Detection Of Hate Speech in Twitter Space Using LSTM Neural Network

A DISSERTATION

submitted in partial fulfillment of the requirements

for the award of the degree of

Master Of Computer Application

by

RUPAM SAHA

Roll No:-002010503023

Exam Roll:-MCA2360016

Registration No

154231 of **2020- 2021**

Under the supervision of

Dr. Diganta Saha

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY,KOLKATA

188, Raja Subodh Chandra Mallick Rd

Jadavpur, Kolkata, West Bengal 700032.

CERTIFICATE OF RECOMMENDATION

This is to certify that the dissertation titled **Detection Of Hate Speech in Twitter Space Using LSTM Neural Network** was completed by **Rupam Saha,** Roll No:- **002010503023** , Exam Roll:-**MCA2360016** ,Registration Number: **154231** of **2020- 2021**, under the supervision of **Dr. Diganta Saha**, Computer Science and Engineering Department , Jadavpur University.

The findings of the research detailed in this project work have not been incorporated into any other work submitted for the purpose of earning a degree at any other academic institution.

Dr. Diganta Saha
Department of CSE
Jadavpur University

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature Of Dean , FET
Prof. Ardhendu Ghoshal
Dean , FET
Jadavpur University

Signature of HOD
Dr . Nandini Mukhopadhay
Head Of The Department
Computer Science & Engineering Dept
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the dissertation entitled **Detection Of Hate Speech in Twitter Space Using LSTM Neural Network** is a bona fide record of work carried out by **Rupam Saha,** Roll No:- **002010503023** , Exam Roll :- **MCA2360016**, Registration No :- **154231** of **2020- 2021** in partial fulfillment of requirements for the award of the degree of the **Master of Computer Application** during the period of January, 2023 to May,2023 . It is understood that by this approval that the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the project work only for the purpose for which it has been submitted .

Signature of Examiner

Date :-

Signature of Supervisor

Date : -

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* 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 dissertation.

I would also like to thank Mr. *Zeerak Talat, Mrs. Sara Tonelli,* Mrs. Melanie Siegel and Mr. Rezaul Karim 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 *Somashree Nandy* and *Srijan Das* for providing me with regular encouragement and mental support throughout this curriculum.

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 this far.

RUPAM SAHA
MASTER OF COMPUTER APPLICATION
COMPUTER SCIENCE & ENGINEERING
ROLL NO : 002010503023
EXAM ROLL : MCA2360016
REGISTRATION NO : 154231 of 2020-21
JADAVPUR UNIVERSITY

DECLARATION

I certify that,

(a) The work **Detection Of Hate Speech in Twitter Space Using LSTM Neural Network** contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

--------------------

Signature

RUPAM SAHA
MASTER OF COMPUTER APPLICATION
COMPUTER SCIENCE & ENGINEERING
ROLL NO : 002010503023
EXAM ROLL : MCA2360016
REGISTRATION NO : 154231 of 2020-21
JADAVPUR UNIVERSITY

ABSTRACT

With the exponential increase in internet usage in last decade, the way of communication has become more digitalized. This has driven to numerous positive outcomes . At the same time, it has brought many negatives too. The volume of destructive substance online, such as hate discourse, is huge. This intrigued within the scholastic community to investigate an automated tool for discovering the hate speech.

It become very necessary to find out an automated technique to put a check on these hateful contents. In this project work I have proposed a deep learning model that helps to find out the hateful contents automatically .

In this work , I tried out my experiment over four different languages . English [35] dataset contains 15,777 tweets classified over three different classes such as Non-Hate , Racism and Sexism .German[38] dataset has 3031 tweets in german language classified over two different classes Non – Hate and Hate . Italian[37] dataset has 3000 tweets in Italian classified over two different classes Non – Hate and Hate . Bengali[36] dataset has 3419 tweets in Bengali and classified over five different classes Geopolitical , political , personal , religious and gender abusive .

My proposed method is based on RNN based LSTM deep neural network along with the FASTTEXT word embedding model .

The best result for *English dataset is obtained by FASTTEXT+LSTM* method with an accuracy of **0.81825 .** Also we have get a better result for Bengali dataset by *FASTTEXT+LSTM* method with an accuracy of **0.61988** comparing it to *WITHOUT FASTTEXT+LSTM* method **.**

But for the *Italian* and *German* dataset we get the best result using *WITHOUT FASTTEXT+LSTM* method . In case of *Italian* dataset we get the accuracy of **0.78** and in case of *German* dataset we get the accuracy of **0.70**.

Keywords : *Hate Speech , Twitter , NLP , Word Embedding , LSTM network*

Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

NLP : NATURAL LANGUAGE PROCESSING

CBOW : CONTINIOUS BAG OF WORDS

LSTM : LONG SHORT TERM MEMORY

TP : TRUE POSITIVE

TN : TRUE NEGATIVE

FP : FALSE POSITIVE

FN : FALSE NEGATIVE

RNN : RECURRENT NEURAL NETWORK

URL : UNIFORM RESOURCE LOCATOR

INTRODUCTION

With the increase of number of users in social-media platform , there is an increase of amount of hateful comments in these platform .The popularity of opinion-rich online resources such as review forums and social media sites encourages users to share their thoughts worldwide in real time . These can be directed towards any individuals or communities to express their opposition. For this reason, finding discrimination is very important for legislators and social media platforms to prevent unnecessary events.

After going through previous research works in this topic thoroughly, I had chosen *A Multilingual Evaluation for Online Hate Speech Detection* [10] and *DeepHateExplainer: Explainable Hate Speech Detection in Under-resourced Bengali Language* [35] as my baseline papers. I had tried to check the accuracy of their used datasets using LSTM model with and without *FASTTEXT* embedding.

In this work , I had tried out my experiment over four different languages . English dataset contains 15,777 tweets classified over three different classes such as Non-Hate , Racism and Sexism . German dataset has 3031 tweets in german language classified over two different classes Non – Hate and Hate . Italian dataset has 3000 tweets in Italian classified over two different classes Non – Hate and Hate . Bengali [35] dataset has 3419 tweets in Bengali and classified over five different classes Geopolitical , political , personal , religious and gender abusive .

The best result for *English dataset was obtained by FASTTEXT+LSTM* method with an accuracy of **0.81825 .** Also we had got a better result for Bengali dataset by *FASTTEXT+LSTM* method with an accuracy of **0.61988** comparing it to *WITHOUT FASTTEXT+LSTM* method **.**

But for the *Italian* and *German* dataset we get the best result using *WITHOUT FASTTEXT+LSTM* method . In case of *Italian* dataset we get the accuracy of **0.78** and in case of *German* dataset we get the accuracy of **0.70** .

This article can be useful in future for detecting hateful comments in social media as well as creating new type of architecture that can be useful in finding hate speech more precisely.

CHAPTER 2: *RELATED WORKS*

This section contained the state of art about previous research work of hate speech recognition.

Previous Research Works:

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 Speech Detection [10]	Corazza, M., Menini, S., Cabrio, E., Tonelli, S. and Villata, S.	*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, Italian, and German. It has used FastText embedding and LSTM model.	Max F1 score of English 0.823, of Italian 0.805, and German 0.758
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	Aluru, S.S., Mathew, B.,	*arXiv preprint arXiv:2004.06465*.	The paper proposes a framework for multilingual hate speech detection	Accuracy using CNN-BiLSTM is 87.0%, using BERT 91.0%, using DistilBERT is

	Hate Speech Detection [13]	Saha, P. and Mukherjee, A.		using deep learning models. The framework is evaluated on a dataset of tweets in 9 languages, and it achieves state-of-the-art results.	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	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.

	Twittersphere [16]				
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:	Kiela, D., Firooz, H., Mohan, A.,	*Advances in Neural Information*	The Hateful Memes Challenge is a benchmark for	It achieves accuracy of 59.3% using Unimodal BERT, 64.73% by

	Detecting Hate Speech in Multimodal Memes [21]	Goswami, V., Singh, A., Ringshia, P. and Testuggine, D.	*Processing Systems*, *33*, pp.2611-2624.	detecting hate speech in multimodal memes. It is constructed such that unimodal models struggle and only multimodal models can succeed	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,	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	It achieved accuracy of 78.3% using SVM, and 79.6% using LSTM model.

		M. and Tesconi, M.		the need to balance accuracy with fairness.	

Conclusion :

This chapter provided a detailed discussion about the previous research work done in this topic . And also helped to find a proper methodology that was more accurate than the previous methodologies .

From this chapter we found out that mostly used methodologies were LSTM , BiLSTM ,BERT and GRU based methods.

CHAPTER 3: *DATA SETS*

We took four different dataset to train and test our deep learning model . Here we tested four different languages and out of them three languages are low resourced language except English. Here we had used four different languages *English , Bengali , Italian and German* . The detailed description about them listed as below.

English Dataset

The English dataset was taken from the paper **Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter** [35] .

It contains 15,777 tweets and classified over three different classes. The distribution of tweets among these three classes is as followed.

Label	Count
Non Hate	10841
Racism	3017
Sexism	1919

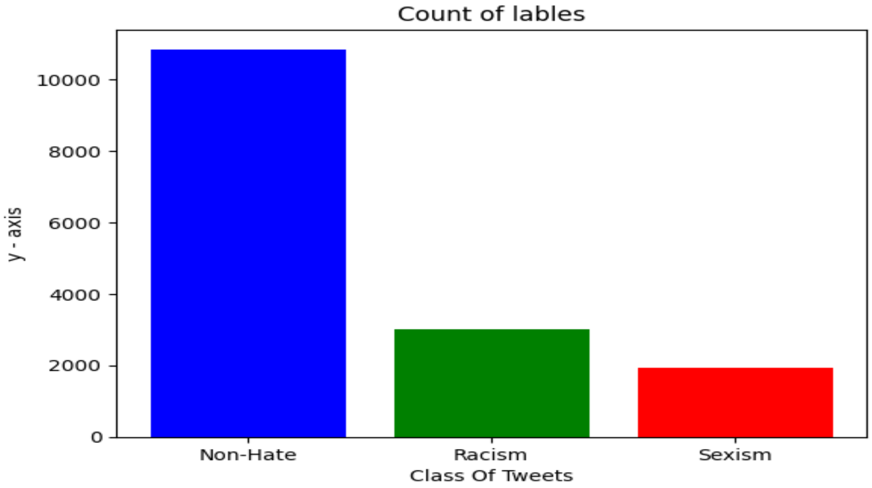


***Figure 1 :** Distribution of English Tweets*

Bengali Dataset

The Bengali dataset was taken from the paper **DeepHateExplainer: Explainable Hate Speech Detection in Under-resourced Bengali Language** [36].

It contains 3419 tweets and classified over five different classes. The distribution of tweets among these five classes was as followed.

Label	**Count**
Geopolitical	1379
Personal	629
Political	592
Religious	502
Abusive	316

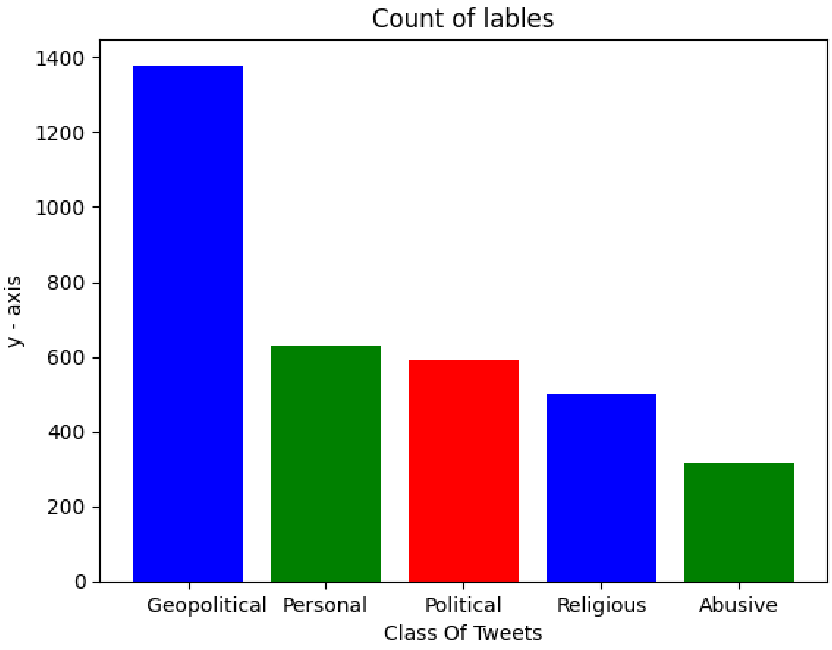


Figure 2 : Distribution of Bengali Tweets

Italian Dataset

The Italian dataset was taken from the paper **Overview of the EVALITA 2018 Hate Speech Detection Task** [37] .
It contains 3000 tweets and classified over two different classes. The distribution of tweets among these two classes is as followed.

Label	Count
Non Hate	2028
Hate	972

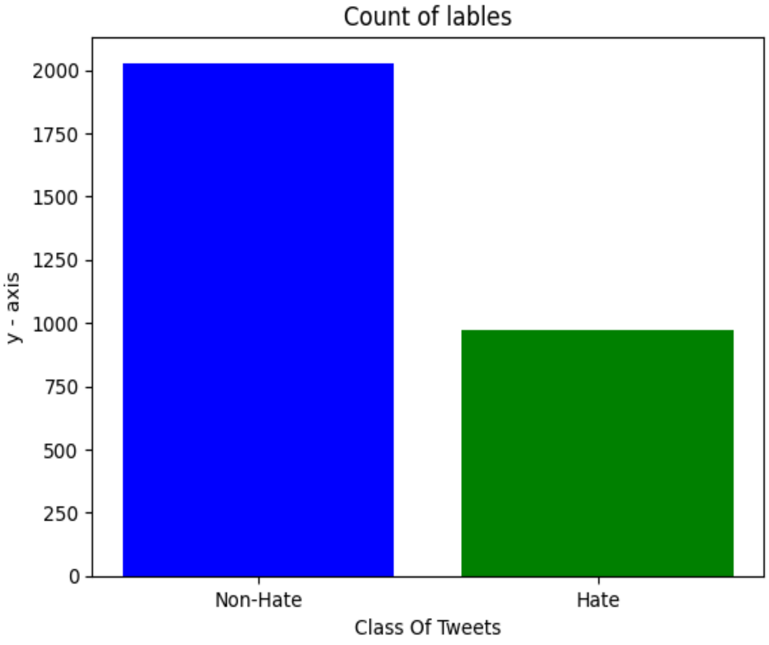


Figure 3: Distribution of Italian Tweets

German Dataset

The German dataset is taken from the paper **Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language** [38] .

It contains 3031 tweets and classified over two different classes. The distribution of tweets among these two classes is as followed.

Label	Count
Non Hate	2061
Hate	970

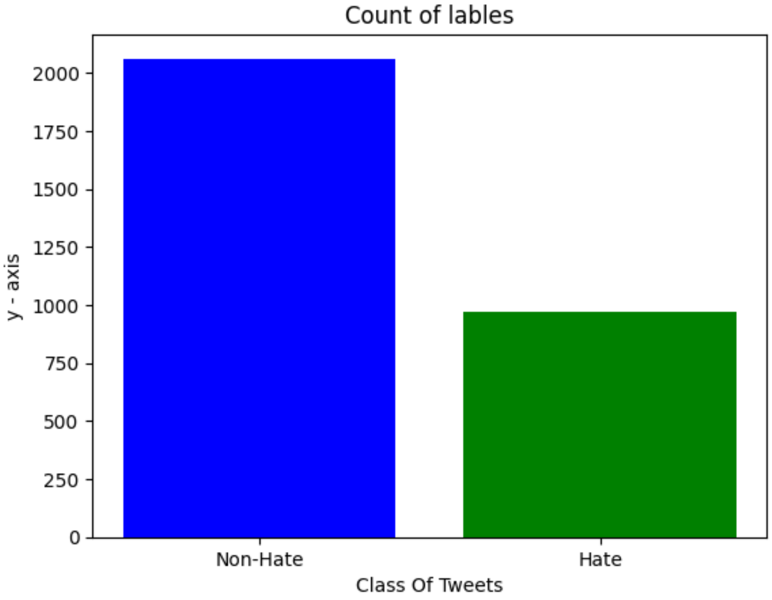


***Figure 4:** Distribution of German Tweets*

➢ Challenges with dataset

As far we have mentioned the properties of our four different datasets . But working with these three datasets can poses certain challenges .

- ✓ Italian , German and Bengali datasets have nearly 3000 plus tweets which is not a satisfactory quantity of data to train a model . Accuracy of model may decrease due to this .
- ✓ Bengali dataset has 3419 tweets which is divided into five sub classes . Some classes are very close to each other in term of semantic meaning . For example it is very hard to differentiate between Geopolitical and Political class . This may lead to a decrease in accuracy of our model .
- ✓ Labels associated with tweets are in string format . To feed our dataset to the model we have to change them into their corresponding integer representation.
- ✓ As all four dataset are collected from the Twitter , most of the tweets contain URLs , username and emojis . URLs and username don't add up any semantic meaning to the tweets . These may lead to the lower accuracy of our model .

Conclusion :

Here we gave a detailed discussion about our four different datasets. Now in next chapter we discussed about how to overcome these difficulties using our proposed methodology.

CHAPTER 4 : *PROBLEM STATEMENT*

Detection Of Hate Speech in Twitter Space Using LSTM Neural Network

MATHEMATICAL NOTATION :

- H = f(T, L)

  Where

  - T was an input in string format . This is a sentence which we tried to predict whether it was a Hateful comment or a Non Hateful one .
  - L was our LSTM neural network model which we trained to predict the class of Input T .
  - f is a function that took a string input T and LSTM neural network model L and showed the class of input T .

We trained our LSTM neural network model L using both hate and non hate tweets and then we tried to predict the class of the input tweet T.

The model L implemented as a neural network model that takes the word embeddings of the input T as input and outputs the most probable class of that input .

The function f trained using a supervised learning method , where the training data contained a set of messages that was already labeled with either hate or non-hate speech. The function trained to predict the correct class of input T using LSTM model .

We could also attach different type of word embedding model such as *Fasttex*t with model M .Respective embedding model trained on a particular language could also be helpful in increasing the accuracy of our model M for a particular language’s tweet .

CHAPTER 5 : *METHODOLOGY*

In this section , I discussed my proposed methodology using LSTM network with and without FASTTEXT word embedding . This section involved a detailed description of methods that we had used . We discussed the components of methodology here and in next chapter we discussed how did we use them and combined with one another.

❖ Schematic Description Of Model

In this section we discussed about the detailed methods of our work and discussed about how did we converged towards our final result .

We also described how do we preprocessed the available dataset to make a more accurate training dataset for our model and helped us to get a nearly perfect result.

The impact of a project depended on the results from the experiment results, which could be evaluatd as a function of the use of appropriate inputs and technologies used on the input data. We discussed various methods that were experimented previously on dataset collected from social media in related work section .

Here , we used LSTM deep neural network with *fasttext* embedding and without *fasttext* embedding and recorded our results .

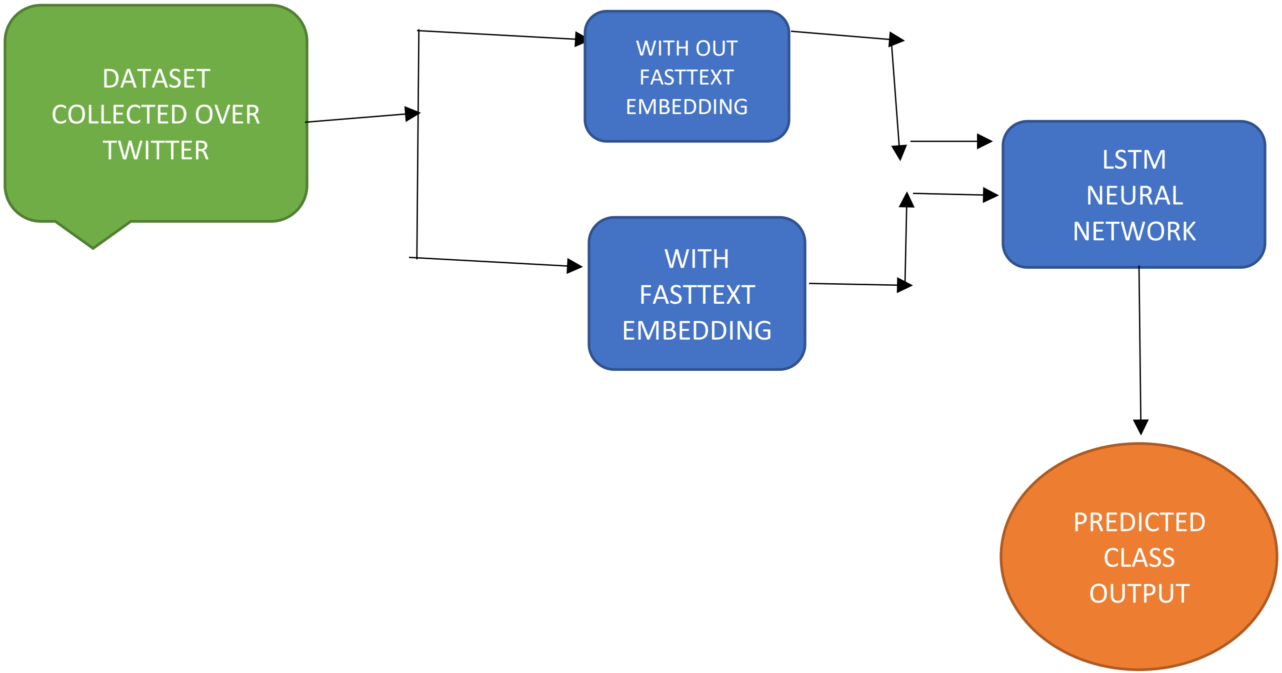
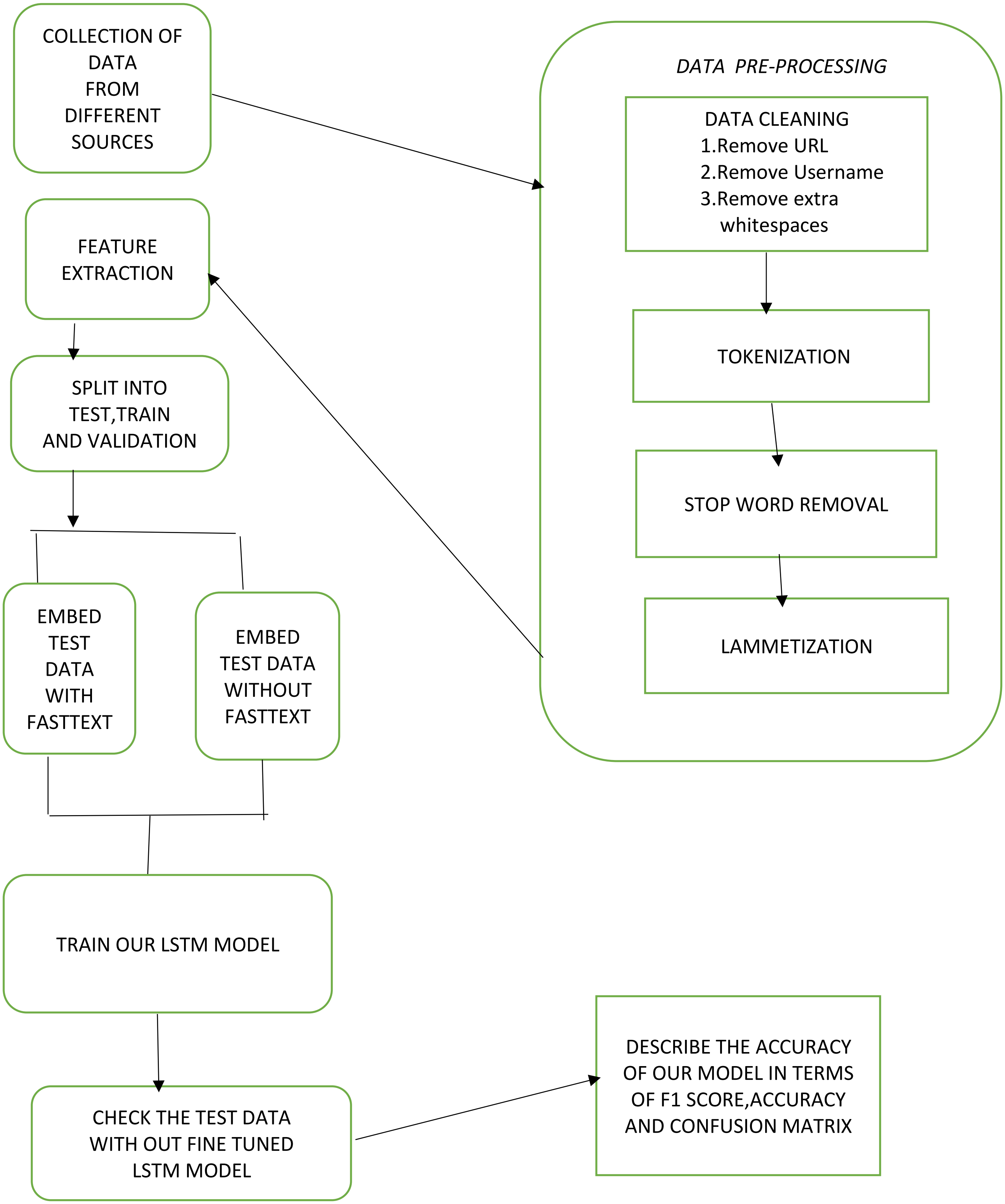


Figure 5 *: Schematic Description Of The Model*

- **Flow Chart Diagram Of Model**


COLLECTION OF DATA FROM DIFFERENT SOURCES
DATA PRE-PROCESSING
DATA CLEANING
1.Remove URL
2.Remove Username
3.Remove extra whitespaces
TOKENIZATION
STOP WORD REMOVAL
LAMMETIZATION
FEATURE EXTRACTION
SPLIT INTO TEST,TRAIN AND VALIDATION
EMBED TEST DATA WITH FASTTEXT
EMBED TEST DATA WITHOUT FASTTEXT
TRAIN OUR LSTM MODEL
CHECK THE TEST DATA WITH OUT FINE TUNED LSTM MODEL
DESCRIBE THE ACCURACY OF OUR MODEL IN TERMS OF F1 SCORE,ACCURACY AND CONFUSION MATRIX

❖ FastText Embedding

FastText [38] is a library as well as a method for word embedding . It is also used for text classification .This is created by Facebook's AI Research lab . Facebook make available pretrained version of 294 different languages model .

This word embedding technique is based upon *Mikolov et al.*'s **Efficient Estimation of Word Representations in Vector Space** [39] paper , which introduced the concept of word vectors by two main methods : ***Skip–Gram*** and ***Continuous Bag of Words (CBOW).*** After that many word embedding methods have created based upon this . Fasttext word embedding technique is also not an exemption from it .

Instead of learning directly like other word embedding technique (W2V) , this method uses the n-gram representation of words.

For example , take the word *"House" with n = 3.* The fasttext representation of this word is <Ho,Hou,ous ,use,se> , where the angular brackets use to denote the end and beginning of a word .

This help to capture the meaning of shorter words and allow the embeddings to understand the suffix and prefix of a word . After representing a word in character n-grams , skip gram model to learn the word embeddings .

Apart from this , FASTTEXT works really well on rare words . Even if a word is not seen in training the n-gram representation helps to get the embedding very well .

Now connecting it to our Hate Speech Detection task, FASTTEXT model is very helpful . We studied the dataset and saw that many users didn't use the hate words directly. Rather they used in short form or they had intentionally hide that word with some extra alphabets attached to it. But the n-gram representation was a game changer here and helped to learn those hate words properly.

So I added FASTTEXT embedding to my model .

❖ LSTM Deep Neural Network

Long Short Term Memory (LSTM) is a type of artificial neural network used in artificial intelligence and deep learning. Unlike standard feedforward neural networks, LSTMs have feedback loops. Such neural networks (RNNs) can process entire data sequences (like speech or video), not just data points (like images). This feature makes LSTM networks well suited for processing and predicting data. For example, LSTMs are suitable for tasks such as clustering, text recognition, speech recognition, machine translation, voice recognition, robot control, game video, and therapy.

A typical LSTM cell consists of a **cell**, an **input gate**, an **output gate** , and a **memory gate**. Cells remember values at arbitrary intervals, and three gates control the flow of data into and out of the cell. The memory gate determines what data to discard from the previous state, giving it a value between **0** and **1** (compared to the current input). A value of 1 means keeping the data, a value of 0 (null) means discarding it. The input port uses the same system as the memory port to determine what new data is currently stored. The output port controls the output of the current state by providing a data value between 0 and 1, including the previous and current state. Optionally extracting relevant information from the current state allows LSTM networks to manage long-term dependencies for prediction of current and future steps.

Now connecting it to our Hate Speech Detection task , there were several advantages of using LSTM network over other deep neural network .
LSTM is designed in such a way that it can easily learn the sequential data more accurately and their long- term dependencies more precisely . LSTM particularly have a long term “memory” that can easily predict the context more precisely over the other neural network and by this overcoming the long term dependency problem very easily .

So due to these mentioned advantages , we chose LSTM network as our main experimental artificial neural network .

Conclusion :

Here we discussed our methodology in details. In next chapter, we discussed how we applied our methodology using algorithm .

CHAPTER 6 : *ALGORITHM*

In this experiment we tried out the RNN based LSTM Neural Network for out datasets. We tried out the experiment four different languages . Four different languages are 1. English 2. Italian 3. German 4. Bengali . Our dataset consisted of hate comments which were mainly collected from twitter .

Out of 4 datasets in 2 datasets (Italian , German) tweets were labeled with either hate or non hate .Rest two datasets(English , Bengali) are labeled with more than two different labels . Here tweets were classified into more detailed subclass .

So for 2 dataset we did binary classification and for rest of 2 dataset we did multiclass classification .

Now we used the LSTM neural network into two different ways and recorded the result .
We had done our experiment 1.Without fasttext embedding
2.With fasttext embedding .
We discussed about them separately .

❖ Without FASTTEXT Embedding

This algorithm mainly divided into six parts . These parts were as followed :

1.Data Preprocessing

2.Feature Extraction

3.Test Train Validation split

4.Define the model

5. Fit the model

6.Check the accuracy of the model

1. *Data Preprocessing*

Input -> .csv file containing of raw tweets along with their respective labels.
Output -> .csv file containing of cleaned tweets along with their respective labels .

Step 1: Read the tweets of the dataset.
Step-2: Deleted the username starting with @.
Step-3: Deleted the url from each single tweet.
Step-4: Replaced multiple white spaces with single white space.
Step-5: Preprocessed dataset was ready.

2. *Feature Extraction*

Input -> Cleaned Dataset
Output -> Feature extracted and moved for test train split

Step 1: Initialized the maximum sequence length to be 250.

Step 2: Defined the embedding dimension to be 100.

Step 3: From keras.preprocessing we imported Tokenizer and applied it on the column

of 'Cleaned Tweets'.

Step 4: Now we imported texts_to_sequences from keras.preprocessing and applied it on the column of 'Cleaned Tweets' and saved it in a variable.

Step 5: Now we imported pad_sequence from keras.preprocessing and applied it over the variable that we got from step 4 and passed the parameter value of padding_length that we had initialized in step 1.

Step 6 : We one hot encoded the column that was associated with labels of tweets.

Step 7: Feature extraction was completed .

3. *Test Train Validation split*

Input -> Feature extracted dataset from part 2
Output-> Feature extracted test,train and validation data

Step 1: We had done the test,train split in 60:40 ratio over both 'Cleaned Tweets' and 'Label' and saved them in respective variable.

Step 2: Test,Train and Validation split was complete.

4.*Define the model*

Input ->None
Output-> A sequential model with LSTM layer

Step 1: We imported Sequential() from keras and saved it a variable namaed model.

Step 2: We added an Embedding layer at the first with parameters MAX_NB_WORDS, EMBEDDING_DIM that we had defined in part-2 .

Step 3: We had then added a special drop out layer with rate = 0.2 to reduce overfitting.

Step 4: Now we added a LSTM layer with 100 memory units with dropout rate of 0.2 for the input layer, and recurrent dropout rate = 0.2. This is to learn the to learn the temporal dependencies between words in the sequences.

Step 5: Now we added a dense() layer with 'n' output units and the activation function as 'softmax' or 'sigmoid'(according to binary or multiclass classification) which gave the probability distribution over 'n' classes .

Step 6: Now our model was ready to compile.

Step 7: Now we compiled the model with categorical cross-entropy loss or binary cross-entropy, Adam optimizer, and accuracy metric for evaluation during training.

5.Fit the model

Input ->Pre-processed dataset from part 3 and compiled model from
part 4 .
Output->Training of LSTM model using the preprocessed dataset and
a trained model .

Step 1: We had fitted our model with the preprocessed dataset and predefined Epoch number and batch size as our choice .

Step 2: Made the validation split of 50 % over test data.

Step 3: Our Model was trained.

6. Check the accuracy of the model

Input ->Trained model from part 5 and test dataset from part 3

Output ->Overall Accuracy score of the model , F1 score of each classes and confusion matrix from each classes.

Step 1: Using the trained model we predicted the labels of each tweet from test data set and stored them in a variable.

Step 2: From sklearn.metrics we had imported the predefined obejects and using the output from step 1 and test data from part 3 we had got our Overall accuracy , F1 score and confusion matrix.

Step 3: Recorded accuracy, F1 score and confusion matrix in a word document.

Step 4 : Completed .

Following these mentioned parts we have procceded through the experiment and saved our experiment results.

- **With FASTTEXT Embedding**

This algorithm is mainly divided into six parts . These parts are as followed :

1.Data Preprocessing

2.Load fasttext model

3.Feature Extraction

4.Test,Train,Validation Split

5.Define the model

6. Fit the model

7. Check the accuracy of the model

1.Data Preprocessing

Input -> .csv file containing of raw tweets along with their respective labels.

Output -> .csv file containing of cleaned tweets along with their respective labels.

Step 1: Read the tweets of the dataset.

Step-2: Deleted the username starting with @.

Step-3: Deleted the url from each single tweet.

Step-4: Replaced multiple white spaces with single white space.

Step-5: Preprocessed dataset is ready.

2. Load fasttext model

Input -> None

Output-> Load the pretrained word vectors from fasttext Model in a word dictionary.

Step 1 : Downloaded the pretrained word vector file from fasttext file.

Step 2: Unziped the downloaded file and extract the word vector file.

Step 3: Now we initialized an empty dictionary that would be used to store the word embeddings.

Step 4: Now we opened the vector file and sets the encoding to 'utf-8'. This file contained pre-trained word embeddings in a text format where each line represented a word and its corresponding vector values.

Step 5: Now we saved each word and associated word vector in the dictionary.

3.Feature Extraction

Input -> Cleaned Dataset

Output -> Feature extracted and moved for test train split and embedding input to model.

This contains two parts .

In first part we worked with the raw text tweets and tokenize them. In next part we created an embedding matrix for these words .

Step 1: From keras.preprocessing we imported Tokenizer and applied it on the column of ‘Cleaned Tweets’ and saved it in a variable.

Step 2: Now we imported texts_to_sequences from keras.preprocessing and applied it on the column of ‘Cleaned Tweets’ and saved it in a variable.

Step 3: Now we imported pad_sequence from keras.preprocessing and applied it over the variable that we got from step 4 and passed the parameter value of padding_length that we had initialized in step 1.

Step 4 : We one hot encoded the column that was associated with labels of tweets.

Step 5: Feature extraction completed.

Now we move to the next part .

Step 1 : Initialized the the embedding dimension.

Step 2: Created an empty embedding matrix.

Step 3: For each word in the cleaned text dictionary we checked whether it existed in the pre-trained **embeddings_index** dictionary. If it existed, it added the corresponding embedding vector to the embedding matrix .

Step 4: If it did not exist, the row for that word was left as all zeros in the embedding matrix. The words that were not found in the pre-trained embeddings were added to the **words_not_found** list.

Step 5: Embedding matrix was ready and we passed it to the model .

Now , our feature extraction was ready **.**

4. *Test Train Validation split*

Input -> Feature extracted dataset from part 2

Output-> Feature extracted test,train and validation data

Step 1: We had done the test,train split in 60:40 ratio over both 'Cleaned Tweets' and 'Label' and saved them in respective variable.

Step 2: Test,Train and Validation split was complete

5.Define the model

Input ->None
Output-> A sequential model with LSTM layer

Step 1: We import Sequential() from keras and saved it in a variable named as model.

Step 2: We added an Embedding layer at the first with parameters MAX_NB_WORDS, EMBEDDING_DIM that we had defined in part-2 .

Step 3: We had then added a special drop out layer with rate = 0.2 to reduce overfitting.

Step 4: Now we added a LSTM layer with 100 memory units with dropout rate of 0.2 for the input layer, and recurrent dropout rate = 0.2. This was to learn the to learn the temporal dependencies between words in the sequences.

Step 5: Now we added a dense() layer with 'n' output units and the activation function as ‘softmax’ or ‘sigmoid’(according to binary or multiclass classification) which would gave the probability distribution over ‘n’ classes .

Step 6: Now our model was ready to compile.

Step 7: Now we compiled the model with categorical cross-entropy loss or binary cross-entropy, Adam optimizer, and accuracy metric for evaluation during training.

6.Fit the model

Input ->Pre-processed dataset from part 4 and compiled model from part 5

.

Output->Training of LSTM model using the preprocessed dataset and a trained model .

Step 1: We had fitted our model with the preprocessed dataset and predefined

Epoch number and batch size as our choice .

Step 2: Made the validation split of 50 % over test data.

Step 3: Our Model wass trained.

7.Check the accuracy of the model

Step 1: Using the trained model we predicted the labels of each tweet from test data set and store them in a variable.

Step 2: From sklearn.metrics we have imported the predefined obejects and using the output from step 1 and test data from part 3 we have got our Overall accuracy , F1 score and confusion matrix.

Step 3: Record accuracy, F1 score and confusion matrix in a word document.

Step 4: Complete.

Following these mentioned parts we have procedded through the experiment and saved our experiment results.

Conclusion :

This chapter contained the detailed discussion about algorithm that we used . The algorithm was the description about how did we used our methodology to train and test the model with our datasets . In next chapter we discussed about the results obtained by different results on different datasets.

CHAPTER 7 : *RESULT AND PERFORMANCE ANALYSIS*

We tested the algorithm in two different ways and stored our result .

For each language we had two different tables describing the accuracy of the model . Where one table described the result of model without fasttext embedding , another table described the result with fasttext embedding.

All over the dataset we maintained the 60:20:20 Test : Train : Validation split of the dataset.

ENGLISH DATASET

- ***<u>WITHOUT FASTTEXT EMBEDDING</u>***

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	*ACCURACY*
3	NONE	4024	1138	811	338	0.85	0.89	0.87	0.79575
	SEXISM	377	5470	110	354	0.77	0.52	0.62	
	RACISM	747	4851	242	471	0.76	0.61	0.68	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	NONE	3637	1390	559	705	0.87	0.82	0.85	0.78165
	SEXISM	498	5349	231	233	0.68	0.68	0.68	
	RACISM	864	4591	502	354	0.63	0.71	0.67	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
7	NONE	3972	1199	750	390	0.85	0.90	**0.87**	**0.80589**
	SEXISM	484	5394	186	247	0.72	0.66	**0.69**	
	RACISM	684	4858	235	534	0.74	0.56	**0.64**	

EXPLANATION : Hence forth we experimented the English dataset and find the accuracy of the model for 3 epochs . We got the best result for **epoch =7** .
We got the accuracy of **0.80589** and F1 score of **0.87** , **0.69** and **0.64** .

- *WITH FASTTEXT EMBEDDING*

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	NONE	3721	1378	571	641	0.87	0.85	0.86	0.81239
	SEXISM	818	4649	44	400	0.65	0.67	0.66	
	RACISM	542	5365	215	189	0.72	0.74	0.73	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	NONE	4010	1109	840	352	0.84	0.90	0.87	0.79987
	SEXISM	582	4929	164	636	0.78	0.48	0.59	
	RACISM	520	5385	195	211	0.73	0.71	0.72	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
7	NONE	3991	1188	750	641	0.84	0.91	**0.88**	**0.81825**
	SEXISM	662	4900	193	556	0.77	0.54	**0.64**	
	RACISM	516	5392	188	215	0.73	0.71	**0.72**	

EXPLANATION : Hence forth we experimented the English dataset and found the accuracy of the model for 3 epochs . We got the best result for **epoch =7** .

We got the accuracy of **0.81825** and F1 score of **0.88** , **0.64** and **0.72**.

- ✓ Overall : We got the best result using fasttext embedding .
- ✓ We got the best result for **epoch** =7 .
- ✓ We got the accuracy of **0.81825** and F1 score of **0.88** , **0.64** and **0.72**.

BENGALI DATASET

- *WITHOUT FASTTEXT EMBEDDING*

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	**Gender abusive**	504	467	339	58	0.78	0.81	0.80	0.58991
	Geopolitical	92	1110	43	123	0.68	0.43	0.53	
	Personal	86	1073	43	166	0.67	0.34	0.45	
	Political	39	1210	33	86	0.54	0.31	0.40	
	Religious	133	1098	56	81	0.70	0.62	0.66	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	**Gender abusive**	523	445	361	39	0.80	0.85	**0.83**	**0.5972**
	Geopolitical	102	1109	44	113	0.70	0.47	**0.57**	
	Personal	97	1061	55	155	0.64	0.38	**0.48**	
	Political	24	1231	12	101	0.67	0.19	**0.30**	
	Religious	117	1121	33	97	0.78	0.55	**0.64**	

EXPLANATION : Hence forth we experimented the Bengali dataset ***without fasttext embedding*** and found the accuracy of the model for 2 epochs . We got the best result for **epoch =5**.

We got the accuracy of **0.5972** and F1 score of **0.83**,**0.57,0.48,0.30 and 0.64**.

- ***WITH FASTTEXT EMBEDDING***

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	**Gender abusive**	521	482	324	41	0.74	0.91	0.81	0.61623
	Geopolitical	69	1118	35	146	0.66	0.32	0.43	
	Personal	84	1066	50	168	0.63	0.33	0.44	
	Political	46	1209	34	79	0.57	0.37	0.45	
	Religious	134	1083	71	80	0.65	0.63	0.64	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	**Gender abusive**	496	560	246	666	0.80	0.82	**0.81**	**0.61988**
	Geopolitical	79	1114	39	136	0.67	0.37	**0.47**	
	Personal	115	1028	88	137	0.57	0.46	**0.51**	
	Political	40	1220	23	85	0.63	0.32	**0.43**	
	Religious	152	1064	90	62	0.63	0.71	**0.67**	

EXPLANATION : Hence forth we experimented the Bengali dataset ***with fasttext embedding*** and found the accuracy of the model for 2 epochs . We got the best result for **epoch =5**.

We got the accuracy of **0.61988** and F1 score of **0.81**,**0.47,0.51,0.43 and 0.67**.

- Overall: LSTM model with fasttext embedding work better
- We got the best result for **epoch =5**.
- We got the accuracy of **0.61988** and F1 score of **0.81**,**0.47,0.51,0.43 and 0.67**.

ITALIAN DATASET

- *WITHOUT FASTTEXT EMBEDDING*

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	**Non Hate**	701	228	164	107	0.81	0.86	0.84	0.77
	Hate	228	701	107	164	0.68	0.58	0.63	
-------------	**---------------**	------	------	------	-----	--------	--------------	-------	-----------

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	**Non Hate**	690	233	159	118	0.81	0.85	**0.83**	**0.78166**
	Hate	233	690	118	159	0.68	0.72	**0.63**	

EXPLANATION : Hence forth we experimented the Italian dataset ***without fasttext embedding*** and find the accuracy of the model for 2 epochs . We got the best result for **epoch =5**.

We got the accuracy of **0.78166** and F1 score of **0.83 and 0.63**.

- *WITH FASTTEXT EMBEDDING*

➢ EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	**Non Hate**	701	228	164	107	0.81	0.86	0.84	0.76412
	Hate	228	701	107	164	0.68	0.58	0.63	
	------------	-----	----	-----	-----	-----------	-----------	-------	-----------

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	**Non Hate**	704	220	172	104	0.78	0.81	**0.83**	**0.765**
	Hate	220	704	104	172	0.68	0.57	**0.63**	

.EXPLANATION : Hence forth we experimented the Italian dataset ***with fasttext embedding*** and found the accuracy of the model for 2 epochs . We got the best result for **epoch =5**.

We got the accuracy of **0.765** and F1 score of **0.83 and 0.63**.

- ✓ Overall : We get the best result for without embedding .
- ✓ We got the best result for **epoch =5**.
- ✓ We got the accuracy of **0.78166** and F1 score of **0.83 and 0.63**.

GERMAN DATASET

- ✓ *WITHOUT FASTTEXT EMBEDDING*

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	**Non Hate**	765	86	300	86	0.72	0.92	0.81	0.68343
	Hate	86	765	86	300	0.55	0.27	0.37	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	**Non Hate**	665	194	192	162	0.78	0.80	**0.79**	**0.70734**
	Hate	194	665	162	192	0.54	0.50	**0.52**	

EXPLANATION : Hence forth we experimented the Italian dataset ***with fasttext embedding*** and found the accuracy of the model for 2 epochs . We got the best result for **epoch =5**.

We got the accuracy of **0.7073** and F1 score of **0.79 and 0.52**.

- ✓ *WITH FASTTEXT EMBEDDING*

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
3	**Non Hate**	679	154	232	148	0.75	0.82	0.78	0.68343
	Hate	154	679	148	232	0.51	0.40	0.45	

EPOCH	CLASS	TP	TN	FP	FN	PRECISION	RECALL	F1 SCORE	ACCURACY
5	**Non Hate**	679	154	232	148	0.75	0.82	**0.78**	**0.68343**
	Hate	154	679	148	232	.51	.40	**0.45**	

EXPLANATION : Hence forth we experimented the German dataset and found the accuracy of the model for 2 epochs . We got the best result for **epoch =5** without fasttext embedding .

We got the accuracy of **0.70734** and F1 score of **0.78 and 0.45.**

- ✓ Overall : We get the best result for without embedding .
- ✓ We got the best result for **epoch =5**.
- ✓ We got the accuracy of **0.7073** and F1 score of **0.79 and 0.52**

Conclusion :

- ❖ FINDINGS

  - ✓ Till now we got a satisfactory level of accuracy for all four datasets .

  - ✓ For English and Bengali dataset LSTM +FASTTEXT embedding method had out performed the LSTM + Without FASTTEXT model .

  - ✓ For German and Italian dataset LSTM + Without FASTTEXT model outperformed LSTM +FASTTEXT embedding method.

  - ✓ With the increasing epoch value , the accuracy of our model has also increased.

  - ✓ Since the number of hate tweets is much lesser than the number of non hate tweets F1 score for hate class is comparatively lower than non hate class.

NOTE :
True Positive(TP) = number of data for which both the truth value and the predicted value of the model is positive.

True Negative(TN) = number of data for which both the truth value and the predicted value of the model is negative .

False Positive(FP) = number of data for which the truth value is negative but the predicted value of the model is positive.

False Negative (FN) = number of data for which the truth value is positive but the predicted value of the model is negative.

Accuracy = how many predictions we got right = (True Positive + False Positive) / (Total number of data)

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

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

F1 – score = 2* (precision * recall) / (precision + recall)

CHAPTER 8 : COMPARISON OF RESULTS OF TWO DIFFERENT METHODS

We recorded the result using LSTM +FASTTEXT embedding and LSTM + Without FASTTEXT model. In this section we gave a comparative view of result obtained by these two methods .

ENGLISH DATASET

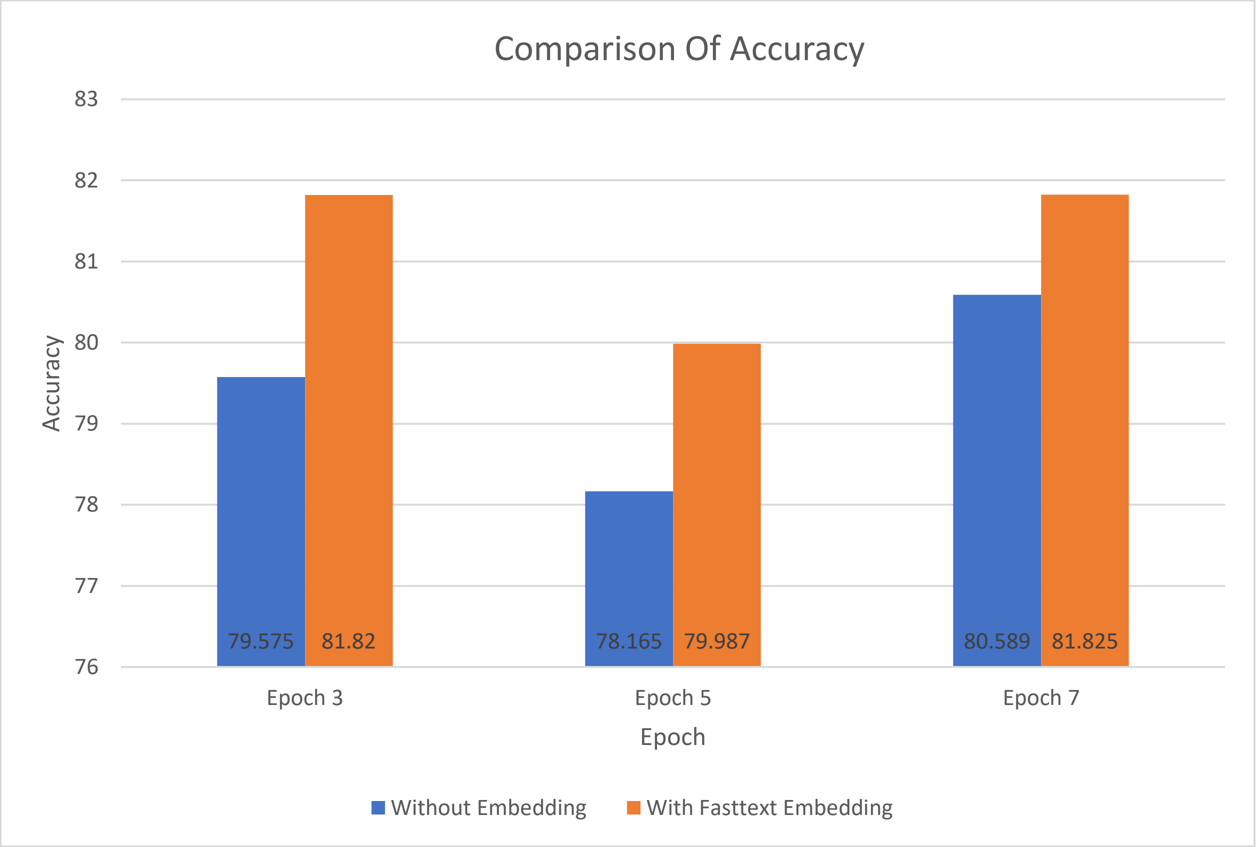


Figure 6 : *COMPARISON OF ACCURACY OF ENGLISH TWEETS*

BENGALI DATASET

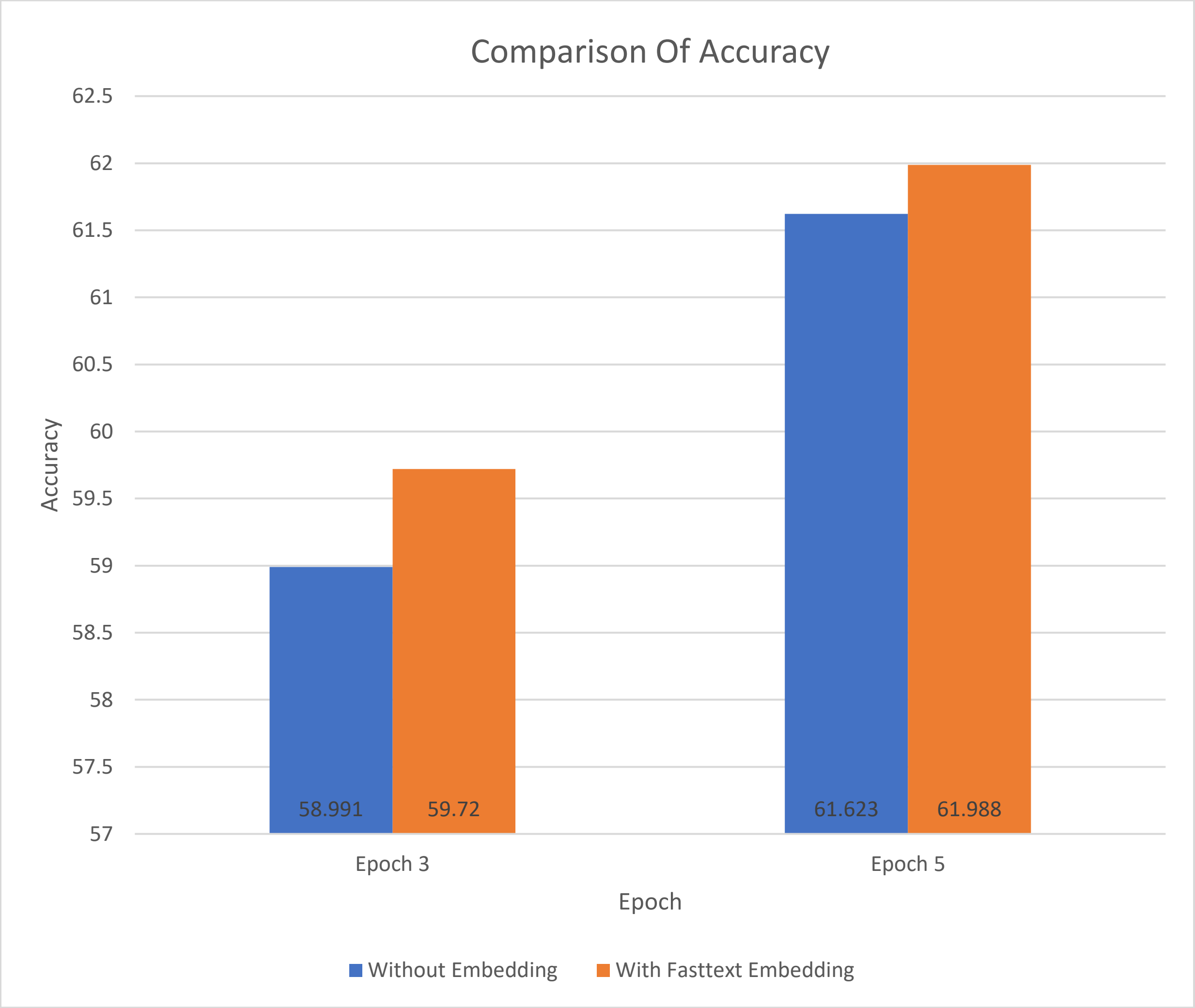


Figure 7 : COMPARISON OF ACCURACY OF BENGALI TWEETS

ITALIAN DATASET

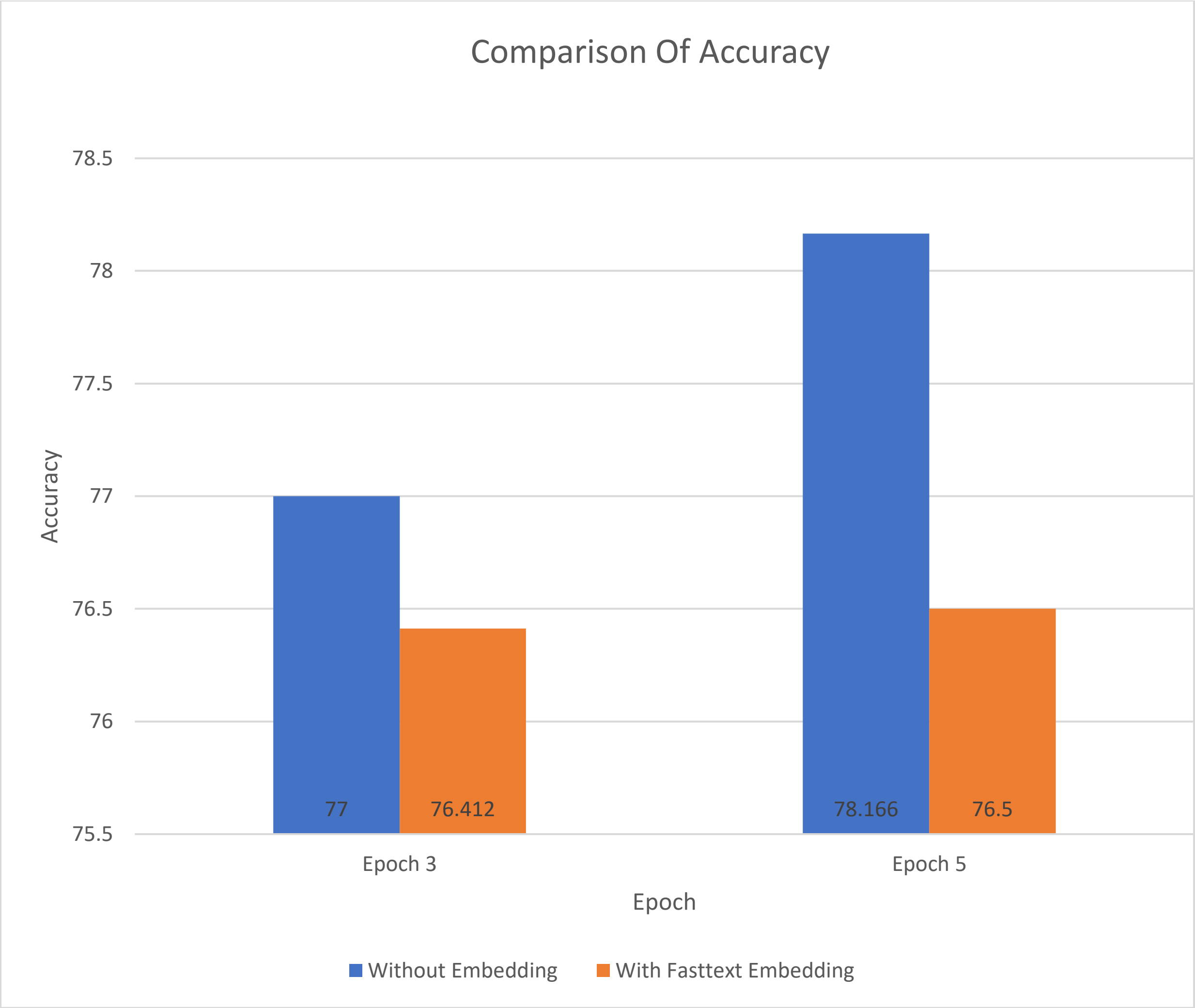


***Figure 8** : COMPARISON OF ACCURACY OF ITALIAN TWEETS*

GERMAN DATASET

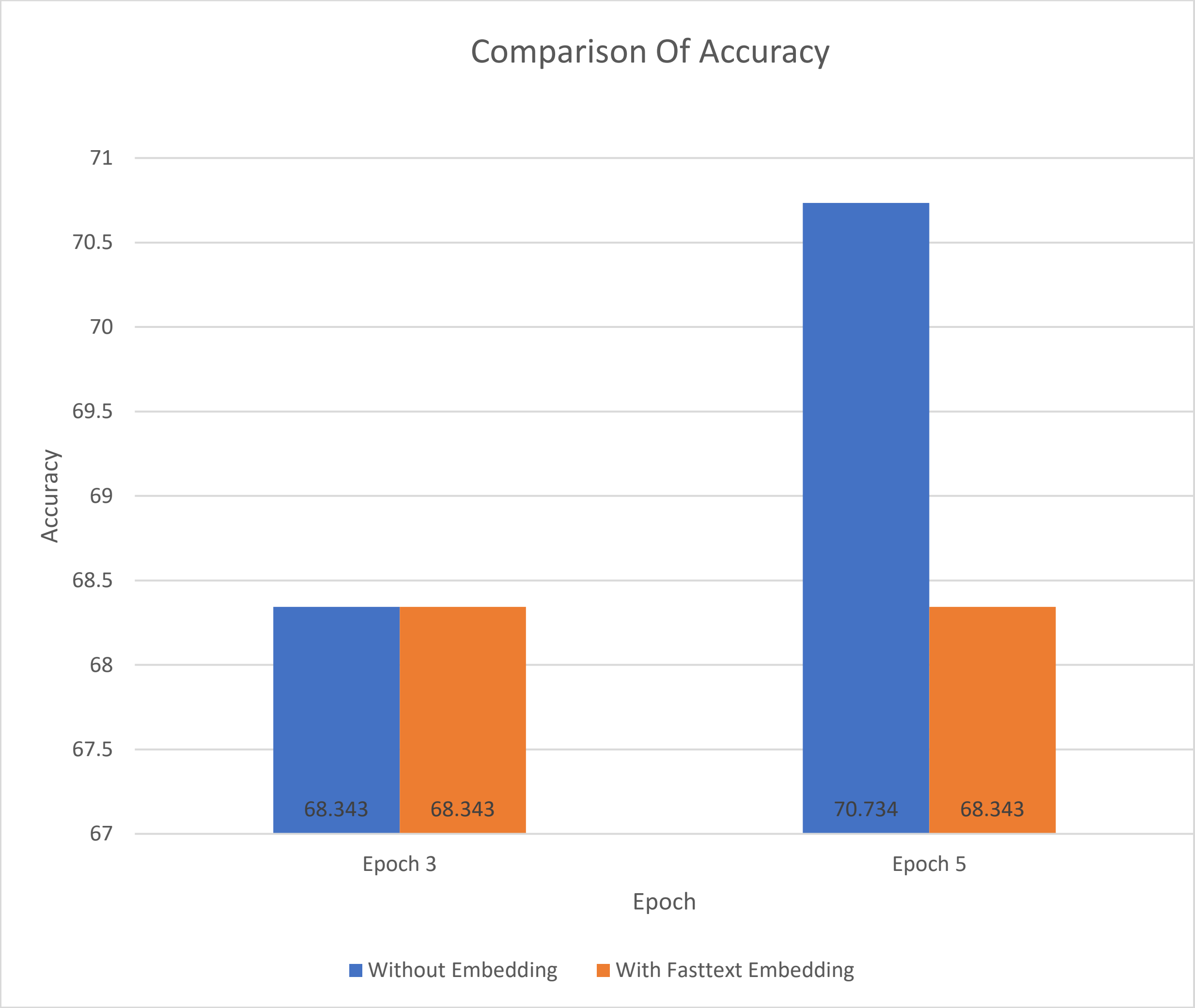


Figure 9 : *COMPARISON OF ACCURACY OF GERMAN TWEETS*

CHAPTER 9 : *CONCLUSION AND FUTURE WORK*

In this article , our primary goal was to establish the usage of LSTM neural network for the task of hate speech detection and then trying to increase the accuracy of LSTM neural network with the freely available fasttext word embedding model . After robust experimenting using both of the methods and recording the results , I came to the conclusion that LSTM is a proper neural network for this type of work . It was also proven that giving the inputs in vector form increased the accuracy of the model for some cases . So fasttext method was also helpful to increase the accuracy of this model .

Now, discussing about the short comings of this was also necessary .

In the result section it was clearly visible that the hate class's F1 score is much lesser than the Non hate class . This is a major shortcoming of this article .

Another shortcoming of the system was that , for some cases our model failed to detect as hate speech .

For example , we had tested our system with this sentence *"It's fu**ing disgusting to see so many hate comments in this comment section".* The meaning of the sentence indicated towards the frustration of a user to see so many hate comments . But our model detected this as the hate comment for both methods .(I have used ** manually)

Another example , *"You are such a pu$$y man "* . In this example a slang word is masked using a special alphabet '$'. This should be hate comment semantically. But our model detected this as a non-hate comment .
Now this showed a serious failure of our model.
Reason behind this can be several . I discussed some of my insights here .

- Only the English dataset had the satisfactory number of tweets . Rest three dataset had very limited number of tweets to train our model .
- Bengali dataset had very limited number of tweets and also divided over many subclasses , which were very close to each other semantically. This leaded to a poor F1 score for this dataset .
- While using fasttext embedding , we replaced the words which didn't match any word in pretrained word vector file by null vector . Some hate words which were masked using some unnecessary alphabets were ignored by the system. This must be taken care into .
- Some tweets which contained some slang words , but *semantically* means non-hate ,detected as hate tweet . Reason behind this could be the presence of slang words . But our model failed to detect the actual semantic meaning of the tweet .

- In English and Bengali dataset some sarcastic tweets were labeled with the hate class. But actually they were not . Sometimes it were very hard to get the actual semantic meaning of those sarcastic tweets . These leaded to the downfall of the accuracy of the model .

So any future research work based upon this article must look into these shortcomings . There are several future scope available to overcome these shortcomings . In future work any researcher may use some other popular RNN based neural network like GRU , BiLSTM model and combine it with the LSTM model . A very important improvisation can be done by feeding the respective language's fasttext model with the hateful words of that particular language . This will lead to the decrease of null embeddings of some hate words . Other available embeddings can be also used to vectorize the words present in a tweet .

We believe that this article we will be beneficial for future research in multilingual hate speech detection.

CHAPTER 10 : 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.

[32] 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).

[33] Aluru, Sai Saket and Mathew, Binny and Saha, Punyajoy and Mukherjee, Animesh,2020, Deep Learning Models for Multilingual Hate Speech Detection

[35] Waseem, Z. and Hovy, D., 2016, June. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop* (pp. 88-93).

[36] Karim, M.R., Dey, S.K., Islam, T., Sarker, S., Menon, M.H., Hossain, K., Hossain, M.A. and

Decker, S., 2021, October. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 1-10). IEEE.

[37] Bosco, C., Felice, D.O., Poletto, F., Sanguinetti, M. and Maurizio, T., 2018. Overview of the evalita 2018 hate speech detection task. In *Ceur workshop proceedings* (Vol. 2263, pp. 1-9). CEUR.

[38] Wiegand, M., Siegel, M. and Ruppenhofer, J., 2018. Overview of the germeval 2018 shared task on the identification of offensive language.

[39] https://fasttext.cc/

[40] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.