer and TF-IDF vectorizer methods. Most frequent features ranging from 100 to 1000 are extracted and stored it in different .CSV (comma separated values) files. For experimental results, publicly available open-source software WEKA is used. The algorithms are evaluated for all the features and it was found out that the accuracy of all the algorithms were converging for 650 features, so we sed 700 most frequent features as standard for performance analysis.

Metric used to evaluate the performance of the proposed classifiers are accuracy, precision, and F-measure using 10-fold cross-validation.

5.1 Performance Comparison of Traditional/ Shallow Classifiers

Experimental results show that the smallest value of accuracy has been achieved from Naïve Bayes algorithm for both BoW and TF-IDF approaches. Its accuracy value converges at 80.7% for BoW and 73.7% for TF-IDF. An advantage of this algorithm is that it takes less time to build the classification model. SVM has performed very well as compared to Naïve Bayes and its classification accuracy for BoW converges at 92.76% as shown in figure 5.1. The TF-IDF gives an accuracy value of 93.05%. For both cases, SVM takes much more time to build the classification model. Logistic regression gives an accuracy of 93.16% for BoW and 93.63% for TF-IDF as shwon in figure 5.2. For

Table 5.1: Accuracy comparison of different classifiers using BoW

Performance / **Classifiers**	**Precision**	**F-Measure**	**Accuracy**
Naïve Bayes	85.3 %	80.0 %	80.77 %
Logistic	93.2%	93.2%	93.16%
SVM	93.4%	93.2%	93.18%
Random Forest	94.1%	94.1%	**94.07**%

Table 5.2: Accuracy comparison of different classifiers using TF-IDF

Performance / Classifiers	Precision	F-Measure	Accuracy
Naïve Bayes	84.9%	78.8%	73.99%
Logistic	93.7%	93.6%	93.63%
SVM	93.5%	93.4%	93.39%
Random Forest	94.0%	93.9%	**93.90%**

Table 5.3: Performance comparison of different classifiers

Performance / Classifier	Precision	F-Measure	Accuracy
BLSTM	97.22%	97.217%	**97.217%**
Naïve Bayes	85.3 %	80.0 %	80.77 %
Logistic	93.2%	93.2%	93.16%
SVM	93.4%	93.2%	93.18%
Random Forest	94.1%	94.1%	94.07%

classification model building, it takes half the less time than SVM. Random Forest outperforms all other algorithms in terms of accuracy, precision, and F-measure value. Its classification accuracy is 94.07% for BoW and 94.09% for TF-IDF. Its accuracy value is slightly higher, and it also takes less time for classification model building compared to logistic regression.

It is concluded from results analysis that Random Forest has given better results for the classification of offensive documents as compared to all other machine learning algorithms used in this research work. Tables 3 and 4 show a comparison of accuracy, precision, and F-measure value for all the algorithms for standard 700 features. Both BoW and TF-IDF have used tri-gram words for words vectorization and features extraction. Words sequence of occurrence in documents is not being considered. For words sequence, a sequential model with a deep learning algorithm is used for classification as shown in figure 5.3 and table 5.3.

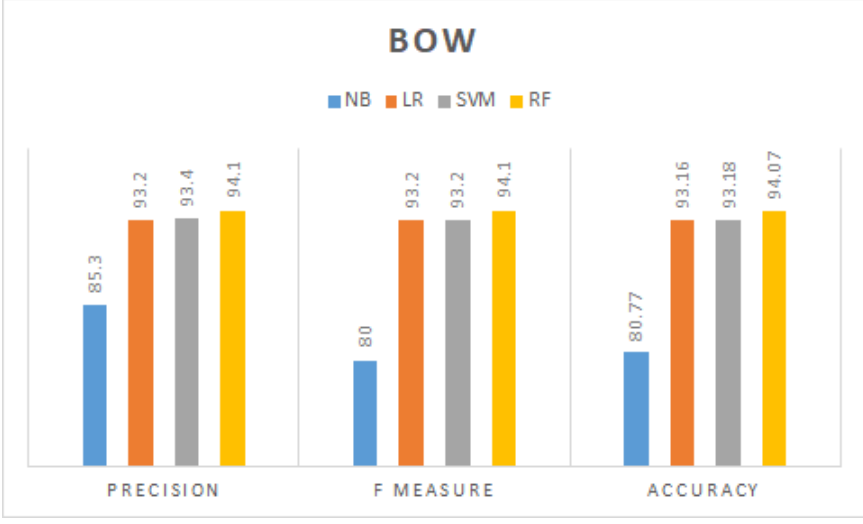

Figure 5.1: Performance comparison of classifiers with BoW

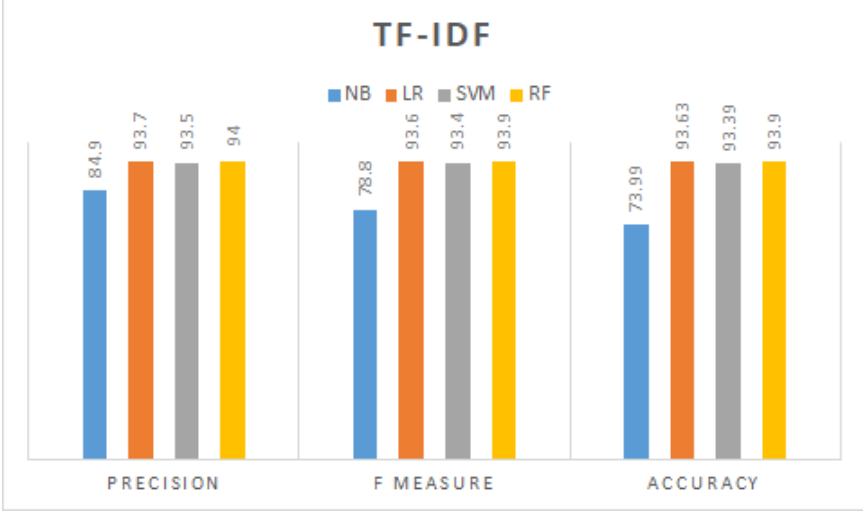

Figure 5.2: Performance comparison of classifiers with TF-IDF

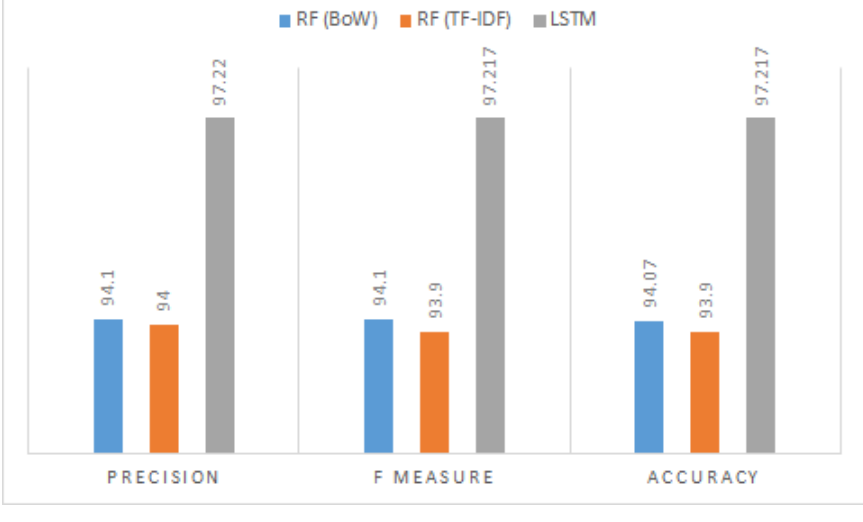

Figure 5.3: Overall performance comparison

Chapter 6

FUTURE WORK AND CONCLUSION

This thesis proposes an offensive text detection model for the roman Pashto language. To the best of our knowledge, no dataset for the roman Pashto language is available online. The dataset is compiled by gathering the comments from YouTube, Twitter, and Facebook. A total of 60 thousand comments were gathered for the creation of the roman Pashto corpus. One of the main contributions of this work is the collection and compilation of the roman Pashto dataset. The corpus created will be made available for the researcher working in this domain. Both sequential and non-sequential models are used for offensive language detection. For the non-sequential model, 2 approaches named BoW and TF-IDF were used. Both methods use word n-grams as features. Four different machine learning classifiers were trained and tested. Naïve Bayes classifier took less time to build the model, but its classification accuracy was less than others. Logistic regressing took more time to build the model and it gave better accuracy than Naïve Bayes. Random Forest outperformed all the 4 classifiers with an overall accuracy of 94. 07%. To have a more accurate model for offensive text detection we also built a sequential model and used the deep learning algorithm BLSTM for classification. The overall highest accuracy of 97.21% is achieved using BLSTM. The corpus created in this work is made available to the researchers working in this domain. In the future, I am planning to collect the same amount of data in Unicode script for automatic detection of Pashto language.

Bibliography

[1] R. Sunil, "Understanding support vector machine(svm) algorithm from examples (along with code)," *Analytics Vidhya*.

[2] B.-S. Kwon, R.-J. Park, and K.-B. Song, "Short-term load forecasting based on deep neural networks using lstm layer," *Journal of Electrical Engineering & Technology*, vol. 15, no. 4, pp. 1501–1509, 2020.

[3] S. Lewandowsky, M. Jetter, and U. K. Ecker, "Using the president's tweets to understand political diversion in the age of social media," *Nature communications*, vol. 11, no. 1, pp. 1–12, 2020.

[4] K. Buchanan, L. B. Aknin, S. Lotun, and G. M. Sandstrom, "Brief exposure to social media during the covid-19 pandemic: Doom-scrolling has negative emotional consequences, but kindness-scrolling does not," *Plos one*, vol. 16, no. 10, p. e0257728, 2021.

[5] J. W. Patchin and P. D. S. Hinduja, "Tween cyberbullying."

[6] D. Ali and L. Xiaoying, "The influence of content and non-content cues of tourism information quality on the creation of destination image in social media: A study of khyber pakhtunkhwa, pakistan," *Liberal Arts and Social Sciences International Journal (LASSIJ)*, vol. 5, no. 1, pp. 245–265, 2021.

[7] S. Kiritchenko, I. Nejadgholi, and K. C. Fraser, "Confronting abusive language online: A survey from the ethical and human rights perspective," *Journal of Artificial Intelligence Research*, vol. 71, pp. 431–478, 2021.

[8] H. Watanabe, M. Bouazizi, and T. Ohtsuki, "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection," *IEEE access*, vol. 6, pp. 13 825–13 835, 2018.

[9] A. Bisht, A. Singh, H. Bhadauria, J. Virmani *et al.*, "Detection of hate speech and offensive language in twitter data using lstm model."

[10] A. Balayn, J. Yang, Z. Szlavik, and A. Bozzon, "Automatic identification of harmful, aggressive, abusive, and offensive language on the web: A survey of technical biases informed by psychology literature," *ACM Transactions on Social Computing (TSC)*, vol. 4, no. 3, pp. 1–56, 2021.

[11] C. S. Park, Q. Liu, and B. K. Kaye, "Analysis of ageism, sexism, and ableism in user comments on youtube videos about climate activist greta thunberg," *Social Media+ Society*, vol. 7, no. 3, p. 20563051211036059, 2021.

[12] A. Klein, "Social networks and the challenge of hate disguised as fear and politics," *Journal for deradicalization*, no. 26, pp. 1–33, 2021.

[13] F. Husain and O. Uzuner, "A survey of offensive language detection for the arabic language," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 20, no. 1, pp. 1–44, 2021.

[14] F. A. Vargas, I. Carvalho, F. R. de Góes, F. Benevenuto, and T. A. S. Pardo, "Building an expert annotated corpus of brazilian instagram comments for hate speech and offensive language detection," *arXiv preprint arXiv:2103.14972*, 2021.

[15] M. Herath, T. Atapattu, H. A. Dung, C. Treude, and K. Falkner, "Adelaidecyc at semeval-2020 task 12: Ensemble of classifiers for offensive language detection in social media," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 2020, pp. 1516–1523.

[16] S. Dowlagar and R. Mamidi, "Hasocone@ fire-hasoc2020: Using bert and multilingual bert models for hate speech detection," *arXiv preprint arXiv:2101.09007*, 2021.

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

[18] D. Bhimani, R. Bheda, F. Dharamshi, D. Nikumbh, and P. Abhyankar, "Identification of hate speech using natural language processing and machine learning," in *2021 2nd Global Conference for Advancement in Technology (GCAT)*. IEEE, 2021, pp. 1–4.

[19] A. Alakrot, L. Murray, and N. S. Nikolov, "Towards accurate detection of offensive language in online communication in arabic," *Procedia computer science*, vol. 142, pp. 315–320, 2018.

[20] M. O. Ibrohim and I. Budi, "A dataset and preliminaries study for abusive language detection in indonesian social media," *Procedia Computer Science*, vol. 135, pp. 222–229, 2018.

[21] J. M. Schneider, R. Roller, P. Bourgonje, S. Hegele, and G. Rehm, "Towards the automatic classification of offensive language and related phenomena in german tweets," in *14th Conference on Natural Language Processing KONVENS*, 2018, p. 95.

[22] Z. TORWALI, "Language documentation and description."

[23] C. E. Basch, C. H. Basch, G. C. Hillyer, and C. Jaime, "The role of youtube and the entertainment industry in saving lives by educating and mobilizing the public to adopt behaviors for community mitigation of covid-19: successive sampling design study," *JMIR public health and surveillance*, vol. 6, no. 2, p. e19145, 2020.

[24] T. Sajid, M. Hassan, M. Ali, and R. Gillani, "Roman urdu multi-class offensive text detection using hybrid features and svm," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*. IEEE, 2020, pp. 1–5.

[25] M. A. K. M. T. Shibli Nisar, Ibrahim Shahzad, "Pashto spoken digits recognition using spectral and prosodic based feature extraction," *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*.

[26] M. T. Shibli Nisar, "Dialect recognition for low resource language using an adaptive filter bank," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 16, no. 4, p. 1850031, 2018.

[27] Z. ZAHID *et al.*, "The current situation of teaching and learning pashto and dari languages at primary level in afghanistan," *NUE Journal of International Educational Cooperation*, vol. 13, pp. 1–7, 2020.

[28] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, "Deep learning methods for forecasting covid-19 time-series data: A comparative study," *Chaos, Solitons & Fractals*, vol. 140, p. 110121, 2020.

[29] J. Mejia, L. Avelar-Sosa, B. Mederos, E. S. Ramírez, and J. D. D. Roman, "Prediction of time series using an analysis filter bank of lstm units," *Computers & Industrial Engineering*, vol. 157, p. 107371, 2021.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] D. Q. Zeebaree, A. M. Abdulazeez, L. M. Abdullrhman, D. A. Hasan, and O. S. Kareem, "The prediction process based on deep recurrent neural networks: A review," *Asia J. Res. Comput. Sci*, vol. 11, pp. 29–45, 2021.

[32] K. Smagulova and A. P. James, "A survey on lstm memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019.