JADAVPUR UNIVERSITY

Hate Speech Detection in Social Media messages (Tweets) using BERT model

By,

Somashree Nandy

Master of Computer Application - III

Class Roll No.: 002010503024
Registration No.: 154232 of 2020-2021
Examination Roll No.: MCA2360027
Under the supervision of

Prof. (Dr.) DIGANTA SAHA

Project submitted in partial fulfilment for the degree of

Master of Computer Application

in the

Department of Computer Science & Engineering
FACULTY OF ENGINEERING AND TECHNOLOGY

2023

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

To Whom It May Concern

I hereby recommend that the project entitled "**Hate Speech Detection in Social Media messages (Tweets) using BERT model**" has been carried out by **SOMASHREE NANDY** (Reg. No.:154232 of 2020-2021, Roll No: 002010503024) under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of **MASTER of COMPUTER APPLICATION** in **DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY** during the academic year 2022-23.

Prof.(Dr.) Diganta Saha
Project Supervisor
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-700032

Prof.(Dr.) Nandini Mukhopadhyay
Head of the Department
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-700032

Dean, Faculty Council of Engineering & Technology
Jadavpur University, Kolkata-700032

CERTIFICATE OF APPROVAL

This is to certify that the project entitled "**Hate Speech Detection in Social Media messages (Tweets) using BERT model**" is a bonafide record of work carried out by **SOMASHREE NANDY** in fulfilment of the requirements for the award of the degree of ***Master of Computer Application*** in ***the Department of Computer Science and Engineering, Jadavpur University***. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

.....	.....
Signature of Examiner 1	Signature of Examiner 2
Date:	Date:

DECLARATION OF ORIGINALITY AND COMPILANCE OF ACADEMIC PROJECT

This is to certify that the work in the project entitled "**Hate Speech Detection in Social Media messages (Tweets) using BERT model**" submitted by **SOMASHREE NANDY** is a record of an original research work carried out by her under the supervision and guidance of **Prof.(Dr.) Diganta Saha** for the award of the degree of ***Master of Computer Application*** in the ***Department of Computer Science and Engineering, Jadavpur University, Kolkata-32***. Neither this project nor any part of it has been submitted for any degree or academic award elsewhere.

Name : SOMASHREE NANDY

Class Roll No. : 002010503024

Registration No. : 154232 of 2020-2021

Project Title : **Hate Speech Detection in Social Media messages (Tweets) using BERT model**

Signature :

ACKNOWLEDGEMENT

First and foremost, I want to express my gratitude to God Almighty for providing me with the strength, wisdom, and capability to go on this amazing adventure and to continue and successfully finish the embodied research work. I'd like to thank **Professor (Dr.) Diganta Saha**, *Department of Computer Science and Engineering at Jadavpur University, my supervisor* and Dr. Arijit Das for their excellent assistance, consistent support, and inspiration during the course of my dissertation. I owe Jadavpur University a great debt of gratitude for providing me with the chance and facilities to complete our thesis.

I would also like to thank Mr. Zeerak Talat, Mrs. Sara Tonelli, Mrs. Melanie Siegel for providing us the annotated datasets.

I am grateful to everyone of the teaching and non-teaching personnel whose assistance has made our trip during my research time much easier.

I would like to thank my batch mates, my seniors, and my friends especially Rupam Saha and Srijan Das for providing me with regular encouragement and mental support throughout this curriculam.

Last but not the least, my family deserves great recognition. There are no words to express my gratitude to my mother and father for all of the sacrifices you've made on my behalf. Your prayers for me have kept me going thus far.

Somashree Nandy
Master of Computer Application
Department of Computer Science & Engineering
Class Roll: 002010503024
Exam Roll: MCA2360027
Jadavpur University

Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

NLP : Natural language processing

ML : Machine Learning

BERT : Bidirectional Encoder Representations from Transformers

CBOW: Continuous Bag of Words

Chapter 1: ABSTRACT

Hate speech is harmful online content that directly attacks or promotes hatred against members of groups or individuals based on actual or perceived aspects of identity, such as racism, religion, or sexual orientation. This can affect social life on social media platforms as hateful content shared through social media can harm both individuals and communities. As the prevalence of hate speech increases online, the demand for automated detection as an NLP task is increasing. In this project, the proposed method is using transformer based model i.e. BERT to detect hate speech in social media texts, mainly in tweets. This model works well in various languages such as Bengali, English, German, Italian. The success rate of using this model for hate speech detection is high enough to prove that BERT model is appropriate model for hate speech detection (Accuracy in Bengali dataset is 89%, in English: 91%, in German dataset 91% and in Italian dataset it is 77%).

Keywords: Social media, Hate Speech, NLP, ML, Deep Learning, English, Bengali, German, Italian, Hate Speech detection, Text classification, Abusive, racism;

Chapter 2: INTRODUCTION

Overview:

Online hate speech is a serious issue with potential consequences for both individuals and the community. It can be used to stir up conflict, advance prejudice, and create an atmosphere aggressive to vulnerable groups.

Online hate speech detection can be done in a variety of ways. Utilizing machine learning algorithms to recognize text that contains hate speech is one method. Large datasets of text that have been classified as hate speech or not are used to train these algorithms. Once the algorithm has been trained, it can be used to detect hate speech in messages.

The use of human moderators to examine content is another method for detecting hate speech online. Compared to machine learning algorithms, this method may be more effective, but it is also costlier and time-consuming.

The detection of online hate speech has some challenges also. One issue is that the definition of hate speech can be arbitrary and different depending on the culture. Another issue is that hate speech can be covered up in a variety of ways, such as by using code words or symbols.

Despite the difficulties, detecting hate speech online is a crucial tool for preventing it and fostering an inclusive online community.

The following are some advantages of online hate speech detection:

- Online platforms can be made safer for users by identifying and removing hate speech from them.
- It can aid in increasing public awareness of the issue of hate speech and its detrimental effects on society.
- Users can benefit from being informed about hate speech definitions and reporting procedures.

- It might contribute to creating a more accepting and compassionate online community.

Motivation:

The detection of hate speech is driven by a variety of factors. The following are a few of the most typical causes:

- To shield customers from harm. Online environments can become hostile as a result of hate speech, which can be harmful to both individuals and groups. By identifying and removing hate speech, hate speech detection can help make online platforms safer for users.

- To increase understanding of the issue with hate speech. The issue of hate speech must be brought to light because it is a serious one. The problem can be brought to light and people can be inspired to speak out against it with the aid of hate speech detection.

- To create a more tolerant and inclusive online community. Online environments hostile due to hate speech can make it difficult for users to feel safe and welcomed. By removing hate speech and educating users about the problem, hate speech detection can contribute to the development of a more welcoming and tolerant online community.

- To inform users about offensive speech. The definition of hate speech is not widely known, and many people may not know how to report it. Users can learn about hate speech and how to report it from hate speech detection.

Contribution:

The BERT model has been shown to be effective in hate speech detection. Another advantage of using BERT model is: it uses its own embedding to represent words into vector. BERT embedding is also one of the most efficient embedding method since it uses contextualize embedding of words to represent words into vectors.

My contribution in hate speech detection using BERT model is:

- Collecting various language datasets (Bengali, English, German, Italian)
- Preprocessing the datasets
    - Replacing mentions, links, multiple white spaces
- Stop word removal from the dataset
- Embedding of texts (BERT embedding) as well as of emojis
- Splitting the dataset into 60% train, 20% test, and 20% valid using split-train-valid
- Training the model using training part
- Testing the model and validation checking of the model
- Got the accuracy of the model for different languages which were quite good (for example, 89% for Bengali dataset)

Chapter 3: STATE OF THE ART

Let's look back at the history of hate speech detection.

Table-1

Year	Event
2006	The first hate speech detection system for social media was developed by John Dunning and Emily Thornburg.
2009	The first large-scale hate speech detection system for social media was developed by Michael Bernstein and Jeffrey Cohn.
2012	The first hate speech detection system for social media to use machine learning was developed by John Hewitt and Patrick Juola.
2015	The first hate speech detection system for social media to use deep learning was developed by the University of California, Berkeley.
2017	The first hate speech detection system for social media to be used in a commercial product was developed by Twitter.
2019	First use of multilingual hate speech detection
2020	The first hate speech detection system for social media to be used by a government was developed by the United States Department of Homeland Security.

In the next chapter, previous research works so far will be discussed.

Chapter 4: Previous Research Works

In recent years a lot of work has been done in the field of "Sentiment Analysis on Social Media" by number of researchers. Majority of the work have been performed on Twitter data i.e. tweets.

Hate speech detection has gained more and more attentions in recent years. Offensive comments such as hate speech and cyberbullying are the most researched areas in NLP in the past decades.

Table-2:

Year	*Title of the Paper*	*Author*	*Publication*	*Overview*	*Result*
2022	Detecting racism and xenophobia using deep learning models on Twitter data: CNN, LSTM and BERT [1]	Benítez-Andrades, J.A., González-Jiménez, Á., López-Brea, Á., Aveleira-Mata, J., Alija-Pérez, J.M. and García-Ordás, M.T.	*PeerJ Computer Science* , *8*, p.e906.	Author proposed a novel approach for detecting racism and xenophobia on Twitter using deep learning models. They evaluate the performance of three different models: CNN, LSTM, and BERT.	They find that BERT outperforms the other two models, achieving an F1 score of 85.22%.
2022	BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection [2]	Khan, S., Fazil, M., Sejwal, V.K., Alshara, M.A., Alotaibi, R.M., Kamal, A. and Baig, A.R.	*Journal of King Saud University-Computer and Information Sciences*, *34*(7), pp.4335-4344	The model, called BiCHAT, combines the strengths of bidirectional LSTM (BiLSTM), deep convolutional neural network (CNN), and hierarchical attention.	BERT based contextual embedding,Bichat with 89% success rate in english tweets
2022	Emotion Based Hate Speech Detection using Multimodal Learning [3]	Rana, A. and Jha, S.	*arXiv preprint arXiv:2202.06218*.	The paper proposes a multimodal deep learning framework for hate speech detection in multimedia data.	The result of precision, recall, and f1-score using the BERTA+CLS model is 93.00, 92.89 and 92.94.
2021	Towards generalisable hate speech detection: a review on obstacles and solutions [4]	Yin, W. and Zubiaga, A.	*PeerJ Computer Science*, *7*, p.e598.	This paper reviews the obstacles and solutions to generalisable hate speech detection, and proposes directions for future research.	Achieved only a precision of around .234 and a recall of 0.098 for the implicit class, in contrast to .864 and .936 for non-abusive and .640 and .509 for explicit.

2021	HATECHECK: Functional Tests for Hate Speech Detection Models [5]	Röttger, P., Vidgen, B., Nguyen, D., Waseem, Z., Margetts, H. and Pierrehumbert, J.B.	*arXiv preprint arXiv:2012.15606*.	HateCheck is a suite of functional tests for hate speech detection models. It consists of 29 tests that evaluate model performance on a variety of types of hateful or non-hateful content.	Accuracy for Hateful class is 89.5% and Non-hateful is 48.2%
2021	Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review [6]	Mullah, N.S. and Zainon, W.M.N.W.	*IEEE Access*, *9*, pp.88364-88376.	The paper discusses the challenges of hate speech detection, the different machine learning algorithms that have been used for this task, and the evaluation metrics that are used to measure performance.	Model precision 0.67, Recall 0.8, F-measure 0.72
2021	Racism, Hate Speech, and Social Media: A Systematic Review and Critique [7]	Matamoros-Fernández, A. and Farkas, J.	A systematic review and critique. *Television & New Media*, *22*(2), pp.205-224.	It provides a systematic review of the literature on racism, hate speech, and social media. The paper identifies the key challenges and issues in this area, and it provides recommendations for future research.	For term "hate speech", 67.65% on quantitative methods, 11.77% on qualitative method. For "racism", 59.26% on qualitative methods, 16.67% on quantitative methods.
2021	Human-in-the-Loop for Data Collection: a Multi-Target Counter Narrative Dataset to Fight Online Hate Speech [8]	Fanton, M., Bonaldi, H., Tekiroglu, S.S. and Guerini, M.	*arXiv preprint arXiv:2107.08720*	It proposes a novel human-in-the-loop data collection methodology to generate high-quality counter narratives to fight online hate speech.	This paper does not report any accuracy results. The paper focuses on the development of a methodology for collecting hate speech and counter-narrative data, and it does not evaluate the accuracy of any models that were trained on this data.
2020	Resources and benchmark corpora for hate speech detection: a systematic review [9]	Poletto, F., Basile, V., Sanguinetti, M., Bosco, C. and Patti, V.	*Language Resources and Evaluation*, *55*, pp.477-523.	It systematically analyzes the resources made available by the community at large for hate speech detection.	The paper does not report any accuracy results. The paper focuses on the identification and description of resources and benchmark corpora for hate speech detection, and it does not evaluate the accuracy of any models that were trained on these resources.
2020	A Multilingual Evaluation for Online Hate	Corazza, M., Menini, S., Cabrio, E.,	*ACM Transactions on Internet Technology (TOIT)*, *20*(2), pp.1-22.	It presents a multilingual evaluation of hate speech detection systems. The evaluation is conducted on three languages: English,	Max F1 score of English 0.823, of Italian 0.805, and German 0.758

	Speech Detection [10]	Tonelli, S. and Villata, S.		Italian, and German. It has used FastText embedding and LSTM model.	
2020	Online Multilingual Hate Speech Detection: Experimenting with Hindi and English Social Media [11]	Vashistha, N. and Zubiaga, A.	*Information*, *12*(1), p.5.	It explores the use of machine learning algorithms to detect hate speech in Hindi and English social media.	Accuracies are 71.75, 66.7%, 66.6% and 69.8% by SVM, Random Forest, Hierarchical LSTM with attention and Sub-word level LSTM model respectively.
2020	Hate speech detection and racial bias mitigation in social media based on BERT model [12]	Mozafari, M., Farahbakhsh, R. and Crespi, N.	*PloS one*, *15*(8), p.e0237861.	It proposes a novel approach to hate speech detection in social media that mitigates racial bias.	Accuracy is 82.4% using BERT baseline and 84.4% by BERT with bias mitigation
2020	Deep Learning Models for Multilingual Hate Speech Detection [13]	Aluru, S.S., Mathew, B., Saha, P. and Mukherjee, A.	*arXiv preprint arXiv:2004.06465*.	The paper proposes a framework for multilingual hate speech detection using deep learning models. The framework is evaluated on a dataset of tweets in 9 languages, and it achieves state-of-the-art results.	Accuracy using CNN-BiLSTM is 87.0%, using BERT 91.0%, using DistilBERT is 90% and using XLNET is 92%.
2020	Automatic Hate Speech Detection using Machine Learning: A Comparative Study [14]	Abro, S., Shaikh, S., Khand, Z.H., Zafar, A., Khan, S. and Mujtaba, G.	*International Journal of Advanced Computer Science and Applications*, *11*(8).	It compares the performance of different machine learning algorithms for hate speech detection. The authors found that the best performing algorithm was support vector machines (SVMs) with bigram features.	F1-score using SVM is 79%, using Naïve Bayes is 75%, using Decision Tree is 72%, using Random Forest 71%, using K-nearest Neighbors 69%, using Logistic Regression is 67%, using Multinomial Naïve Bayes is 65%, and Bernoulli Naïve Bayes is 63%
2020	A Framework for Hate Speech Detection Using Deep Convolutional Neural Network [15]	Roy, P.K., Tripathy, A.K., Das, T.K. and Gao, X.Z.	*IEEE Access*, *8*, pp.204951-204962.	The paper proposes a deep convolutional neural network (DCNN) framework for hate speech detection in social media. The framework uses GloVe word embeddings to represent the text of tweets, and it uses a DCNN to learn the semantic features of hate speech.	It achieves a F1 score of 0.92 by using DCCN, 0.77 using Logistic Regression and 0.64 using Naïve Bayes

2020	A Deep Learning Approach for Automatic Hate Speech Detection in the Saudi Twittersphere [16]	Alshalan, R. and Al-Khalifa, H.	*Applied Sciences*, *10*(23), p.8614	It proposes a deep learning approach for automatically detecting hate speech in Arabic tweets.	It achieves accuracy of 79% by CNN, 77% by GRU, 81% by CNN+GRU, and 83% by BERT.
2020	In Data We Trust: A Critical Analysis of Hate Speech Detection Datasets [17]	Madukwe, K., Gao, X. and Xue, B.	In *Proceedings of the Fourth Workshop on Online Abuse and Harms* (pp. 150-161).	It critically analyzes the datasets used for hate speech detection, identifying their limitations and recommending approaches for future research.	The paper does not report any accuracy results. The paper focuses on the analysis of the design and construction of hate speech detection datasets.
2020	A Comparative Study of Different State-of-the-Art Hate Speech Detection Methods for Hindi-English Code-Mixed Data [18]	Rani, P., Suryawanshi, S., Goswami, K., Chakravarthi, B.R., Fransen, T. and McCrae, J.P.	In *Proceedings of the second workshop on trolling, aggression and cyberbullying* (pp. 42-48).	This paper compares the performance of different state-of-the-art hate speech detection methods on a Hindi-English code-mixed dataset. The results show that deep learning models perform better than traditional machine learning models on this type of data.	It achieves accuracy of 71.7% by using SVM, 69.3% by Random Forest, 72.6% by Bidirectional LSTM, and 73.9% by using CNN.
2020	Detecting Hate Speech in Memes Using Multimodal Deep Learning Approaches: Prize-winning solution to Hateful Memes Challenge [19]	Velioglu, R. and Rose, J.	*arXiv preprint arXiv:2012.12975*.	It proposes a multimodal deep learning approach to detect hate speech in memes. The approach achieved an accuracy of 0.765 on the Hateful Memes Challenge test set	It reports an accuracy of 76.5% on the challenge test set.
2020	Detecting Hate Speech in multi-modal Memes [20]	Das, A., Wahi, J.S. and Li, S.	*arXiv preprint arXiv:2012.14891*.	It proposes a novel approach to detect hate speech in multi-modal memes by combining the text and image modalities.	It achieves accuracy of 67.2% using Concat BERT, 70.4% using Multimodal BERT, and 72.1% using Multimodal BERT + sentiment
2020	The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes [21]	Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P. and Testuggine, D.	*Advances in Neural Information Processing Systems*, *33*, pp.2611-2624.	The Hateful Memes Challenge is a benchmark for detecting hate speech in multimodal memes. It is constructed such that unimodal models struggle and only multimodal models can succeed	It achieves accuracy of 59.3% using Unimodal BERT, 64.73% by Multimodal ViLBERT CC, 68.4% by OSCAR+RF.

2020	EVALITA Evaluation of NLP and Speech Tools for Italian [22]	Basile, V., Maria, D.M., Danilo, C. and Passaro, L.C.	In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)* (pp. 1-7). CEUR-ws.	EVALITA is a biennial evaluation campaign that aims to promote the development of natural language processing and speech technologies for the Italian language.	The overall accuracy of the systems that participated in the 2020 EVALITA campaign was high, with an average accuracy of 85%. However, with some tasks, such as part-of-speech tagging, achieving an accuracy of over 90%, while others, such as sentiment analysis, achieving an accuracy of only 70%.
2019	Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter [23]	i Orts, Ò.G.	In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 460-463).	This paper presents a system for detecting hate speech against immigrants and women in Twitter, in multiple languages.	The Fermi system achieved accuracy of 0.651, the MITRE system achieved 0.729, the CIC-2 system achieved 0.727, the Panaetius system achieved 0.571 The baseline system achieved the lowest accuracy of 0.500.
2019	Hateful Speech Detection in Public Facebook Pages for the Bengali Language [24]	Ishmam, A.M. and Sharmin, S.	In *2019 18th IEEE international conference on machine learning and applications (ICMLA)* (pp. 555-560). IEEE	This paper proposes a machine learning approach to detect hateful speech in Bengali language posts on Facebook.	It achieved 52.20% accuracy using Random Forest, and 70.10% using a GRU based deep neural network.
2019	OFFENSIVE LANGUAGE AND HATE SPEECH DETECTION FOR DANISH [25]	Sigurbergsson, G.I. and Derczynski, L.	*arXiv preprint arXiv:1908.04531*.	It constructs a Danish dataset DKHATE containing user-generated comments from various social media platforms, and to their knowledge, the first of its kind, annotated for various types and target of offensive language. They develop four automatic classification systems, each designed to work for both the English and the Danish language.	It achieved a macro-averaged F1-score of 0.70
2019	Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation [26]	Arango, A., Pérez, J. and Poblete, B.	In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (pp. 45-54).	It constructs a Danish dataset DKHATE containing user-generated comments from various social media platforms, and to their knowledge, the first of its kind, annotated for various types and target of offensive language.	The results showed that accuracy of the models varied from 60% - 90%. Model1 achieves 60% accuracy, model2 70%, Model3 80% and Model4 achieved 90% accuracy respectively.

2019	A Levantine Twitter Dataset for Hate Speech and Abusive Language [27]	Mulki, H., Haddad, H., Ali, C.B. and Alshabani, H.	In *Proceedings of the third workshop on abusive language online* (pp. 111-118).	It introduces the first publicly-available Levantine Twitter dataset for the task of hate speech and abusive language detection. The dataset consists of 5,846 tweets from Syria and Lebanon, which have been manually labeled as normal, abusive, or hate.	It achieved accuracy of 90.5% using Naïve Bayes, 54.7% using SVM and 86.3% using Random Forest.
2018	Hate Speech Dataset from a White Supremacy Forum [28]	De Gibert, O., Perez, N., García-Pablos, A. and Cuadros, M.	*arXiv preprint arXiv:1809.04444*.	A dataset of 10,568 sentences extracted from a white supremacist forum, manually annotated as hate speech or not.	It achieved 85% accuracy using LSTM-based classifier and 80% accuracy using SVM based classifier.
2018	Effective hate-speech detection in Twitter data using recurrent neural networks [29]	Pitsilis, G.K., Ramampiaro, H. and Langseth, H.	*Applied Intelligence*, *48*, pp.4730-4742.	It proposes an ensemble of recurrent neural network classifiers to detect hate speech in Twitter data.	It achieved accuracy of 90% while trained on dataset of 10,000 tweets and achieved 88% accuracy while trained on a dataset of 16,000.
2018	A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection [30]	Bohra, A., Vijay, D., Singh, V., Akhtar, S.S. and Shrivastava, M.	In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media* (pp. 36-41).	It presents a dataset of Hindi-English code-mixed social media text that has been annotated for hate speech. The dataset can be used to train and evaluate hate speech detection models.	Accuracy of the model in this paper is 71.7%.
2018	A Survey on Automatic Detection of Hate Speech in Text [31]	Fortuna, P. and Nunes, S.	*ACM Computing Surveys (CSUR)*, *51*(4), pp.1-30.	It provides a comprehensive overview of the state-of-the-art in automatic hate speech detection in text.	The highest accuracy reported in the paper is 92.1%
2018	Characterizing and Detecting Hateful Users on Twitter [32]	Ribeiro, M.H., Calais, P.H., Santos, Y.A., Almeida, V.A. and Meira Jr, W.	In *Twelfth international AAAI conference on web and social media*	The paper proposes a method to characterize and detect hateful users on Twitter by analyzing their social network.	It achieved accuracy of 95%.
2017	Hate me, hate me not: Hate speech detection on Facebook [33]	Del Vigna12, F., Cimino23, A., Dell'Orletta, F., Petrocchi, M. and Tesconi, M.	In *Proceedings of the first Italian conference on cybersecurity (ITASEC17)* (pp. 86-95).	Hate speech detection on Facebook is a challenging task due to the evolving nature of hate speech and the need to balance accuracy with fairness.	It achieved accuracy of 78.3% using SVM, and 79.6% using LSTM model.

Chapter 5: PROBLEM STATEMENT AND ITS MATHEMATICAL NOTATION

Hate Speech Detection in Social Media messages (Tweets) using BERT model

Mathematical notation:

```
H = f(M, B)
```

where:

- H is the probability that a message M contains hate speech,
- f is a function that takes a message M and a BERT model B as input, and
- B is a BERT model that has been trained on a dataset of hate speech and non-hate speech messages.

The function f can be implemented as a neural network that takes the word embeddings of the message M as input and outputs a probability that the message contains hate speech. The BERT model B is used to learn the word embeddings of the message M.

The function f can be trained using a supervised learning approach, where the training data consists of a set of messages M that have been labeled as either hate speech or non-hate speech. The function f is then trained to predict the correct label for the messages in the training data.

Once the function f has been trained, it can be used to detect hate speech in new messages. The function f takes a new message M as input and outputs a probability that the message contains hate speech. If the probability is above a certain threshold, then the message is considered to contain hate speech.

The BERT model B can be pre-trained on a large dataset of text, such as the Wikipedia corpus. The pre-trained BERT model can then be fine-tuned on a smaller dataset of hate speech and non-hate speech messages. This fine-tuning process can improve the performance of the BERT model on the task of hate speech detection.

Chapter 6: BASIC CONCEPTS

➢ 6.1. Machine Learning

A subfield of artificial intelligence (AI) and computer science called machine learning focuses on using data and algorithms to simulate how humans learn, gradually increasing the accuracy of the system. The main goal is to create software for computers that can access data and use it to expand their knowledge. In order to find patterns and improve future decisions based on the data we provide, the learning process starts with data. The ultimate focus is to let computers learn on their own, without human assistance or intervention, and adjust their behavior accordingly.

A machine learning algorithm's learning system can be divided into three primary components:

a) A decision process: Machine learning algorithms are typically used to make a prediction or classification. Algorithm will generate an estimate about a pattern in the data based on some input data, which can be labeled or unlabeled.
b) An error function: It evaluates the model's prediction. If there are known examples, an error function can compare them to determine the model's accuracy.
c) Model Optimization Process: If the model fits the data points in the training set better, the weights are adjusted to reduce the difference between the known example and the model estimate. The algorithm will repeat this "evaluate and optimize" process, updating weights autonomously until an accuracy threshold is reached.

Supervised learning and unsupervised learning are the two main categories of machine learning algorithms. Supervised learning is a type of learning in which we train the machine using well-labeled data (training data). Following that, the

trained system is given a new set of data (test data) to evaluate the correct label of the data. Unsupervised learning is the process of developing a system using unlabeled data and allowing the algorithm to act on that data without supervision. The machine's task here is to group the data based on similarities, patterns, and differences without any prior knowledge of class value.

There is another learning method called semi supervised learning method. It uses a smaller labeled data set to guide classification and feature extraction from a larger, unlabeled data set during training. Semi-supervised learning can address the issue of insufficient labeled data for a supervised learning algorithm. It also helps if labeling enough data is prohibitively expensive.

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E, according to Tom Mitchell.

Machine learning is used in a variety of applications, including biometric identification, computer vision, game play, Natural Language Processing (NLP), recommendation systems, and financial market analysis, to name a few.

➢ 6.2. Natural Language Processing

A. Brief Introduction

NLP is the study of how natural (human) language is processed computationally. Teaching computers to generate (and understand) human language, to put it another way.

(U) -> Understanding (G) -> Generation

Figure - 1: Language Encoding and Decoding

NLP is HARD. Human language is heavily ambiguous. There are types of ambiguity as:

- Morphological: Joe is quite impossible. Joe is quite important.
- Phonetic: Joe's finger got number.
- Part of speech: Joe won the first round.
- Syntactic: Call Joe a taxi
- Pp Prepositional Phrase attachment: Joe ate pizza with a fork / with meatballs / with Samantha / with pleasure.
- Sense: Joe took the bar exam.
- Modality: Joe may win the lottery
- Subjectivity: Joe believes that stocks will rise
- Cc Conjunctional attachment: Joe likes ripe apples and pears.
- Negation: Joe likes his pizza with no cheese and tomatoes.
- Referential: Joe yelled at Mike. He had broken the bike. Joe yelled at Mike. He was angry at him.
- Reflexive: John bought him a present. John bought himself a present.
- Ellipsis and parallelism: Joe gave Mike a beer and Jeremy a 5 glass of wine.
- Metonymy: Boston called and left a message for Joe

There are a large variety of underlying tasks and machine learning models powering NLP applications. Recently, deep learning approaches have obtained very high performance across many different NLP tasks. These models can often be trained with a single end-to-end model and do not require traditional, task- specific feature engineering.

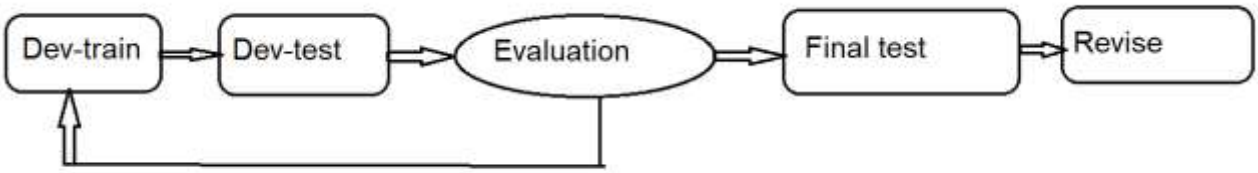


Figure 2: Train-Test-Evaluate cycle of machine learning

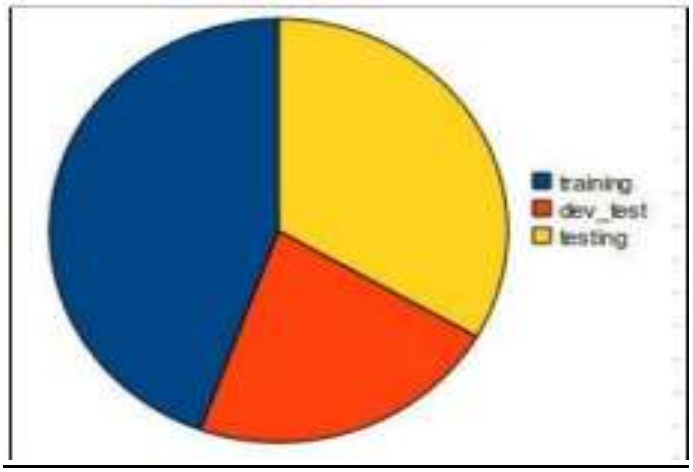


Figure 3: Typical corpus division

B. *Text Analysis*

Text Analytics, also called Text Mining, is the process of converting unstructured text data into meaningful data for analysis. Text Analytics tries to solve the crisis of information overload by combining techniques from data mining, machine learning, natural language processing, information retrieval, and knowledge management.

On a functional level, Text Analytics systems have four main areas:

- Pre-processing
- Core mining operations
- Presentation layer components
- Refinement techniques

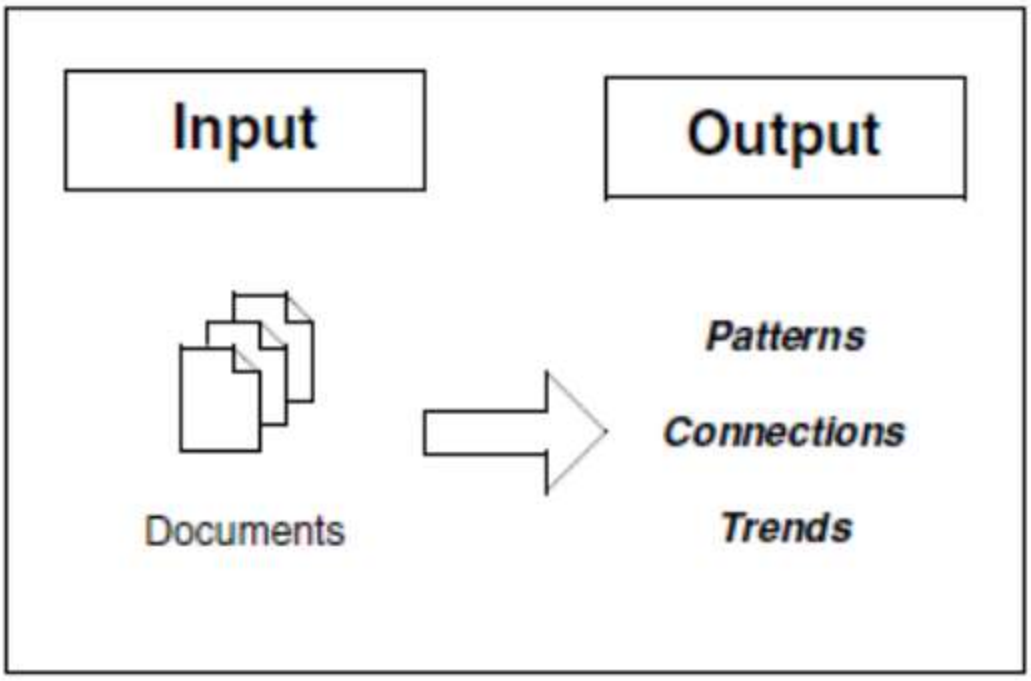


Figure 4: Simple Input-Output model

C. *Computational Linguistics*

This interdisciplinary field examines the nature of a language, including its morphology, syntax, and dynamic use, and develops any potentially useful models from this observation to support in the handling of language by machines.

D. *Sentimental Analysis*

The computational study of perceptions, decisions, and feelings toward things, events, and their characteristics is known as sentiment analysis. This is the process of identifying and extracting subjectivity from source documents using text analytics, computational linguistics, and natural language processing.

One common text classification task is sentiment analysis. Subjectivity analysis needs to be used at the statement level. Private feelings are expressed in the context of a text or conversation using subjective

analysis. A general term for opinions, assessments, feelings, and speculations is "private state." A subjective sentence expresses some personal feelings or beliefs, whereas an objective sentence expresses some factual information about the world.

Sentiment analysis can be applied on two different levels. Level 1 is the sentence level, which detects positive, negative and neutral sentiment for each sentence. Level 2 is the document level, which detects the whole document sentiment as one unit or one entity positive or negative or neutral.

E. *Tokenization*

Tokenization is the process of dividing a text document into tokens, which are discrete words.

Tokenization aims at segmenting words from running text.

Text: The child ate the cake with the fork

Tokens: ["the", "child", "ate", "the", "cake", "with", "the", "fork"]

➢ 6.3. Word Embedding

It is a technique of representing words into vectors. It maps word into vectors of real numbers using neural network architecture. Alternatively, we can say that it is distributed representation of text into an n-dimensional space.

There are few techniques available to perform word embedding, they are:

- One hot encoding
- TF-IDF
- Word2Vec
- Fast Text
- BERT
- Glove etc.

One hot embedding is Basic technique used to represent data numerically. But it has some disadvantages such as lack of meaning representation and creating d-dimensional vector for each instance which are sparse.

The full form of *TF* is **Term Frequency** (TF). **TF =** *no. of times terms occurrences in a document / total number of words in a document. The full form of IDF is* ***Inverse Document Frequency****.* ***IDF =*** *log e (total number of documents / number of documents which are having term).* ***TF - IDF = TF * IDF***

But this method also has some disadvantages

- Computing document similarity directly can be slow for large vocabularies
- Assuming counts of different words provides evidence of similarity
- No use of semantic similarities between words

Word2Vec is an unsupervised technique with supervised tasks. It takes text corpus as input and produces feature vectors. There are two architectures of word2vec, they are Skip-gram and CBOW (Continuous Bag of Words). CBOW predicts the probability of a word given some input context (neighboring words) whereas Skip-gram is complete opposite of CBOW, it predicts the context words (neighbour words) of given word. But the main disadvantage of this method is out of vocabulary words.

FastText Improves vector representation from skip-gram method. The modification to the skip-gram method is applied by doing sub-word generation and Skip-gram with negative sampling.

BERT is Bidirectional Encoder Representations from Transformers. It is used to extract features – word and sentence embedding vectors, from text data.

- Neural Machine Translation
- Question Answering
- Sentimental Analysis
- Text Summarization

These problems can be solved with the help of BERT. BERT training is done by pre-train and fine tuning.

Applications of word embedding are:

- Compute similarities between words
- Used for document clustering
- Sentiment analysis
- Use as features in text classification
- Text summarization
- Question Answering

➢ 6.4. NEURAL NETWORK

Neural networks are modeled after how the human brain processes information and are used to simulate some of the basic functions of the brain. Because of its quick computation and response times, it is used to complete a variety of real-time tasks.

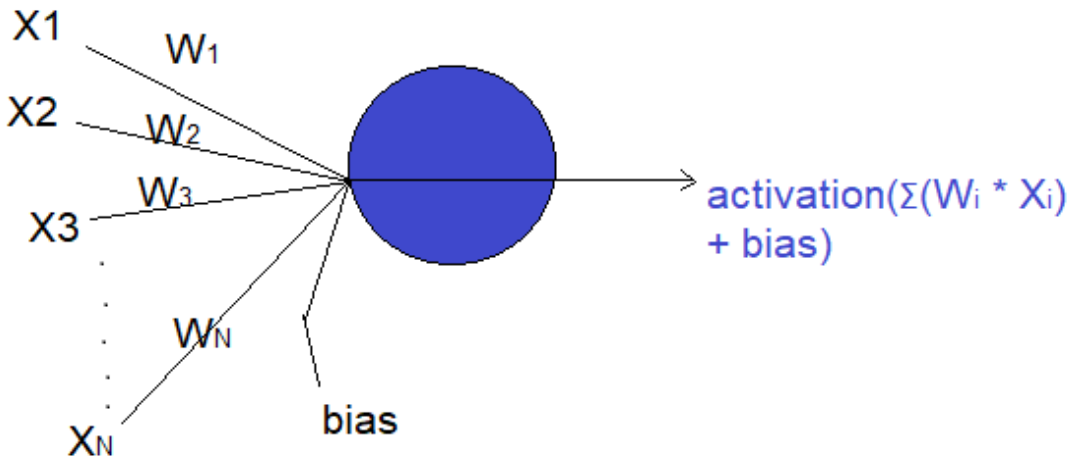


Figure -5: A single neuron with Xi inputs with their weights Wi, a bias and applied activation function

A model of an artificial neural network includes a number of elements that draw inspiration from the biological nervous system.

A large number of interconnected processing components, also referred to as Nodes, make up an artificial neural network. These nodes share a connection link with other nodes to connect them. Weights are present in the connection link and

these weights contain information about the input signal. These weights are updated with each iteration and input. The final weights of the neural network and its architecture are known as the "trained neural network" after all of the training data instances have been entered. The training of neural networks is the name of this procedure. To resolve particular issues as outlined in the problem statement, this trained neural network is used.

Types of Neural Network are ANN, CNN, RNN.

In ANN, because the inputs are sent forward, it is a feed-forward neural network. Additionally, it might have undiscovered layers that would add to the model's density. The programmer has set a fixed length for them. It is utilized for tabular or textual data.

In CNN, Image data is its main use. In computer vision, it is employed. Object detection in autonomous vehicles is one of the real-world applications. Convolutional layers and neurons are both present. It is more effective than RNN and ANN combined.

In RNN, Data from time series are processed and interpreted using it. The output from a processing node in this kind of model is fed back into nodes in the same or earlier layers.

There are three different learning types in neural networks, including

- supervised learning
- Unsupervised learning
- Reinforcement learning

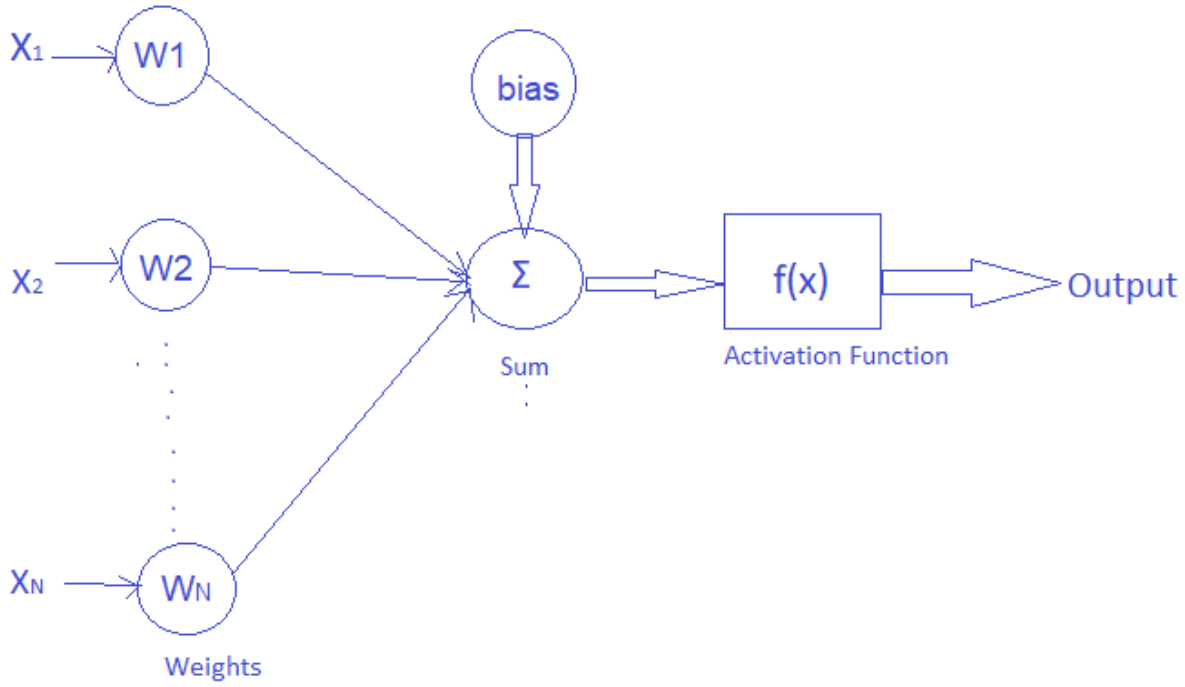


Figure-6: Working of Neural Network

A simple or multiple linear regression model with an activation function at the conclusion can be compared to an artificial neuron. In order to calculate the weighted sum and add bias to it, a neuron from layer i will use the output of every neuron from layer i-1 as inputs. The activation function is then contacted, as shown in the previous diagram. Every input from the preceding layer is connected to the first neuron from the first layer, and similarly, every input from the preceding layer is connected to the second neuron from the first hidden layer, and so on for all the neurons in the first hidden layer. Because they are connected to earlier neurons in the same way, the neurons in the second hidden layer (which are the outputs of the first hidden layer) are considered inputs. Forward propagation is the general term for this procedure. It is then compared to the actual output after we have predicted it. The loss is then calculated, and we attempt to reduce it and it is done by a method known as Back propagation. To try to minimize the loss, weights and biases are first adjusted after the loss has been calculated. With the aid of another algorithm called gradient descent, weights and biases are updated.

➢ 6.5. BERT model

In 2018, Google Research created the BERT NLP model, which has since achieved cutting-edge accuracy on a number of NLP tasks. BERT employs Transformer, a quirky technique for understanding the relationships between words (or small words) in a text. It is based on a deep learning model called Transformers, in which every input and output element is connected, and the weights between them are dynamically determined based on that connection.

Transformer architecture is known as encoder-decoder architecture because it has an encoder stack and a decoder stack, whereas BERT is only a part of transformer architecture. The architecture complexity of the two variants, BERT-base and BERT-large, varies. The base model's encoder has 12 layers, while the Large's encoder has 24 layers. BERT was developed using a large text corpus for training, which allows the architecture or model to learn the variability in data patterns and perform well across a variety of NLP tasks. Being bidirectional, BERT gathers knowledge from both the left and the right sides of the context of a token during training.

The input ids are frequently the only parameters that must be provided to the model as input. Token indices, or numerical representations of the tokens used to construct the sequences, will be used as the model's input.To avoid paying attention to the padding token indices, use the attention mask. If a token is NOT MASKED, the mask value can be either 0 or 1, and if it is MASKED, it can only be 0. Token type ids are employed in applications like question answering and sequence classification. Since these call for the encoding of two distinct sequences into the same input IDs. The sequences are divided using specialized tokens like the classifier[CLS] and separator[SEP] tokens.

The bidirectional Transformers at the core of BERT's design allow it to be the first NLP technique to solely rely on self-attention mechanisms. This is important because a word's meaning frequently changes as a sentence progresses. The overall meaning of the word that the NLP algorithm is focusing on is enhanced by each additional word. By reading in both directions, taking into account how all other words in a sentence affect the focus word, and removing the left-to-right

momentum that causes words to be skewed towards a particular meaning as a sentence progresses, BERT accounts for the augmented meaning.

Chapter 7: METHODOLOGY

➢ 7.1. Schematic of the System

In this section, we describe the technical details of the work done. In the process we also provide the descriptions of the type of data utilized as well as the probable output from the system. The impact of a project depends majorly on the outcome obtained from experimental results whereas the outcome can be considered as a function of inputs and proper implementation of technologies employed on those input. We have already discussed in previous research work section, various techniques available to detect hate speech based on the social media interaction messages. Here hate speech detection using the BERT model has been discussed.

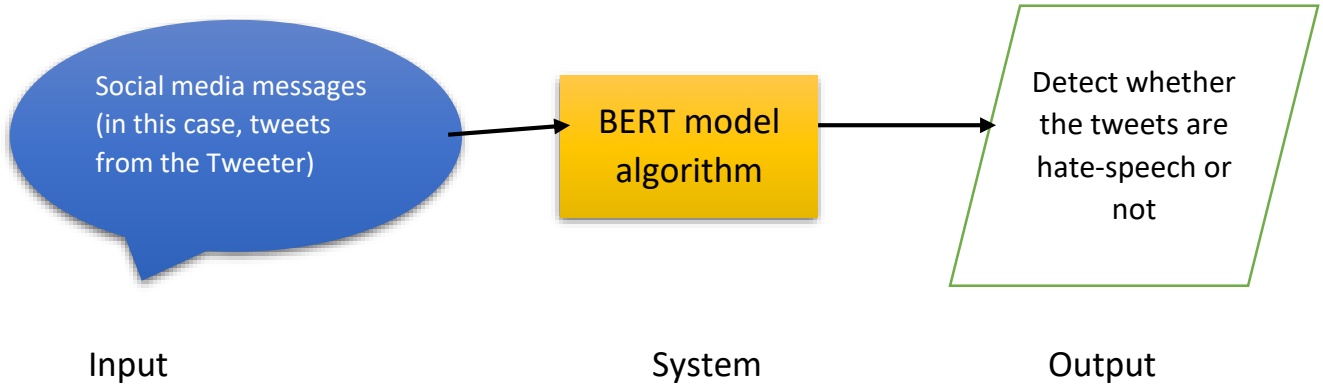


Figure-7: Schematic diagram of System

We mainly use here a Dictionary Based approach with different types of Lexical Resources on the social media messages.

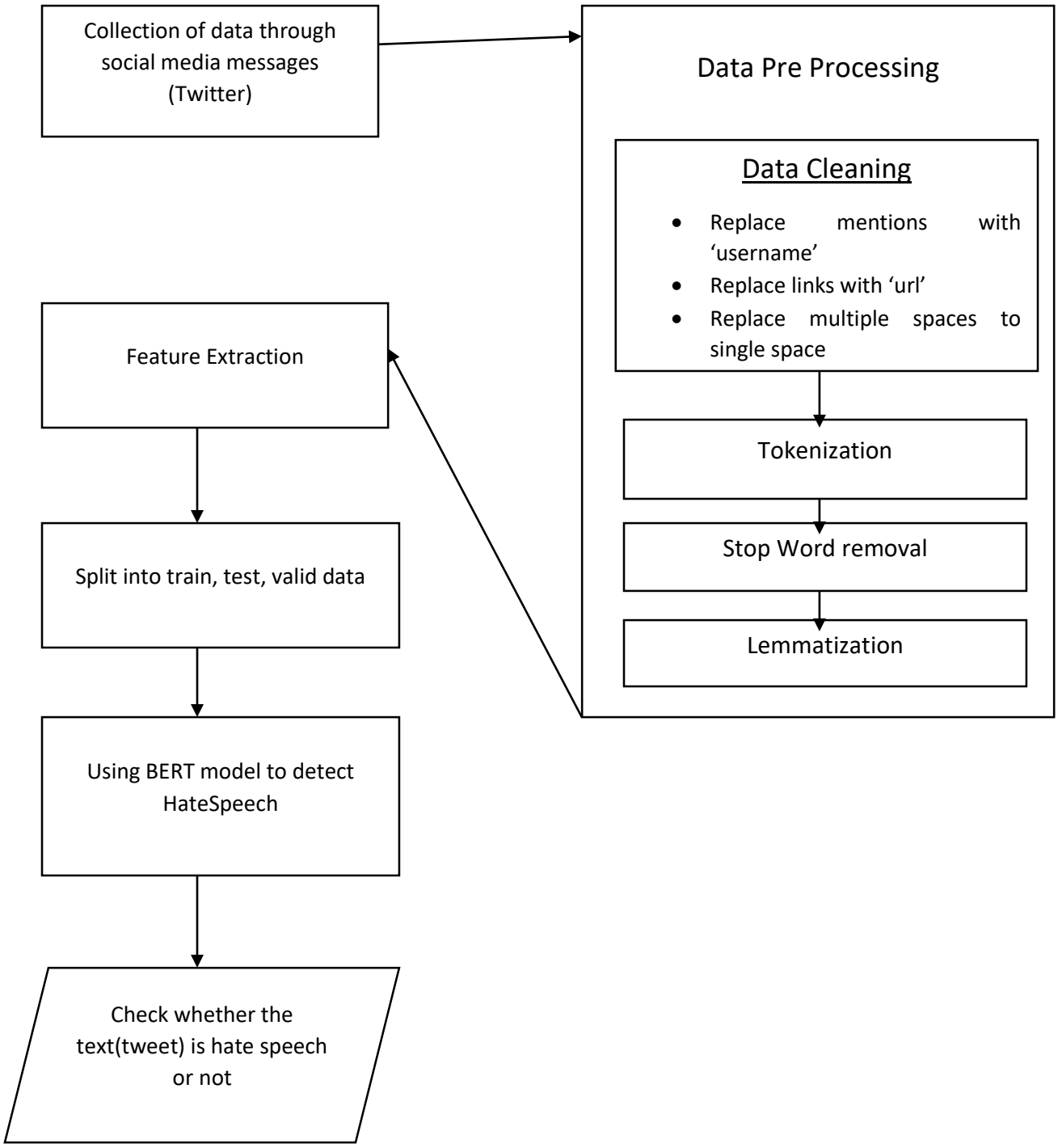


Figure-8: schematic diagram

➢ 7.2. Data Collection

The English, German and Italy dataset used here is collected from the authors of the paper “A Multilingual Evaluation for Online Hate Speech Detection”: MICHELE CORAZZA, STEFANO MENINI ELENA CABRIO, SARA TONELLI, SERENA VILLATA.

English:

We employ the 16,000 English tweets that have been manually labeled as containing hate speech. More specifically, 10,884 tweets are annotated as not containing offensive language, while 1,924 are noted as containing racism and 3,082 as containing sexism. 5,006 tweets would be categorized as positive examples of hate speech if tweets that are both sexist and racist are combined into a single class.

Italian:

We make use of the Twitter dataset made available for the shared HaSpeeDe (Hate Speech Detection) task organized at Evalita 2018, the Italian language evaluation of NLP and speech processing tools. There are 4,000 tweets total in this dataset. "Hateful post" or "not" are the two classes that are taken into account in the annotation.

German:

At Germeval 2018, a workshop featuring a series of shared tasks on German processing, we use the dataset distributed for the shared task on the identification of offensive language. 5,009 German tweets were manually annotated at the message level with the labels "offense" (abusive language, insults, and profane statements) and "other" (i.e., not offensive) in order to identify offensive comments from the set of German tweets (binary classification). More specifically, 3,321 messages are marked as "other," and 1,688 messages are marked as "offense."

Bengali:

Bengali hate speech dataset is collected web pages through web API or previously stored data from local databases. There are total 3418 Bengali tweets available, they are classified as “geopolitical”, “personal”, “abusive”, “religious” and “political”.

➢ 7.3. Data Preprocessing

Here we perform the necessary data preprocessing and cleaning on the collected dataset. It involves several steps, as described below:

- ✓ Data Cleaning
- ✓ Tokenization
- ✓ Stop Word removal
- ✓ Lemmatization

i. Data Cleaning

Data Cleaning is the process of detecting and identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the irrelevant data.

Here in our dataset, the mentions of tweets have been replaced by “username”. Links replaced by keyword “url”. Multiple white spaces are replaced by single white space. Emojis are replaced wuth short text. Also the special characters like @,$,” are removed as these words are not required to analyse the sentiment of the tweet.

ii. Tokenization

Tokenization is the process of breaking down a text document into smaller units, such as individual words or phrases. The phrase or each of these smaller pieces is referred to as a token. Word segmentation is another name for tokenization. Tokenization is the division of a character sequence into tokens, given a character sequence and a specified document unit. Tokens can be either words or numbers. This task is completed by tokenization by identifying word boundaries.

iii. Stop Word removal

Stop words are words that are filtered out before processing natural language data because they frequently refer to a language's most frequently used words. These words shouldn't occupy any room in our dataset. We eliminated the stop words using NLTK and its easily accessible "Stop Word Dictionary" because they weren't helpful for sentiment analysis.

For example,

A text with stop words: “I love to eat biryani”
After stop word removal, the text becomes [“love”, “eat”, “biryani”]

iv. Lemmatization

In linguistics, lemmatization is the process of combining a word's inflected forms into a single unit for analysis. Lemmatization, in computational linguistics, is the algorithmic process of figuring out a word's lemma based on its intended meaning. Understanding the intended function and meaning of each word in a sentence is essential to lemmatization.

For example,

am, are, is -> be

car, cars, car’s, cars’ -> car

➢ 7.4. BERT – word embedding

The BERT base model employs 12 layers of transformer encoders, and each token's output from each layer can be used as a word embedding.

Compared to word embedding techniques like Word2Vec, BERT has an advantage because it creates word representations that are dynamically influenced by the words around them. Under Word2Vec, each word has a fixed representation regardless of the context in which it appears. Consider the following two examples:

“The man went to the bank for money withdrawal.”

“He loves fishing by sitting at the bank of the river.”

Word2Vec would produce the same word embedding for the word "bank" in both sentences, whereas BERT would produce different word embeddings for "bank" in each sentence. The context-informed word embeddings capture additional types of information in addition to obvious differences like syntax, which leads to more accurate feature representations and better model performance.
It is an advanced, unsupervised, bidirectional language representation that was previously trained on a corpus of plain text.

Each output per token from each of the 12 layers of transformer encoders used in the BERT base model can be used as a word embedding. The maximum length corresponds to 512 subword tokens because BERT uses the WordPiece subword tokenizer. Both pre-training and fine-tuning are phases of it.

BERT uses context from both sides of the current word, i.e., the entire sequence rather than just the first few words, making it bidirectional.

Instead of ReLU and ELU, GELU is used as the activation function in BERT.

BERT model uses its own embedding.

➢ 7.5. BERT Model

BERT architecture:

The BERT developers produced two main models:

(a) BERT base model: Transformer block count is 12, Hidden layer size is 768, and Attention heads: 12

(b) BERT Large model: There are 24 transformer blocks, 1024 hidden layers, and 16 attention heads.

Input formatting:

BERT expects input data in a particular format with unique tokens to indicate the start ([CLS]) and end or separation ([SEP]) of sentences. In addition, we must tokenize our text into words that match BERT's vocabulary. BERT requires input ids, a string of integers that link each input token to its corresponding index number in the BERT tokenizer vocabulary, for each tokenized sentence.

OUTPUT:

The following outputs are produced by BertModel based on the configuration (BertConfig) and inputs;

- last_hidden_state -> sequence of hidden states at the model's final layer's output.
- pooler_output -> the first token in the sequence's last layer hidden state
- hidden_states(optional, returned when output_hidden_states=True or config.output_hidden_states=True is passed) -> The outputs from the initial embedding as well as the hidden states of the model at each layer's output.
- attentions (optional, returned when output_attentions=True or config.output_attentions=True is passed) -> The weighted average in the

self-attention heads is computed using attention weights following the attention softmax.

The entire set of the model's hidden states, which is stored in the object hidden_states, can be a little disorienting. This thing has four dimensions, arranged as follows:

1. layer number (13 layer)
2. batch number (number of sentences)
3. word/token number (maximum length of sentences)
4. hidden unit/feature number (768 feature)

Please note that It has 13 layers because the input embeddings make up the first one, and the outputs from each of the 12 layers of BERT make up the remaining 12.

Here I have used the Hugging Face implementation of BERT.

For English, “bert-base-cased” model is used.

Here hate-BERT have been used for non-English languages like Bengali, German, Italian languages.

For Bengali, "l3cube-pune/bengali-bert" BERT model has been used.

For German, “`bert-base-german-cased`” model has been used.

For Italian, “Hate-speech-CNERG/dehatebert-mono-italian” model has been used.

Chapter 8: ALGORITHM

BERT, LSTM and Bi-LSTM these three models have been used to perform hate speech detection. Here only algorithm of BERT model to detect hate speech in social media texts (mainly tweets) has been discussed:

✓ ***Data Preprocessing:***

Input: Datasets of social media tweets
Output: Preprocessed dataset

Step 1: Read the tweets of the dataset

Step-2: Replace the mentions of users with the phrase "username"

Step-3: Replace the emojis with short textual description / vector (for non-English language)

Step-4: Replace the URLs with the phrase "url"

Step-5: Replace multiple white spaces with single white space

Step-6: Preprocessed dataset is ready

✓ ***Feature Extraction:***

Input: Processed dataset
Output: Features extracted moved to the next layer

Step-1: Tweet dataset is separated into specific classes, i.e. some hate and non-hate

Step-2: In case of same languages, hate dataset has been classified into separate classes, for example: in case of English dataset, hate speech has been classified into racism, sexism and none; and in case of Bengali dataset,

hate speech has been classified into geopolitical, political, personal, religious, gender abusive. In German and Italian dataset, dataset has been divided into hate and non-hate data.

Step-3: These classes are stored in separate column of the pandas data frame.

Step-4: Feature extraction done

✓ ***Test-Train-Valid Splitting:***

Input: Tweets along with their classes
Output: Dividing the dataset for training, testing and validation

Step-1: 60% of the dataset has been selected for training, 20% for validation and 20% for testing

Step-2: To make our experiment reproducible, the train_test_split function from scikit-learn has been used to shuffle and split the dataset into 60% and 40%.

Step-3: The remaining 40% was split in half to obtain the validation and test set, respectively.

Step-4: Here the random state = 42 has been used for the random number generator used for shuffling.

Step-5: Finally, the dataset has been splitted properly. Training dataset has been used for training of ML model using this train data, Test set used for testing and validation dataset for checking validation.

✓ ***Training of BERT model using the Train dataset***

Input: Train dataset
Output: Training of the BERT model using the dataset
Step-1: AutoTokenizer is loaded into tokenizer. It helps to automatically retrieve the BERT model given the

provided pre-trained weight. In case of using hateBERT, the specific BERT model of the respected language has been used.

Step-2: This tokenizer can be used to convert data to tokens.

Step-3: CLS, SEP tokens are used in BERT along with the tokenized format of the data. [SEP] is separator token, [CLS] used to represent the start of the sequence.

Step-4: Calculate the vocabulary size, max length and model input names of the model

Step-5: Write a tokenize function as follows:

```
Tokenize_function(train_dataset)
{
    return tokenizer(train_dataset['Tweets'], padding='max_length', truncation=True)
}
```

Step-6: Create a tokenize dataset as tokenized_dataset = dataset.map(tokenize_function, batched=True)

Step-7: Now, keep the 'train' part of the tokenized dataset into train_dataset, 'valid' part of the tokenized dataset into eval_dataset, and 'test' part of the tokenized dataset into test_dataset.

Step-8: Get the train_features from the train_set

Step-9: Set train_set_for_final_model with the train_features and the 'Class' of the train_set

Step-10: Shuffle the train_set_model for the length of train_set along with the batch 8

Step-11: Extract the eval_features from the model_input_names of tokenizer

Step-12: Set val_set_for_final_model with the eval_features and 'class' of the eval dataset i.e. dataset for validation

Step-13: Update val_set_for_final_model for the batch of 8

Step-14: Extract the test_features from the model_input_names of tokenizer

Step-15: Set test_set_for_final_model with the test_features and 'class' of the test dataset i.e. dataset for testing

Step-16: Set test_set_for_final_model for the batch of 8

Step-17: Load model from the pretrained available BERT model, in case of hateBERT, load model with the available hateBERT of the respective language

Step-18: Compile model with the learning rate 5e-5

Step-19: Fit the model with the train_set_for_final_model and validation_data as val_set_for_final_model and specific number of epochs

Step-20: Plot model sparse categorical accuracy and model loss into graph

- ✓ ***Testing and Validation of BERT model using the Test and Val dataset***

Input: Fitted model and test and validation dataset

Output: Accuracy and F1 score

Step-1: Compute test_loss and test_accuracy using test_set_for_final_model and verbose=2

Step-2: Display the test accuracy

Step-3: Store the truth values of test data into truth

Step-4: Calculate the prediction score of the test data (using tokenization) into prediction

Step-5: Now compute the classification_report using truth and prediction

Step-6: We can get the precision, recall, F1-score, support for each of the specified classes from this classification_report

Step-7: We can also get the accuracy from this report

Chapter 9: IMPLEMENTATION

To implement hate speech detection using BERT model, following steps have been followed:

1. Collection of dataset of hate speech and non-hate speech for English and non-English languages (Bengali, German, Italian) to show that it would work language independently
2. Pre-processing of data (Steps have been discussed in the algorithm section)
3. Fine-tuning a BERT model on the dataset
4. Evaluating model on the test dataset

Initially it was done with the help of "BERT-base-cased" fine-tuning model, later the whole process was done for language specific hate-BERT hugging face model.

Chapter 10: RESULT & PERFORMANCE ANALYSIS

Bengali Results:

Train: test: validation=60:20:20 i.e. the total dataset has been divided into training by 60% of dataset, for testing by 20% and for validation checking by 20%.

The Bengali hate speech dataset has been divided into 5 classes abusive, geopolitical, personal, political, religious.

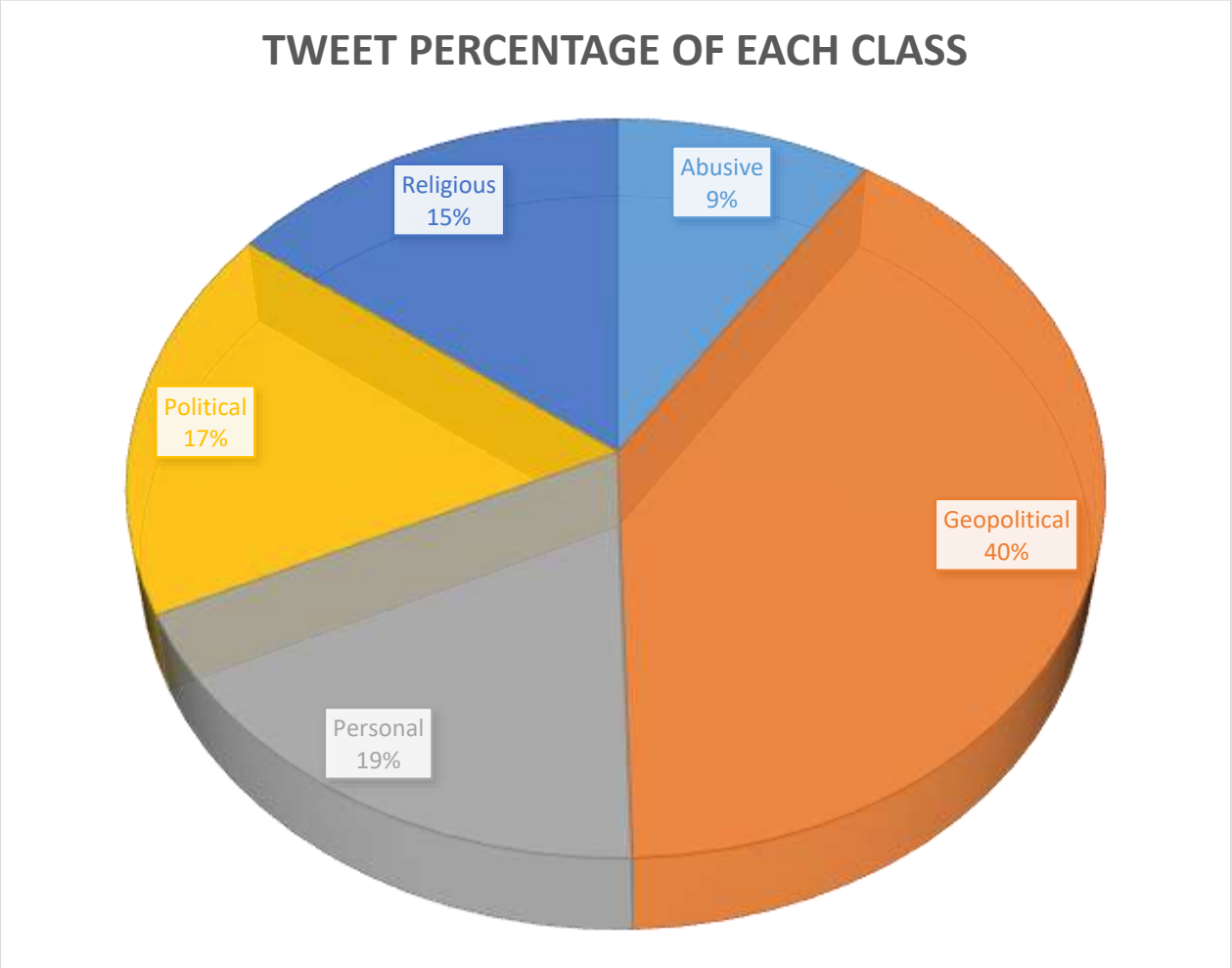


Figure-9: Tweet percentage of each class of Bengali dataset

Table-2:

Epoch	Class	True Positive	True Negative	False Positive	False Negative	Precision	Recall	F1-score of each class	Accuracy
5	***Abusive***	240	2995	107	76	0.69	0.76	0.72	89%
	Geopolitical	1328	1959	80	51	0.94	0.96	0.95	
	Personal	460	2738	51	169	0.90	0.73	0.81	
	Political	566	2744	82	26	0.87	0.96	0.91	
	Religious	457	2869	47	45	0.91	0.91	0.91	
3	***Abusive***	183	18	98	12	0.69	0.58	0.63	88%
	Geopolitical	17	1332	5	10	0.93	0.97	0.95	
	Personal	54	22	480	65	0.80	0.76	0.78	
	Political	6	8	12	564	0.86	0.95	0.90	
	Religious	4	45	5	5	0.94	0.88	0.91	

Note: Accuracy= how many predictions we got right

= (True Positive + False Positive) / (True Positive+ True Negative+ False Positive + False negative)

Precision= True Positive / (True Positive + False Positive)

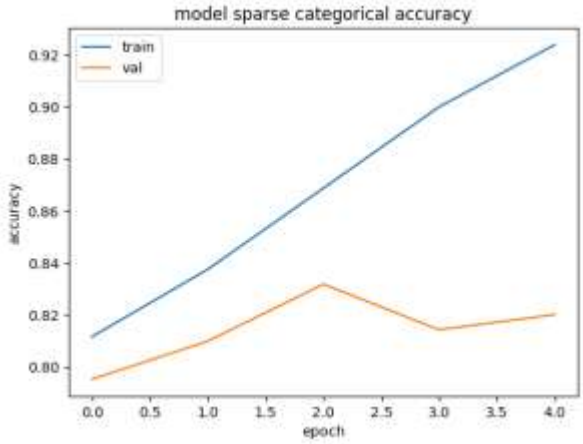
Recall= True positive / (True positive + False negative)

F1 - score = Harmonic mean of precision and recall

= 2* (precision * recall) / (precision + recall)

True positive = number of data for which both the truth value and the predicted value of the model is positive
True negative = number of data for which both the truth value and the predicted value of the model is negative
False positive = number of data for which the truth value is negative but the predicted value of the model is positive
False negative = number of data for which the truth value is positive but the predicted value of the model is negative

For epoch 5:



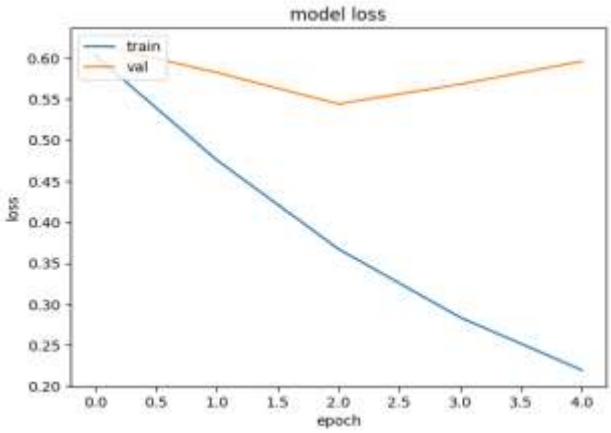
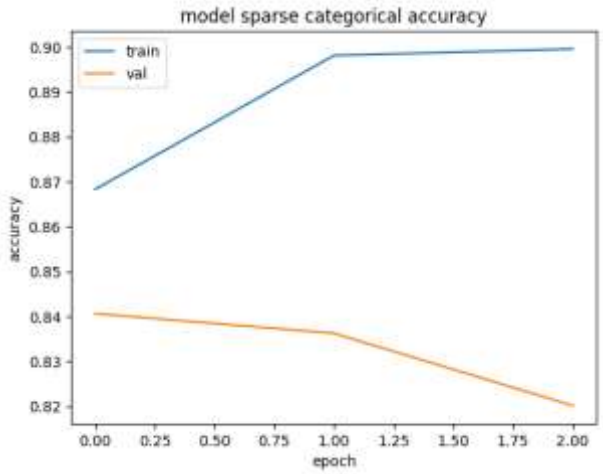


Figure-10: Model sparse accuracy for epoch 3

Figure-11: Model loss for epoch 3

For epoch 3:



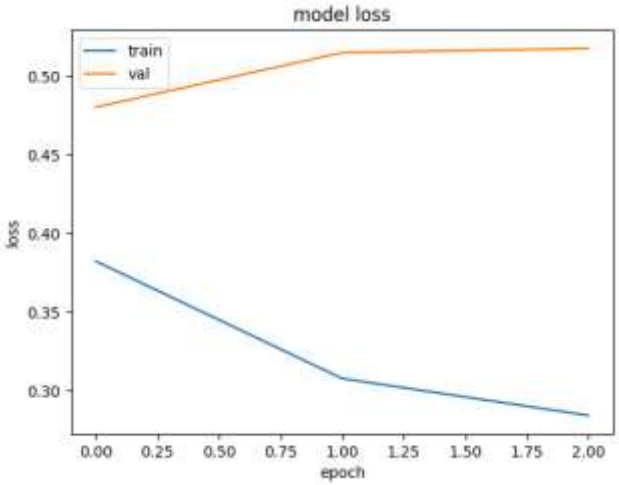


Figure-12: Model sparse accuracy for epoch 5

Figure-13: Model loss for epoch 5

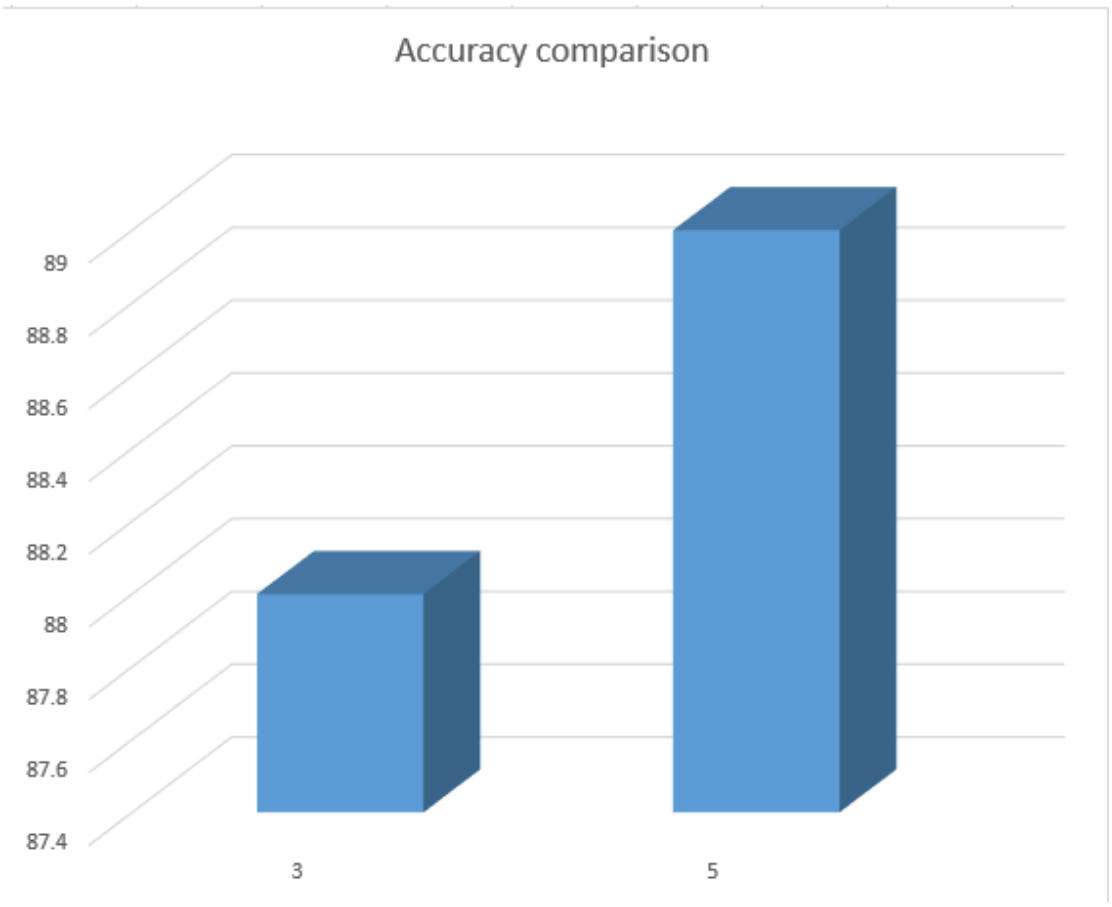


Figure-14: Accuracy comparison of Bengali dataset

English Results:

Train: test: validation=60:20:20 i.e. the total dataset has been divided into training by 60% of dataset, for testing by 20% and for validation checking by 20%.

The English hate speech dataset has been divided into 3 classes none, racism, sexism.

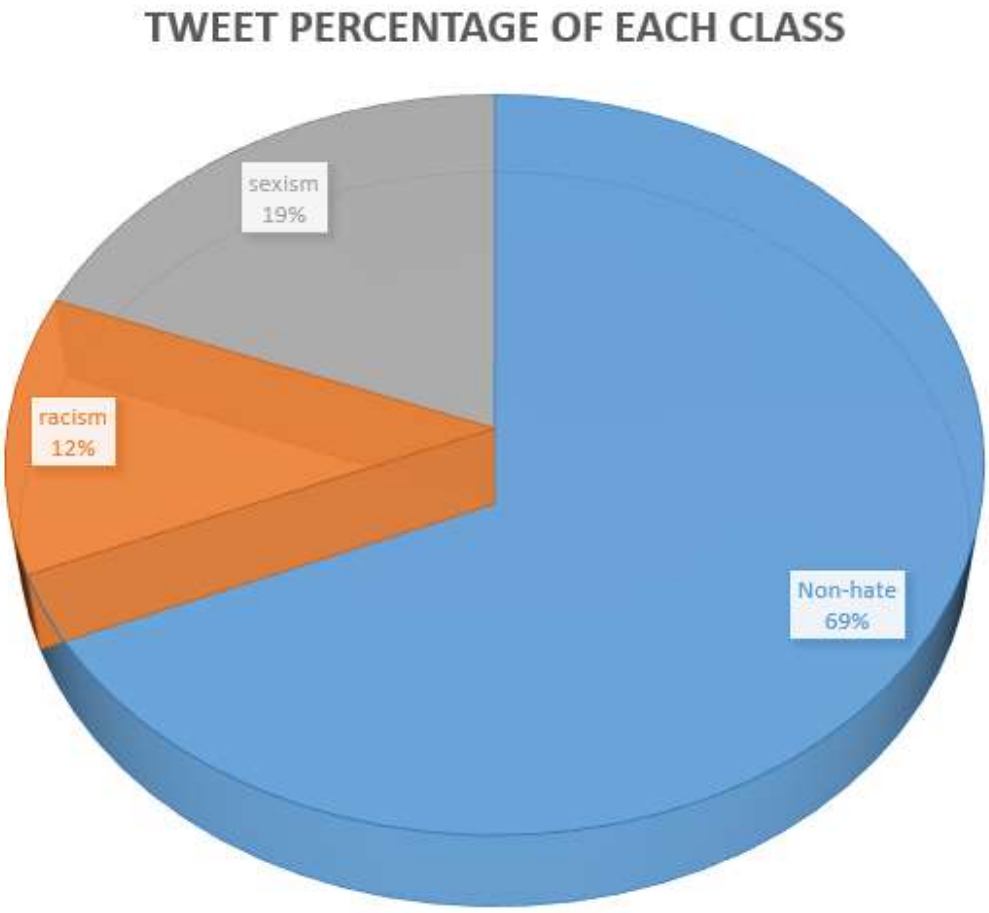


Figure-15: Distribution of different classes of Tweet dataset

Table-3:

Epoch	Class	True Positive	True Negative	False Positive	False Negative	Precision	Recall	F1-score of each class	Accuracy
3	***None***	9862	4472	464	979	0.96	0.91	0.93	91%
	Racism	1695	13580	278	224	0.86	0.88	0.87	
	Sexism	2757	12039	721	260	0.79	0.91	0.85	
5	***None***	10418	4011	925	423	0.92	0.96	0.94	91%
	Racism	1542	13727	131	377	0.92	0.80	0.86	
	Sexism	2455	12454	306	562	0.89	0.81	0.85	
7	***None***	10529	4152	784	312	0.93	0.97	0.95	91%
	Racism	1669	13681	177	250	0.90	0.87	0.89	
	Sexism	2476	12618	142	541	0.95	0.82	0.88	

Total**: 15777**

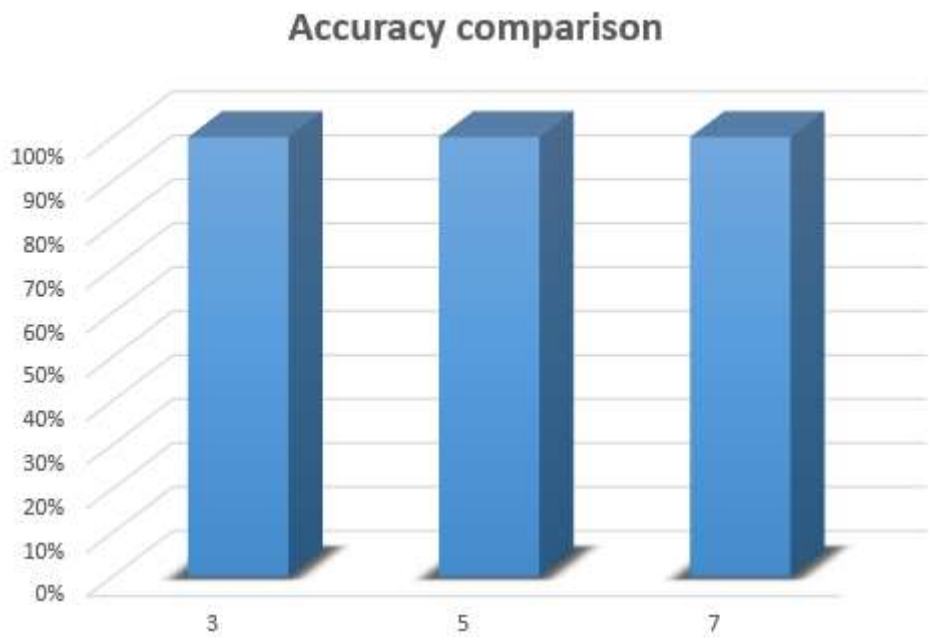


Figure-16: Accuracy comparison of English dataset

German Results:

Train: test: validation=60:20:20 i.e. the total dataset has been divided into training by 60% of dataset, for testing by 20% and for validation checking by 20%.

The German hate speech dataset has been divided into 2 classes hate (abusive) and non-hate.

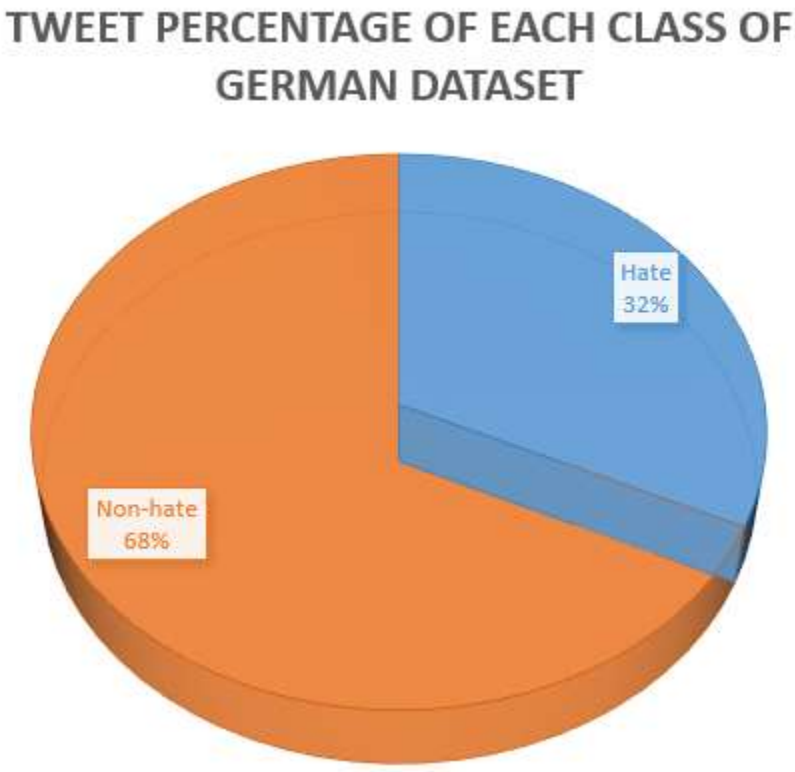


Figure-17: Distribution of different classes of German Tweet dataset

Table-4:

Epoch	Class	True Positive	True Negative	False Positive	False Negative	Precision	Recall	F1-score of each class	Accuracy
3	***Hate***	877	1788	273	93	0.76	0.90	0.83	88%
	Non-Hate	1788	877	93	273	0.95	0.87	0.91	
5	***Hate***	773	1988	73	197	0.91	0.80	0.85	91%
	Non-Hate	1988	773	197	73	0.91	0.96	0.94	
7	***Hate***	797	1957	104	173	0.88	0.82	0.85	91%
	Non-Hate	1957	797	173	104	0.92	0.95	0.93	

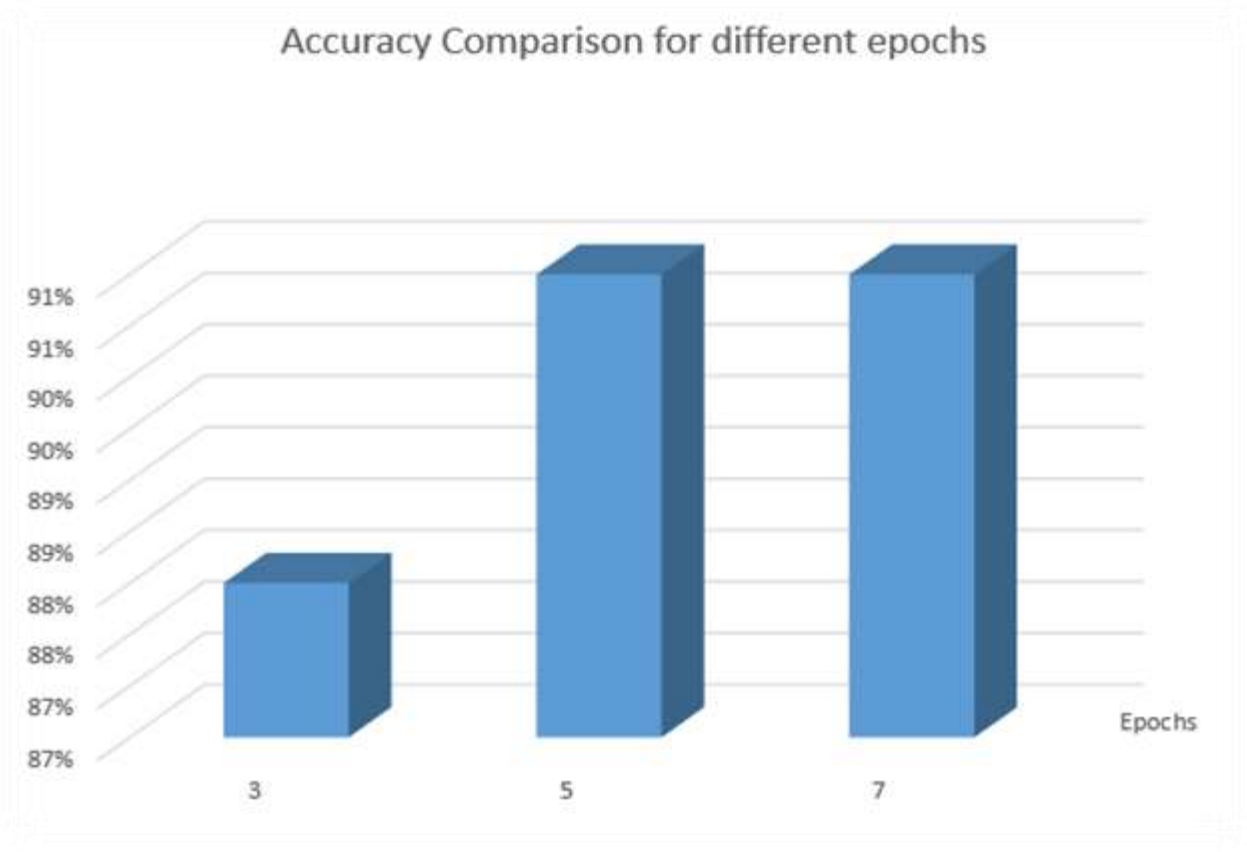


Figure-18: Accuracy comparison of German dataset for different number of Epochs

Italian Results:

Train: test: validation=60:20:20 i.e. the total dataset has been divided into training by 60% of dataset, for testing by 20% and for validation checking by 20%.

The Italian hate speech dataset has been divided into 2 classes hate (abusive) and non-hate.

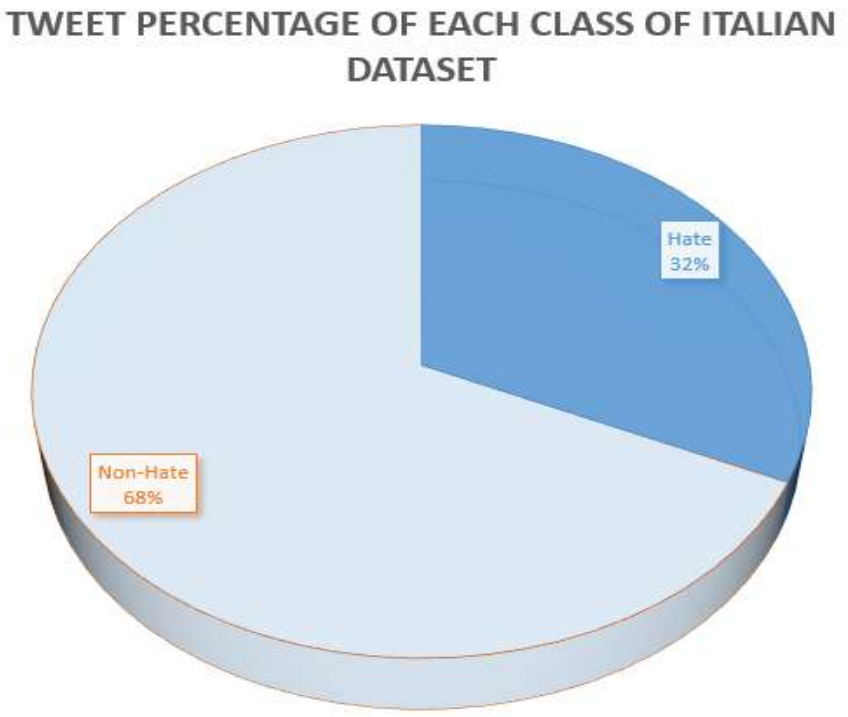


Figure-19: Tweet percentage of each class of Italian dataset

Table-5:

Epoch	Class	True Positive	True Negative	False Positive	False Negative	Precision	Recall	F1-score of each classs	Accuracy
3	*Hate*	228	701	107	164	0.68	0.58	0.63	77%
	Non-Hate	701	228	164	107	0.81	0.86	0.84	

Chapter 11: CONCLUSION & FUTURE SCOPE

This article focuses on the detection of hate speech in social media messages, first identifying an neural network that is very stable and performs well in different languages (i.e. Bengali, English, German, Italian). Here BERT model had been used for performing such detection for its high success rate. We then evaluated the contribution of several commonly used components in this task: the type of embedding (BERT uses its own embedding which uses contextualized embedding to convert words to vectors), the use of additional features (text-based), the role of hashtag normalization, and the role of emoji. A comparative analysis was performed on Bengali, English, Italian, and German Twitter datasets available for hate speech detection (indicating whether they contained hate speech/offensive language). After experimenting this model and recording results, it can be concluded that BERT model is an effective neural network model for hate speech detection.

System Failure

Now, let's discuss some shortcomings/challenges of the system:

1. Among the common error categories found in the surveyed datasets are some examples of implicit abuse. Such messages do not contain offensive language, but rather show that offensive language through sarcasm, jokes, the use of negative stereotypes, or objective statements that imply some form of aggression. For this kind of tweets, the model showing that the tweet is non-hate speech, but in reality it actually a hate speech.

For example, "*And you think that cute young lady can do this alone*?" – the above example is basically a hate speech because it is making fun of someone's capability (example of sarcasm without hate words), but this model cannot recognize this sentence as hate speech as it is not containing any abusive/hate words, so outputs it as non-hate speech – which is wrong.

Also sometimes a sentence may contain abusive words but it is not hate speech at all. But it will result as hate speech sentence.

For example, "*ধুর বাল, এতো ঝামেলা আর ভালো লাগে না।*" (English translation: I don't like so much trouble anymore.)

It is not actually hate speech, because someone is expressing his/her frustration about some happening things which cannot be hate speech but because *of presence of an abusive word "বাল*", it results as hate speech sentence.

2. Sentences with complex syntactic structures (eg, containing multiple negations or interrogative sentences) are common, both in false-positive and false-negative sentences. The same applies to Tweets that contain anaphoric elements that indicate references that may have existed in previous messages, or that require some degree of world knowledge to understand. *In some cases, links to external media fueled the Tweet's disgust. However, this information was not used for classification as the preprocessing step removes the URLs*.

   For example: "*no. You've proven your ignorance here counter to someone who isn't as naive as you. Everyone sees it, but you don't know it.*" – as we can see the above sentence has complex structure and containing multiple negations. This sentence is wrongly classified by the model as "Non-hate" speech but it actually is, because in actual meaning this sentence is encountering someone as dumb or foolish enough to understand what he had done but the model classifies this sentence as non-hate because of its complex structure.

3. Of the false positives, the investigated example is on the English dataset, and similar behavior has been observed for Italian tweets. These datasets are gleaned from keywords associated with potential hate targets such as *women, Roma, and Muslims*, and are enriched with non-offensive tweets, so that the classifier can identify such messages even if they are offensive. We tend to associate references to the target with hate speech, even if it's not relevant. This phenomenon is less evident in the German data, which was actually created differently starting with the list of users.

   Example of false positive is: "*Fine by me. I had few Jewish friends in college. None ever went to a Synagogue*" - The message is classified as hate-speech by the model maybe because of mention of the word

"Jewish". The model is biased towards some words such as "Jewish", that's why it classifies the sentence as "hate" whereas the meaning of the Sentence is very simple- it means that I had some Jewish friends who never went to Synagogue and it's completely okay to me, so it isn't actually hateful.

4. *Data scarcity*: Missing large, labeled hate speech datasets. This makes it difficult to train accurate and reliable BERT models. If we notice the dataset length, we can find that only English dataset has large number of tweets which was very helpful for training the model. Other datasets of three languages had few numbers of tweets which is not so enough to train model.
5. *Imbalanced data*: If we notice the dataset, we find that numbers of hate labelled data is less than numbers of non-hate data. This effected F1-score of the hate class data, i.e. F1-score of non-hate class is more than that of hate class. So if any balanced dataset could be made/found then this process may produce higher F1-score for hate class also.
    a. Bengali dataset has five classes, out of which some tweets are close to each other class (Some tweets are marked as political which can be personal also).
6. Another issue I encountered is limited RAM. For example, my system has RAM limit up to 4GB. That's why I have used Google collab (not google collab pro). But here GPU (type: T14) has some limitations such as it can continue work for maximum 12hrs. So, if it takes 4hrs for model training for only 1 epoch, then we cannot train model by taking more epochs (which happened in case of Italian dataset where each epoch was taking 4hrs for training model, this is the reason record of only epoch 3 has been recorded in the result section)

Any future work to overcome these shortcomings must improve the accuracy of the system to detect hate speech. Also if some properly labelled dataset which is large enough to train the model can be made then BERT model will definitely result more accurately. Another important thing used before training of model is embedding of words i.e. converting words into vectors using BERT embedding. These vectors are then fed to model for further working of model.

I hope this article will be beneficial for future research work in multilingual online hate speech detection of social media texts/messages.

Chapter 12: REFERENCES

[1] Benítez-Andrades, J.A., González-Jiménez, Á., López-Brea, Á., Aveleira-Mata, J., Alija-Pérez, J.M. and García-Ordás, M.T., 2022. Detecting racism and xenophobia using deep learning models on Twitter data: CNN, LSTM and BERT. *PeerJ Computer Science*, *8*, p.e906.

[2] Khan, S., Fazil, M., Sejwal, V.K., Alshara, M.A., Alotaibi, R.M., Kamal, A. and Baig, A.R., 2022. BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection. *Journal of King Saud University-Computer and Information Sciences*, *34*(7), pp.4335-4344.

[3] Rana, A. and Jha, S., 2022. Emotion based hate speech detection using multimodal learning. *arXiv preprint arXiv:2202.06218*

[4] Yin, W. and Zubiaga, A., 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, *7*, p.e598.

[5] Röttger, P., Vidgen, B., Nguyen, D., Waseem, Z., Margetts, H. and Pierrehumbert, J.B., 2020. HateCheck: Functional tests for hate speech detection models. *arXiv preprint arXiv:2012.15606*

[6] Mullah, N.S. and Zainon, W.M.N.W., 2021. Advances in machine learning algorithms for hate speech detection in social media: a review. *IEEE Access*, *9*, pp.88364-88376.

[7] Matamoros-Fernández, A. and Farkas, J., 2021. Racism, hate speech, and social media: A systematic review and critique. *Television & New Media*, *22*(2), pp.205-224

[8] Fanton, M., Bonaldi, H., Tekiroglu, S.S. and Guerini, M., 2021. Human-in-the-loop for data collection: a multi-target counter narrative dataset to fight online hate speech. *arXiv preprint arXiv:2107.08720*

[9] Poletto, F., Basile, V., Sanguinetti, M., Bosco, C. and Patti, V., 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, *55*, pp.477-523.

[10] Corazza, M., Menini, S., Cabrio, E., Tonelli, S. and Villata, S., 2020. A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology (TOIT)*, *20*(2), pp.1-22.

[11] Vashistha, N. and Zubiaga, A., 2020. Online multilingual hate speech detection: experimenting with Hindi and English social media. *Information*, *12*(1), p.5.

[12] Mozafari, M., Farahbakhsh, R. and Crespi, N., 2020. Hate speech detection and racial bias mitigation in social media based on BERT model. *PloS one*, *15*(8), p.e0237861.

[13] Aluru, S.S., Mathew, B., Saha, P. and Mukherjee, A., 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

[14] Abro, S., Shaikh, S., Khand, Z.H., Zafar, A., Khan, S. and Mujtaba, G., 2020. Automatic hate speech detection using machine learning: A comparative study. *International Journal of Advanced Computer Science and Applications*, *11*(8).

[15] Roy, P.K., Tripathy, A.K., Das, T.K. and Gao, X.Z., 2020. A framework for hate speech detection using deep convolutional neural network. *IEEE Access*, *8*, pp.204951-204962.

[16] Alshalan, R. and Al-Khalifa, H., 2020. A deep learning approach for automatic hate speech detection in the saudi twittersphere. *Applied Sciences*, *10*(23), p.8614.

[17] Madukwe, K., Gao, X. and Xue, B., 2020, November. In data we trust: A critical analysis of hate speech detection datasets. In *Proceedings of the Fourth Workshop on Online Abuse and Harms* (pp. 150-161).

[18] Rani, P., Suryawanshi, S., Goswami, K., Chakravarthi, B.R., Fransen, T. and McCrae, J.P., 2020, May. A comparative study of different state-of-the-art hate speech detection methods in Hindi-English code-mixed data. In *Proceedings of the second workshop on trolling, aggression and cyberbullying* (pp. 42-48).

[19] Velioglu, R. and Rose, J., 2020. Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge. *arXiv preprint arXiv:2012.12975*

[20] Das, A., Wahi, J.S. and Li, S., 2020. Detecting hate speech in multi-modal memes. *arXiv preprint arXiv:2012.14891.*

[21] Kiela, D., Firooz, H., Mohan, A., Goswami, V., Singh, A., Ringshia, P. and Testuggine, D., 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in Neural Information Processing Systems*, *33*, pp.2611-2624.

[22] Basile, V., Maria, D.M., Danilo, C. and Passaro, L.C., 2020. EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Sspeech Tools for Italian. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)* (pp. 1-7). CEUR-ws.

[23] i Orts, Ò.G., 2019, June. Multilingual detection of hate speech against immigrants and women in Twitter at SemEval-2019 task 5: Frequency analysis interpolation for hate in speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 460-463).

[24] Ishmam, A.M. and Sharmin, S., 2019, December. Hateful speech detection in public facebook pages for the bengali language. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)* (pp. 555-560). IEEE

[25] Sigurbergsson, G.I. and Derczynski, L., 2019. Offensive language and hate speech detection for Danish. *arXiv preprint arXiv:1908.04531*.

[26] Arango, A., Pérez, J. and Poblete, B., 2019, July. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval* (pp. 45-54)

[27] Mulki, H., Haddad, H., Ali, C.B. and Alshabani, H., 2019, August. L-hsab: A levantine twitter dataset for hate speech and abusive language. In *Proceedings of the third workshop on abusive language online* (pp. 111-118).

[28] De Gibert, O., Perez, N., García-Pablos, A. and Cuadros, M., 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.

[29] Pitsilis, G.K., Ramampiaro, H. and Langseth, H., 2018. Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence*, *48*, pp.4730-4742.

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

[31] Fortuna, P. and Nunes, S., 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, *51*(4), pp.1-30.

[32] Ribeiro, M.H., Calais, P.H., Santos, Y.A., Almeida, V.A. and Meira Jr, W., 2018, June. Characterizing and detecting hateful users on twitter. In *Twelfth international AAAI conference on web and social media*.

[33] Del Vigna12, F., Cimino23, A., Dell'Orletta, F., Petrocchi, M. and Tesconi, M., 2017, January. Hate me, hate me not: Hate speech detection on facebook. In *Proceedings of the first Italian conference on cybersecurity (ITASEC17)* (pp. 86-95).

[34] Joshi, R., 2022. L3Cube-HindBERT and DevBERT: Pre-Trained BERT Transformer models for Devanagari based Hindi and Marathi Languages. arXiv preprint arXiv:2211.11418.

[35] Caselli, T., Basile, V., Mitrović, J. and Granitzer, M., 2020. Hatebert: Retraining bert for abusive language detection in english. arXiv preprint arXiv:2010.12472.