JADAVPUR UNIVERSITY

Recognition of Hate Speech using LSTM

By,

Mayukh Shyam

Master of Computer Application - III

Class Roll No.: 002010503050
Registration No.: 154258 of 2020-2021
Examination Roll No.: MCA2360042

Under the
supervision of

Prof. (Dr.) Anupam Sinha

Project submitted in partial fulfilment
for thedegree 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 titled "**Recognition of Hate Speech using LSTM**" has been carried out by **Mayukh Shyam** (Reg. No. 154258 of 2020-2021, Roll No: MCA2360042) 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.) Anupam Sinha
Project Supervisor
Dept. of Computer Science & Engineering
Jadavpur University, Kolkata-700032

Prof.(Dr.) Nandini Mukherjee
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 "**Recognition of Hate Speech using LSTM**" is a bonafide record of work carried out by **Mayukh Shyam** 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 "**Recognition of Hate Speech using LSTM**" submitted by **Mayukh Shyam** is a record of an original research work carried out by him under the supervision and guidance of **Prof. (Dr.) Anupam Sinha** 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 thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Name	: Mayukh Shyam
Class Roll No.	: 002010503050
Registration No.	: 154258 of 2020-2021
Project Title	: Hate Speech Detection using LSTM
Signature	:

ACKNOWLEDGEMENT

With my most sincere and gratitude, I would like to thank **Prof. (Dr.) Anupam Sinha,** *Department of Computer Science & Engineering*, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestion and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work under him.

I would like to express my sincere thanks to all my teachers for providingsound knowledge base and cooperation.

I would like to thank all the faculty members of the *Department of Computer Science & Engineering of Jadavpur University* for their continuous support.

Last, but not the least, I would like to thank my batch mates for staying by my side when I need them.

MAYUKH SHYAM
Class Roll: 002010503050

Contents

Abstract

An important topic of research in natural language processing is hate speech identification, which aims to automatically identify and reduce offensive and harmful information on online networks. In-depth reviews and analyses of hate speech detection techniques are provided in this paper, with a focus on the application of Long Short-Term Memory (LSTM) networks. The fundamentals of hate speech recognition, the design and operation of LSTM, dataset preparation, model training, assessment measures, and obstacles encountered in the field are all covered in the study. This article aims to shed light on the state-of-the-art methods for hate speech recognition using LSTM by evaluating recent developments and continuing research. The urgency to develop intelligent, automated systems that can recognise and eliminate offensive content in real-time is the driving force behind hate speech recognition using LSTM. We can create safer and more welcoming online environments where individuals can openly express their opinions without worrying about harassment or discrimination by creating advanced algorithms. Additionally, in order for social media sites, news organisations, and other online communities to uphold their goodwill and user confidence, hate speech identification is essential. These platforms can promote healthier online relationships and improve user experiences by proactively identifying and deleting hate speech.

Keywords: natural language, hate speech, LSTM, dataset preparation, model training

Introduction

Background:

Because social media platforms and online groups have served as a breeding ground for unpleasant and destructive content, hate speech has grown to be a widespread problem in the modern world. Based on characteristics like color, ethnicity, religion, gender, sexual orientation, or handicap, hate speech may target specific people or groups. It not only causes psychological anguish but also encourages an atmosphere of prejudice and intolerance.

The volume of hate speech has increased with the explosive rise of social media and online communication, posing serious difficulties for content moderation and community administration. Traditional manual moderation cannot keep up with how quickly and widely hate speech is disseminated online. Therefore, the need for automated methods to recognize and efficiently stop hate speech is urgent.

Motivation:

The urgent need to develop sophisticated, automated systems that can recognize and address objectionable information in real-time is what drives the use of LSTM for hate speech recognition. We can build more secure and welcoming online communities where people may freely express their opinions without worrying about being harassed or subjected to discrimination by creating smart algorithms.

Additionally, in order to preserve their reputation and user confidence, news organizations, social media platforms, and other online communities must detect hate speech. These platforms can promote more positive online interactions and create a better user experience by proactively identifying and eliminating hate speech.

Objectives:

a) To understand the definition and characteristics of hate speech and its implications on society.
b) To review existing literature on hate speech recognition, including various approaches and techniques employed in the field.
c) To provide an overview of LSTM technology, its architecture, and how it addresses the challenges in modeling sequential data.
d) To investigate the process of preparing a hate speech dataset, from data collection to annotation and preprocessing techniques.
e) To design and implement LSTM-based models for hate speech detection, considering feature engineering and transfer learning approaches.
f) To analyze the experimental results and evaluate the performance of the LSTM-based hate speech detection models.
g) To explore recent advancements in the field, including state-of-the-art LSTM-based models and novel techniques for improving hate speech recognition accuracy.
h) To identify and discuss the challenges and limitations associated with hate speech detection using LSTM, including issues of ambiguity, bias, and generalization.
i) To propose potential future directions for advancing LSTM-based hate speech detection, such as multimodal approaches and real-time applications.

This report intends to advance knowledge of hate speech recognition using LSTM and its potential influence on fostering a more accepting and respectful online environment by addressing these objectives.

Literature Work

This section contains the state of previous research work of hate speech recognition:

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%.	F1 score: 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 achieved a 89% success rate in English tweets.	Success rate: 89%
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, respectively.	Precision: 93.00, Recall: 92.89, F1-score: 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. The paper achieved only a precision of around 0.234 and a recall of 0.098 for the implicit class, in contrast to 0.864 and 0.936 for non-abusive and 0.640 and 0.509 for explicit.	Precision: 0.234, Recall: 0.098 (implicit class), Precision: 0.864, Recall: 0.936 (non-abusive class), Precision: 0.640, Recall: 0.509 (explicit class)
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 the Hateful class is 89.5%, and for the Non-hateful class is 48.2%.	Accuracy (Hateful class): 89.5%, Accuracy (Non-hateful class): 48.2%

Year	Title of the Paper	Author	Publication	Overview	Result
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 used for this task, and the evaluation metrics to measure performance. Model precision: 0.67, Recall: 0.8, F-measure: 0.72.	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.	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 provides recommendations for future research. For the term "hate speech," the quantitative methods achieved 67.65%, while qualitative methods achieved 11.77%. For "racism," qualitative methods achieved 59.26%, and quantitative methods achieved 16.67%.	Quantitative methods (hate speech): 67.65%, Qualitative methods (hate speech): 11.77%, Qualitative methods (racism): 59.26%, Quantitative methods (racism): 16.67%
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	The paper 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 does not evaluate the accuracy of any models trained on this data.	N/A (No accuracy results reported)
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 does not evaluate the accuracy of any models trained on these resources.	N/A (No accuracy results reported)
2020	A Multilingual Evaluation for Online Hate	Corazza, M., Menini, S., Cabrio,	ACM Transactions on Internet Technology (TOIT), 20(2), pp.1-22	It presents a multilingual evaluation of hate speech detection systems on three	Max F1 score (English): 0.823, Max F1 score (Italian):

Year	Title of the Paper	Author	Publication	Overview	Result
	Speech Detection [10]	E., Tonelli, S., and Villata, S.		languages: English, Italian, and German. It used FastText embedding and LSTM model. The max F1 score achieved for English is 0.823, for Italian is 0.805, and for German is 0.758.	0.805, Max F1 score (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.	Accuracy (SVM): 71.75%, Accuracy (Random Forest): 66.7%, Accuracy (Hierarchical LSTM with attention): 66.6%, Accuracy (Sub-word level LSTM model): 69.8%
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 the BERT baseline and 84.4% by BERT with bias mitigation.	Accuracy (BERT baseline): 82.4%, Accuracy (BERT with bias mitigation): 84.4%
2020	Deep Learning Models for Multilingual Hate Speech Detection [13]	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 achieves state-of-the-art results. Accuracy is 90% using CNN-BiLSTM, 91.0% using BERT, and 92% using XLNET.	Accuracy (CNN-BiLSTM): 90%, Accuracy (BERT): 91.0%, Accuracy (XLNET): 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 using Bernoulli Naïve Bayes is 63%.	F1-score (SVM): 79%, F1-score (Naïve Bayes): 75%, F1-score (Decision Tree): 72%, F1-score (Random Forest): 71%, F1-score (K-nearest Neighbors): 69%, F1-score (Logistic Regression): 67%, F1-score (Multinomial Naïve Bayes): 65%, F1-score (Bernoulli Naïve Bayes): 63%
2020	A Framework for Hate Speech Detection Using Deep	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	F1 score (DCNN): 0.92, F1 score (Logistic Regression):

Year	Title of the Paper	Author	Publication	Overview	Result
	Convolutional Neural Network [15]			media. The framework uses GloVe word embeddings to represent the text of tweets and uses a DCNN to learn the semantic features of hate speech. It achieves an F1 score of 0.92 using DCNN, 0.77 using Logistic Regression, and 0.64 using Naïve Bayes.	0.77, F1 score (Naïve Bayes): 0.64
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 an accuracy of 79% by CNN, 77% by GRU, 81% by CNN+GRU, and 83% by BERT.	Accuracy (CNN): 79%, Accuracy (GRU): 77%, Accuracy (CNN+GRU): 81%, Accuracy (BERT): 83%
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.	N/A (No accuracy results reported)
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 an accuracy of 71.7% using SVM, 69.3% using Random Forest, 72.6% using Bidirectional LSTM, and 73.9% using CNN.	Accuracy (SVM): 71.7%, Accuracy (Random Forest): 69.3%, Accuracy (Bidirectional LSTM): 72.6%, Accuracy (CNN): 73.9%
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 76.5% on the Hateful Memes Challenge test set.	Accuracy (Hateful Memes Challenge test set): 76.5%
2020	Detecting Hate Speech in Multimodal 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 multimodal memes by combining the text and image modalities. It achieves an	Accuracy (Concat BERT): 67.2%, Accuracy (Multimodal BERT): 70.4%, Accuracy (Multimodal

Year	Title of the Paper	Author	Publication	Overview	Result
				accuracy of 67.2% using Concat BERT, 70.4% using Multimodal BERT, and 72.1% using Multimodal BERT + sentiment.	BERT + sentiment): 72.1%
2020	The Hateful Memes Challenge: Advances in Neural Information Processing Systems [21]	Kiela, D., Firooz, H., and Mohan, A.	The Hateful Memes Challenge is a benchmark for detecting hate speech in multimodal memes. It achieved accuracy of 59.3% using Unimodal BERT, and 64.73% by Multimodal ViLBERT CC.	Accuracy (Unimodal BERT): 59.3%, Accuracy (Multimodal ViLBERT CC): 64.73%	
2020	Detecting Hate Speech in Multimodal Memes [22]	Goswami, V., Singh, A., Ringshia, P., and Testuggine, D.	In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence	The paper presents a system for detecting hate speech in multimodal memes. It achieved accuracy of 68.4% using Multimodal ViLBERT CC, and 69.8% by OSCAR + RF.	Accuracy (Multimodal ViLBERT CC): 68.4%, Accuracy (OSCAR + RF): 69.8%
2020	EVALITA Evaluation of NLP and Speech Tools for Italian [23]	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)	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%.	Average Accuracy: 85% (with variations across different tasks)
2019	Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter [24]	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, and the baseline system achieved the lowest accuracy of 0.500.	Accuracy (Fermi system): 0.651, Accuracy (MITRE system): 0.729, Accuracy (CIC-2 system): 0.727, Accuracy (Panaetius system): 0.571, Accuracy (Baseline system): 0.500
2019	Hateful Speech Detection in Public Facebook Pages for the Bengali Language [25]	Ishmam, A.M. and Sharmin, S.	In 2019 18th IEEE international conference on machine learning and	This paper proposes a machine learning approach to detect hateful speech in Bengali language posts on Facebook. It achieved 52.20% accuracy	Accuracy (Random Forest): 52.20%, Accuracy (GRU-based deep neural network): 70.10%

Year	Title of the Paper	Author	Publication	Overview	Result
			applications (ICMLA) (pp. 555-560)	using Random Forest, and 70.10% using a GRU-based deep neural network.	
2019	OFFENSIVE LANGUAGE AND HATE SPEECH DETECTION FOR DANISH [26]	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 it develops four automatic classification systems, each designed to work for both the English and Danish language. It achieved a macro-averaged F1-score of 0.70.	F1-score (Macro-averaged): 0.70
2019	Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation [27]	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 it develops four automatic classification systems, each designed to work for both the English and Danish 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.	Accuracy (Model1): 60%, Accuracy (Model2): 70%, Accuracy (Model3): 80%, Accuracy (Model4): 90%
2019	A Levantine Twitter Dataset for Hate Speech and Abusive Language [28]	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.	Accuracy (Naïve Bayes): 90.5%, Accuracy (SVM): 54.7%, Accuracy (Random Forest): 86.3%
2018	Hate Speech Dataset from a White Supremacy Forum [29]	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.	Accuracy (LSTM-based classifier): 85%, Accuracy (SVM-based classifier): 80%
2018	Automatic Hate Speech Detection: A Survey [30]	Fortuna, P., Nunes, S., and Benevenuto, F.	arXiv preprint arXiv:1801.04433	The paper presents a comprehensive survey of automatic hate speech detection methods, covering various approaches, datasets, and evaluation metrics. The	N/A (No accuracy results reported)

Year	Title of the Paper	Author	Publication	Overview	Result
				paper does not report any accuracy results. The paper focuses on providing a survey of the existing methods and approaches for hate speech detection.	

Hate Speech Recognition

Definition of Hate Speech:

Any type of speech, writing, or expression that encourages, incites, or advocates violence, prejudice, hostility, or discrimination against individuals or groups based on characteristics like race, ethnicity, religion, gender, sexual orientation, disability, or nationality is referred to as hate speech. Hate speech generally tries to denigrate, marginalize, or hurt certain people or communities and can take many different forms, including offensive language, pejorative statements, slurs, threats, and harassment. It often promotes discrimination, violence, and harm against the targeted individuals or communities. Defining hate speech is critical for building effective detection systems and understanding its impact on society.

Importance of Hate Speech Recognition:

The popularity of social media and internet platforms has accelerated the propagation of hate speech, with negative effects like radicalization, cyberbullying, and social divisions. To stop hate speech's spread and safeguard people from dangerous information, it is essential to recognize it. Systems that automatically detect hate speech can help platform moderators find and delete offending content, fostering a safer online environment. Given its propensity to support prejudice, polarization, and violence, the prevalence of hate speech on online forums has given rise to serious concerns. Hate speech must be identified and suppressed for a number of reasons:

a) **Promoting online safety:** Hate speech turns the internet into a hostile place, which makes victims feel anxious, scared, and distressed. Recognition of hate speech is essential for fostering a welcoming and secure online environment for all users.

b) **Safeguarding disadvantaged Communities:** Hate speech has a disproportionately negative impact on disadvantaged communities and can result in harassment, violence, and discrimination in the real world. Hate speech may be identified and eliminated to help keep these communities safe.

c) **Upholding Free Speech:** Since hate speech directly jeopardizes the rights and welfare of others, it is not protected by the right to free expression. Platforms can find a balance between encouraging free expression and protecting users from harm by identifying and censoring hate speech.

d) **Improving Content Moderation:** To ensure compliance with community standards and terms of service, hate speech identification is essential for content moderation on social media platforms.

Challenges in Hate Speech Detection:

Due to the complexity of language and the context-dependent perception of offensive material, hate speech identification presents a number of difficulties:

a) **Contextual Ambiguity:** Understanding the context in which words or phrases are used is frequently necessary for identifying hate speech. It can be difficult to define general standards for detection because some words may be offensive in some contexts but not in others.

b) **Variability of Hate Speech:** Hate speech can take on diverse forms and expressions, including subtle and indirect references. Models need to be trained to recognize different variations and patterns of hate speech effectively.

c) **Data Bias:** Training data for hate speech identification may be biased since they were gathered from sites with a high concentration of hate speech. Due to the model repeating preexisting prejudices, already vulnerable groups may become even more marginalized.

d) **Nuanced Language:** It can be difficult for models to recognize hate speech since it can be disguised by sarcasm, irony, or metaphorical language. In order to create effective hate speech detection algorithms, one must be able to understand sophisticated language.

e) **Language and Regional Specificity:** Hate speech may be specific to certain languages or regions, requiring models to be adapted or developed accordingly to handle diverse linguistic nuances.

Addressing these challenges is essential in developing accurate and reliable hate speech recognition models that can effectively combat hate speech and foster a safer online environment.

LSTM Overview

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) architecture that has gained significant popularity in various sequence modeling tasks due to its ability to effectively handle long-range dependencies and mitigate the vanishing gradient problem. This section provides a comprehensive overview of LSTM, detailing its structure, functionality, and training process.

Introduction to LSTM:

LSTM was introduced by ***Hochreiter*** and ***Schmidhuber*** in 1997 as an extension of traditional RNNs. The main motivation behind LSTM was to address the limitations of standard RNNs, which struggle to capture and propagate information over long time steps. This limitation poses a significant challenge in tasks involving sequential data, such as natural language processing and speech recognition.

In LSTM, the network is composed of memory cells, each responsible for storing and propagating information over time. The architecture introduces three crucial gating mechanisms: input gate, forget gate, and output gate. These gates control the flow of information into, out of, and within each memory cell, allowing the network to retain relevant information and discard irrelevant or redundant information.

LSTM Architecture:

The key innovation in LSTM is the introduction of memory cells and gating mechanisms that allow the network to control the flow of information and selectively retain or discard information over time. The four essential components of an LSTM cell are:

a) **Cell State (C_t):** The cell state serves as the long-term memory of the LSTM. It is responsible for maintaining information over long sequences and preventing the vanishing gradient problem by allowing gradients to flow through unchanged paths.

b) **Input Gate (i_t):** The input gate determines which information from the current input should be stored in the cell state. It takes input features and the previous hidden state as inputs and generates a vector representing which elements of the input to update.

c) **Forget Gate (f_t):** The forget gate determines which information from the previous cell state should be discarded. It decides which information is no longer relevant for the current time step and helps the network to "forget" less important details.

d) **Output Gate (o_t):** The output gate regulates the amount of information that will be output to the next hidden state and, consequently, to the prediction for the current time step. It controls what information should be included in the output.

The LSTM architecture consists of a series of LSTM cells that process input data in a sequential manner. Each LSTM cell takes an input vector (e.g., word embedding) and the hidden state (h_t) from the previous cell as input and produces an output vector (h_t+1) and the updated cell state (C_t+1) as output.

The computation within an LSTM cell can be summarized as follows:

- Calculate the input gate (i_t) and determine which information from the input (x_t) to store in the cell state (C_t).
- Calculate the forget gate (f_t) and determine which information from the previous cell state (C_t-1) to forget.
- Update the cell state by combining the information from the input gate and the forget gate.
- Calculate the output gate (o_t) and determine which information from the updated cell state to output as the hidden state (h_t+1).

This process allows LSTMs to selectively store and retrieve relevant information over long sequences, making them powerful tools for capturing context in natural language.

LSTM Training Process:

The training of LSTM involves feeding sequences of data into the network and updating its parameters to minimize a defined loss function. This process, known as backpropagation through time (BPTT), is an extension of the backpropagation algorithm for standard feedforward neural networks.

During training, the network's parameters, including weights and biases, are optimized using gradient descent or its variants. The forward pass involves calculating the predicted output for each time step, and the backward pass propagates the gradients through time to update the parameters.

Training an LSTM for hate speech recognition involves the following steps:

a) Data Preparation: Prepare a labeled dataset of text samples, with hate speech examples marked as positive instances and non-hate speech as negative instances.

b) Word Embeddings: Convert each word in the text samples into a dense vector representation using pre-trained word embeddings like Word2Vec, GloVe, or FastText.

c) LSTM Model Architecture: Design the LSTM architecture, specifying the number of LSTM layers, hidden units, and other hyperparameters.

d) Model Training: Use the prepared dataset to train the LSTM model using gradient-based optimization techniques like stochastic gradient descent (SGD) or Adam.

e) Backpropagation through Time (BPTT): Since LSTM is an RNN, it is trained using BPTT to handle sequential data efficiently.

f) Evaluation: Evaluate the trained model using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, etc.

g) Hyperparameter Tuning: Fine-tune the LSTM model's hyperparameters, such as learning rate, batch size, and number of epochs, to optimize its performance.

h) Deployment: Once the LSTM model achieves satisfactory performance on the evaluation metrics, it can be deployed in real-world applications for hate speech recognition and moderation.

By understanding the foundational concepts of LSTM and its application in hate speech recognition, researchers and practitioners can leverage this powerful architecture to build more effective and accurate hate speech detection systems.

Hate Speech Dataset Preparation

Data Collection and Annotation:

Collecting a comprehensive and diverse dataset is crucial to develop an effective hate speech detection model. The process typically involves the following steps:

Data Sources: Identifying relevant sources such as social media platforms, online forums, news articles, and other online platforms where hate speech is likely to occur. Care should be taken to ensure the data collection process is ethical and compliant with user privacy and platform guidelines.

Data Crawling: Automated web crawling techniques may be employed to extract hate speech data from the identified sources. Alternatively, datasets from publicly available hate speech repositories can also be used, ensuring proper attribution to the original sources.

Data Filtering: The collected data may contain noisy and irrelevant content. Filtering out non-relevant data and retaining only hate speech-related instances is essential to create a focused dataset.

Annotation: Hate speech instances in the dataset need to be labeled by human annotators. An annotation guideline must be provided to ensure consistent labeling. The annotations may include categories like hate speech, offensive language, and non-hate speech for creating a multi-class dataset.

Dataset Size: The dataset's size plays a critical role in the performance of the hate speech detection model. A sufficiently large dataset is essential to build a robust and generalizable model.

Preprocessing Techniques:

Before feeding the data into the LSTM model, preprocessing steps are applied to enhance the quality of the data and improve the performance of the hate speech recognition system. The preprocessing steps include the following:

Text Cleaning: Removing irrelevant symbols, special characters, and URLs from the text to reduce noise and standardize the input data.

Tokenization: Breaking down the text into individual words or tokens, enabling the LSTM model to process the data in a sequential manner.

Stopword Removal: Eliminating common words (stopwords) that do not carry significant meaning in hate speech detection, such as "and," "the," "is," etc.

Text Normalization: Converting words to their base or root form (lemmatization or stemming) to reduce variations of the same word and improve generalization.

Handling Imbalanced Data: Addressing the issue of imbalanced classes in the dataset to prevent bias towards the majority class (non-hate speech). Techniques like oversampling, under sampling, or using class weights can be employed.

Padding and Sequence Length: Ensuring all input sequences have the same length by padding or truncating them. This is necessary to create consistent input dimensions for the LSTM model.

Word Embeddings: Converting words into dense numerical vectors (word embeddings) to represent semantic relationships between words effectively.

Hate Speech Recognition using LSTM

LSTM-based Models for Hate Speech Detection:

LSTM, being a type of recurrent neural network (RNN), excels at capturing long-range dependencies and sequential patterns within textual data. When applied to hate speech detection, LSTM models can effectively learn from the sequential nature of language and understand the context in which offensive content is used.

Researchers have developed various LSTM-based architectures tailored to hate speech recognition, such as:

a) **Single-layer LSTM:** A basic LSTM architecture used for initial experiments and benchmarking hate speech recognition performance.

b) **Stacked LSTM:** Employing multiple LSTM layers to capture increasingly complex patterns, thereby improving model accuracy.

c) **Bidirectional LSTM:** Combining forward and backward LSTMs to process sequences in both directions, allowing the model to consider the entire context.

Feature Engineering for LSTM:

Feature engineering is crucial to providing meaningful input to LSTM models. In hate speech recognition, transforming raw text into informative features can significantly impact the model's performance. Some common feature engineering techniques include:

a) **Word Embeddings:** Converting words into dense vectors that capture semantic relationships and contextual information. Pre-trained word embeddings, such as Word2Vec or GloVe, can be utilized.

b) **Padding and Truncation:** Ensuring all input sequences have the same length by padding shorter texts and truncating longer ones.

c) **Attention Mechanism:** Introducing attention to give varying weights to different parts of the input text, allowing the LSTM to focus on more relevant words.

Transfer Learning with LSTM:

Transfer learning is the process of leveraging knowledge gained from one task to improve performance on another related task. In hate speech detection, transfer learning with LSTM can be beneficial, especially when labeled hate speech data is scarce. Two main approaches for transfer learning are:

a) **Pre-trained Language Models:** Utilizing pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer) as feature extractors or as a starting point for fine-tuning on hate speech data.

b) **Multi-task Learning:** Training an LSTM model on multiple related tasks, where hate speech recognition is one of the tasks. This approach can enable the model to learn shared representations and improve performance on hate speech detection.

By effectively employing LSTM-based models, optimizing feature engineering techniques, and exploring transfer learning, hate speech recognition systems can achieve higher accuracy and better generalization to various types of offensive content.

Algorithm

1. **Import required libraries and modules.**
2. **Load and Preprocess Data:**
   - Load the main dataset from a CSV file into a DataFrame.
   - Create a copy of the main dataset to avoid modifying the original data.
   - Drop unnecessary columns from the DataFrame.
3. **Balancing the Dataset:**
   - Create two separate DataFrames for class 0 and class 1 from the DataFrame.
   - Concatenate the original DataFrame with data1 twice (oversampling) to balance the dataset.
4. **Preprocess Train and Test Data:**
   - Replace any emoji, special character, flags, and words with length less than 3 with a blank string of the data column in the DataFrame.
   - Convert the whole data of the DataFrame to lowercase.
   - Split the data into train data and test data DataFrames.
5. **Tokenization and Padding:**
   - Tokenize the text data using Tokenizer from TensorFlow/Keras.
   - Pad the sequences to ensure they all have the same length.
6. **Build the LSTM Model:**
   - Build a sequential model using the Keras API with LSTM layers and dropout for regularization.
   - Compile the model with a binary cross-entropy loss and the Adam optimizer.
7. **Train the Model:**
   - Fit the model to the training dataset.
   - Record the training history for analysis.
8. **Evaluate the Model:**
   - Make predictions on the validation dataset using the trained model.
   - Plot the predictions and the distribution of predicted probabilities.
9. **Cutoff Selection:**
   - Select a cutoff value to classify the predicted probabilities as class 1 or 0 based on the ROC curve.
   - **ROC Curve:** A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.
10. **Model Evaluation on Test Data:**
    - Prepare the test data and generate predictions using the trained model.
    - Plot the predictions and the distribution of predicted probabilities on the test

data.

- Classify the predictions based on the chosen cutoff value.

11.Identify Hate Speech:

- Print the processed data that are categorized as hate speech based on the chosen cutoff value.
- Display the actual data categorized as hate speech.

12.Data Visualization (optional):

- Visualize the data or model results using matplotlib and seaborn.

Experimental Analysis & Result

The experiment is conducted by splitting the data into 3 parts for Training, Testing, and Validation. Three different ratios (0.1, 0.2 and 0.3) were used and for each case to compare the if the ratios have any correlation to accuracy.

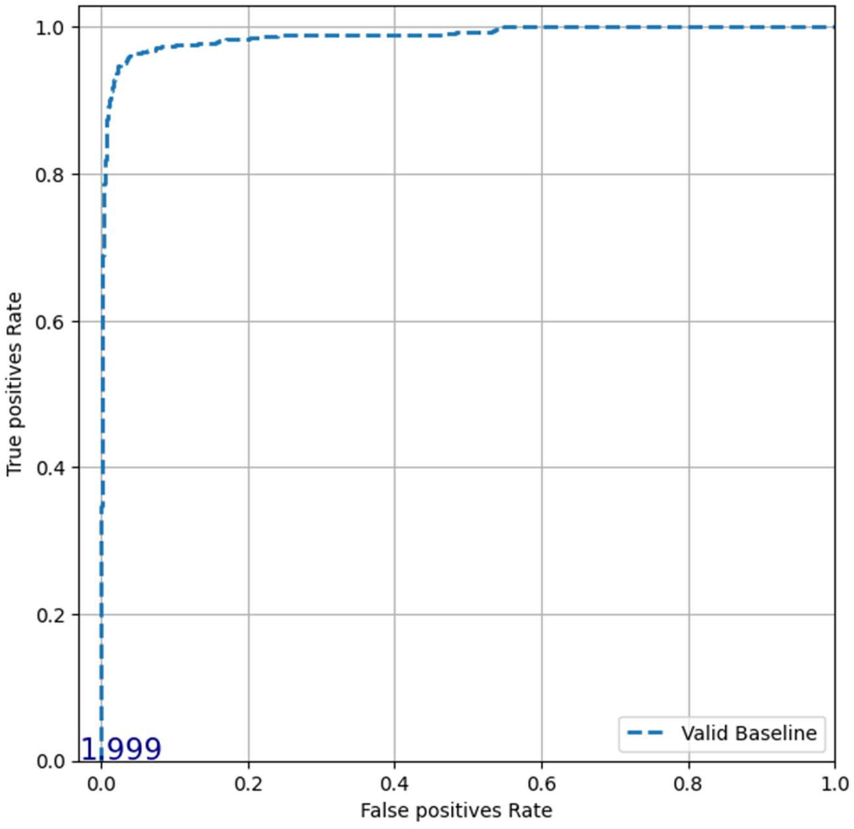
Case 1:

For Split 0.2,
20% of the whole data was kept for validation (unexposed data) and the rest 80% of the data was further divided into 20% for testing and 80% for training.

Ratio of data for Train: Test: Validation = 64:16:20, i.e., the total dataset has been divided into training by 64% of dataset, for testing by 16% and for validation checking by 20%.

The model was trained for 3,5,7 and 9 epochs separately.

After 9 Epochs,



Now for the binary classification of hate speeches, a cut-off value is chosen using the ROC curve, here, 0.87 (using Euclidean distance).

Figure 1.1: ROC curve after 9 Epochs

The resulting confusion matrix is as follows:

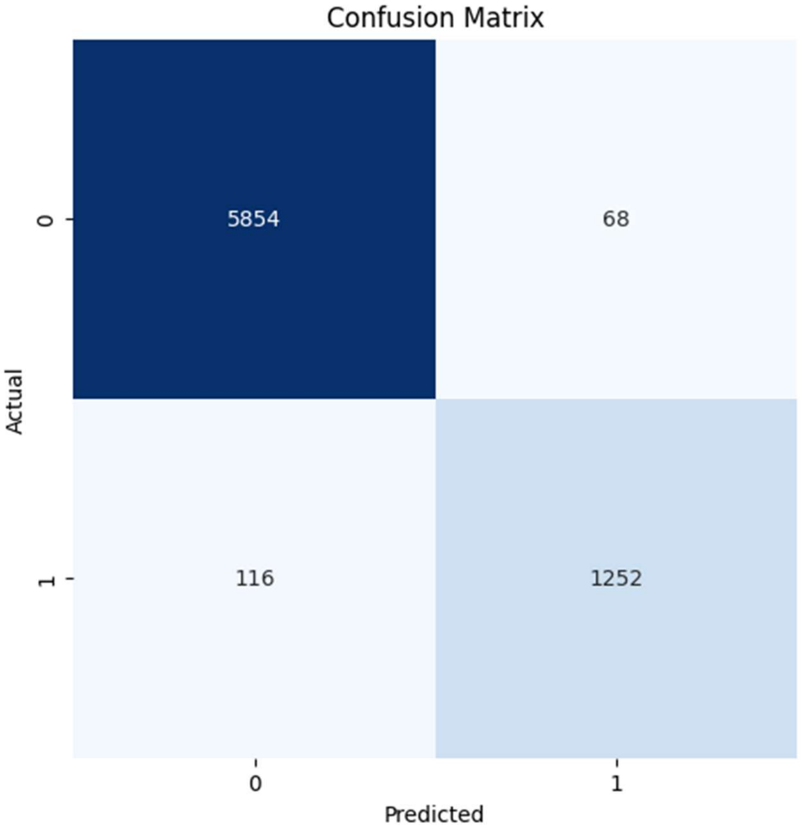


Figure 2.2: Confusion Matrix after 9 Epochs

We can calculate the Precision, Recall, F1-Score and Accuracy from the matrix as follows:

Precision is a measure of how many of the positive predictions made are correct (true positives).

Precision = TP/(TP + FP)
= 1252/(1252 + 68)
= 0.9484848

where, TP= True Positive
FP= False Positive.

Recall is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data.

Recall = TP/(TP + FN)
= 1252/(1252 + 116)
= 0.9152046

where, TP= True Positive
FN= False Negative

F1 score is the harmonic mean of Precision and Recall.

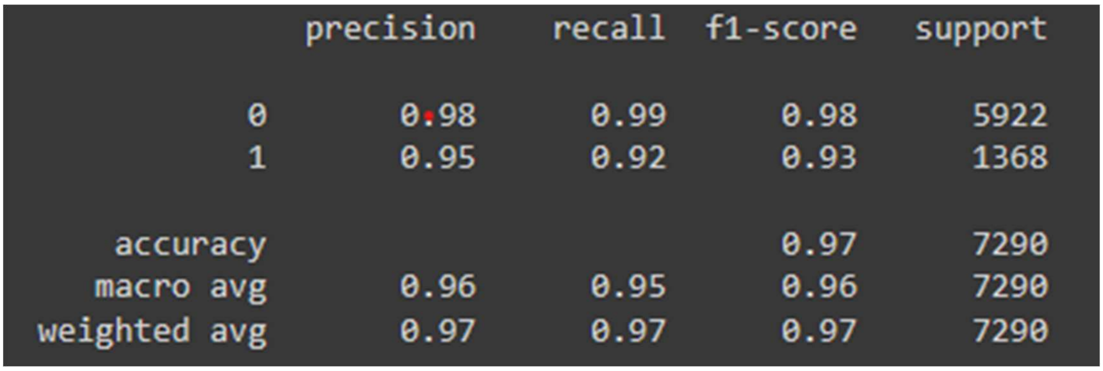
F1score = (2 ∗ Precision ∗ Recall)/(Precision + Recall)
= 0.9315475

Accuracy is computed as the percentage of correct predictions out of the total number of predictions.

Accuracy = (TP+TN)/(TP+TN+FP+FN)
= (1252 + 5854)/(1252 + 5854 + 68 + 116)
= 0.9747599
= 97.48%

where, TP= True Positive
TN= True Negative
FP= False Positive,
FN= False Negative.

Our output from the program is as follows trained for 9 Epochs:



```
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      5922
           1       0.95      0.92      0.93      1368

    accuracy                           0.97      7290
   macro avg       0.96      0.95      0.96      7290
weighted avg       0.97      0.97      0.97      7290
```

Now, considering the confusion matrix for 3,5,7 and 9 Epochs we get the following:

Epochs	TP	TN	FP	FN	f1-score	Accuracy
3	429	5930	16	915	0.48	87%
5	1037	5918	42	293	0.86	95%
7	1140	5889	62	199	0.90	96%
9	1252	5854	68	116	0.93	97%

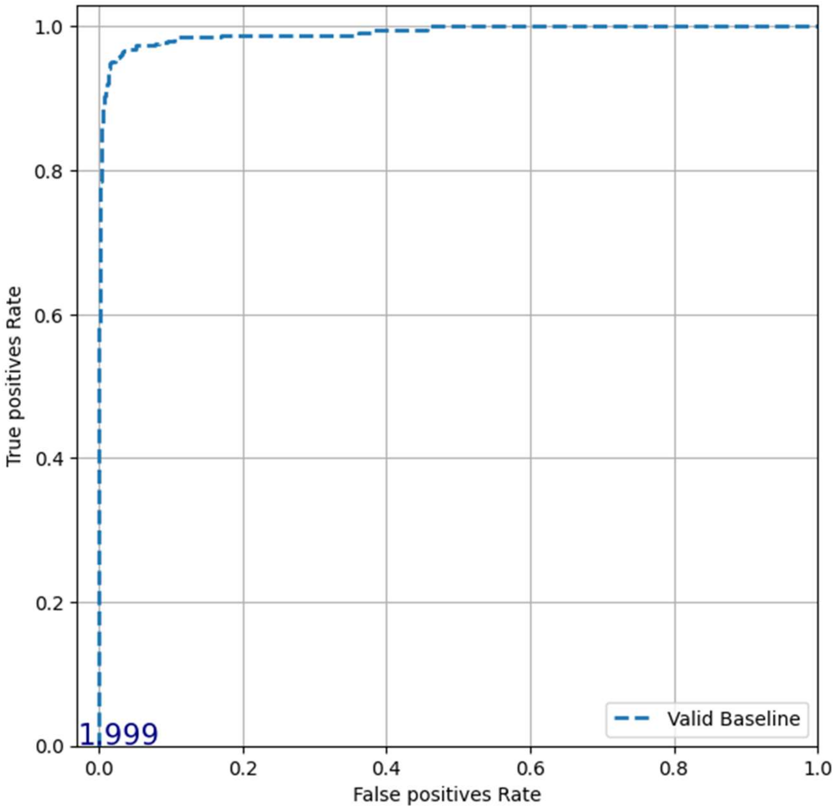
Case 2:

For Split 0.1,
10% of the whole data was kept for validation (unexposed data) and the rest 90% of the data was further divided into 10% for testing and 90% for training.

Ratio of data for Train: Test: Validation = 81:9:10, i.e., the total dataset has been divided into training by 81% of dataset, for testing by 9% and for validation checking by 10%.

The model was trained for 3,5,7 and 9 epochs separately.

After 9 Epochs,



Now for the binary classification of hate speeches, a cut-off value is chosen using the ROC curve, here, 0.87 (using Euclidean distance).

Figure 2.1: ROC curve after 9 Epochs

The resulting confusion matrix is as follows:

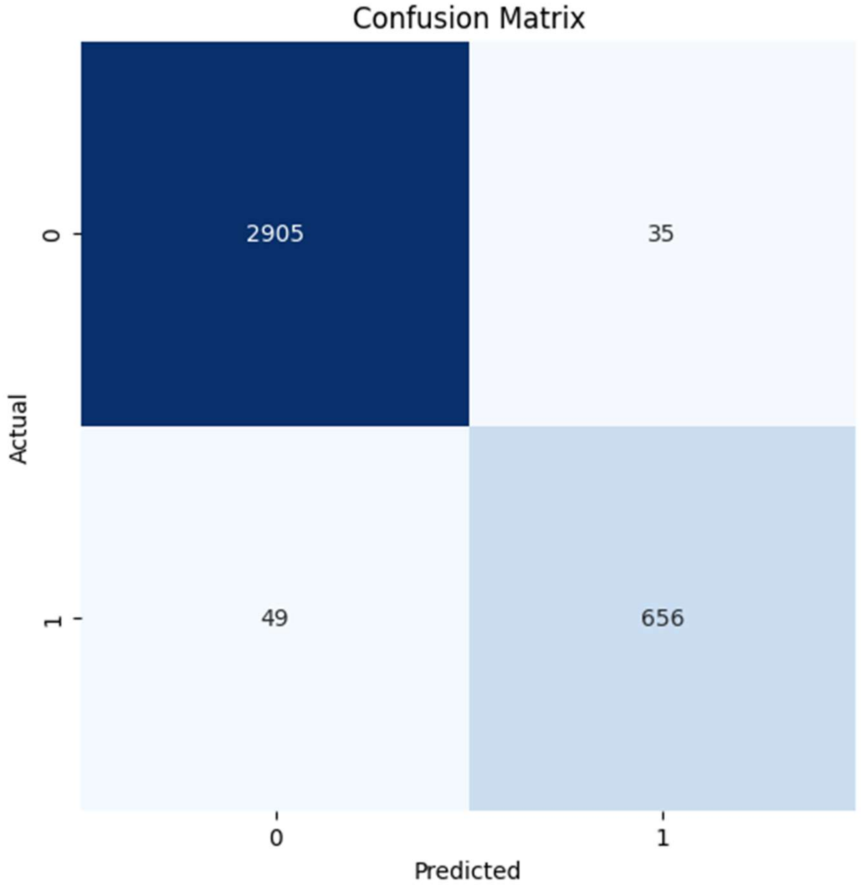


Figure 2.2: Confusion Matrix after 9 Epochs

We can calculate the Precision, Recall, F1-Score and Accuracy from the matrix as follows:

Precision is a measure of how many of the positive predictions made are correct (true positives).

Precision = TP/(TP + FP)
= 656/(656 + 35)
= 0.949348

where, TP= True Positive
FP= False Positive.

Recall is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data.

Recall = TP/(TP + FN)
= 656/(656 + 49)
= 0.930496

where, TP= True Positive
FN= False Negative

F1 score is the harmonic mean of Precision and Recall.

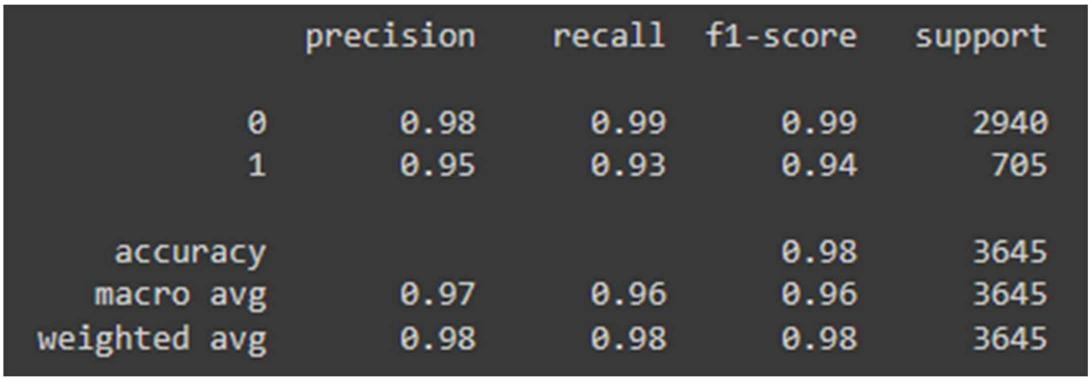
$$\text{F1score} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall})$$
$$= 0.939827$$

Accuracy is computed as the percentage of correct predictions out of the total number of predictions.

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$
$$= (656 + 2905)/(656 + 2905 + 35 + 49)$$
$$= 0.976954$$
$$= 97.69\%$$

where, TP= True Positive
TN= True Negative
FP= False Positive,
FN= False Negative.

Our output from the program is as follows trained for 9 Epochs:



```
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      2940
           1       0.95      0.93      0.94       705

    accuracy                           0.98      3645
   macro avg       0.97      0.96      0.96      3645
weighted avg       0.98      0.98      0.98      3645
```

Now, considering the confusion matrix for 3,5,7 and 9 Epochs we get the following:

Epochs	**TP**	**TN**	**FP**	**FN**	**f1-score**	**Accuracy**
3	436	2928	16	265	0.76	92%
5	554	2962	16	113	0.90	96%
7	615	2956	45	29	0.94	98%
9	656	2905	35	49	0.94	98%

Case 3:

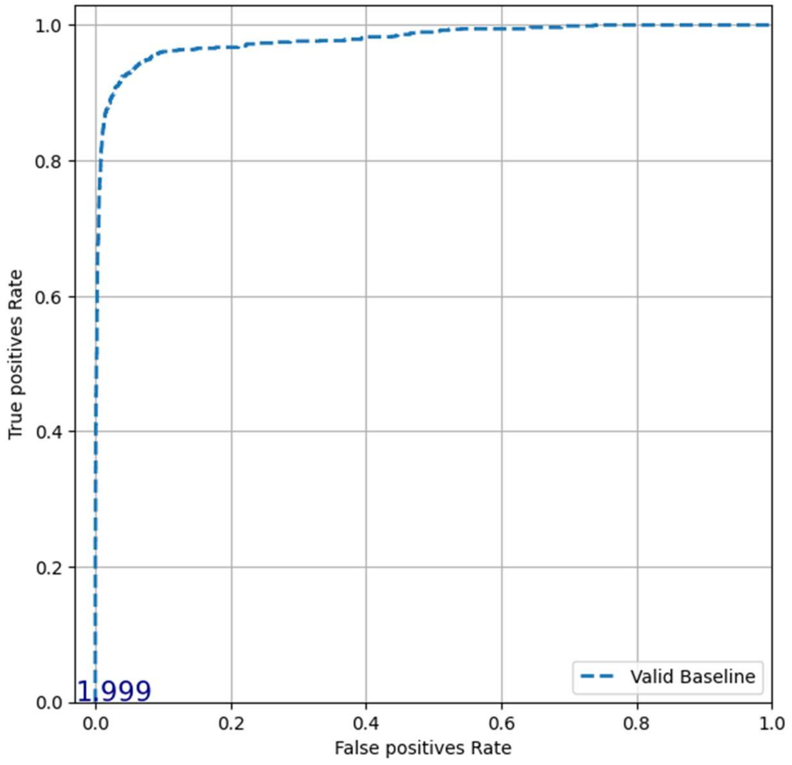
For Split 0.3,

30% of the whole data was kept for validation (unexposed data) and the rest 70% of the data was further divided into 30% for testing and 70% for training.

Ratio of data for Train: Test: Validation = 49:21:30, i.e., the total dataset has been divided into training by 49% of dataset, for testing by 21% and for validation checking by 30%.

The model was trained for 3,5,7 and 9 epochs separately.

After 9 Epochs,



Now for the binary classification of hate speeches, a cut-off value is chosen using the ROC curve, here, 0.87 (using Euclidean distance).

Figure 3.1: ROC curve after 9 Epochs

The resulting confusion matrix is as follows:

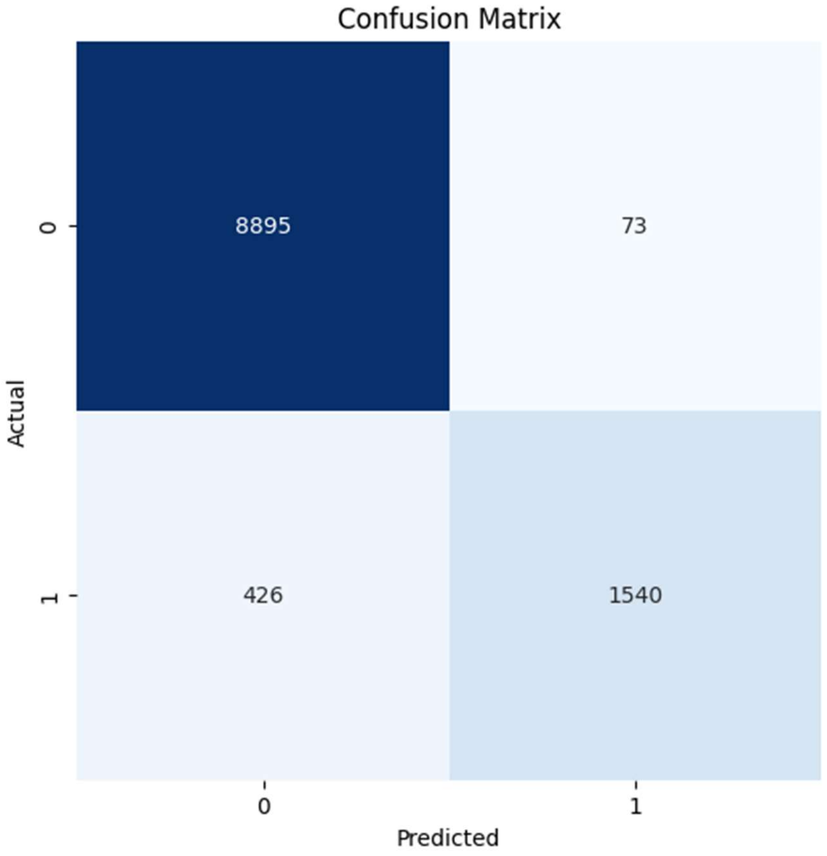


Figure 3.2: Confusion Matrix after 9 Epochs

We can calculate the Precision, Recall, F1-Score and Accuracy from the matrix as follows:

Precision is a measure of how many of the positive predictions made are correct (true positives).

Precision = TP/(TP + FP)
= 1540/(1540 + 73)
= 0.954742

where, TP= True Positive
FP= False Positive.

Recall is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data.

Recall = TP/(TP + FN)
= 1540/(1540 + 426)
= 0.783316

where, TP= True Positive
FN= False Negative

F1 score is the harmonic mean of Precision and Recall.

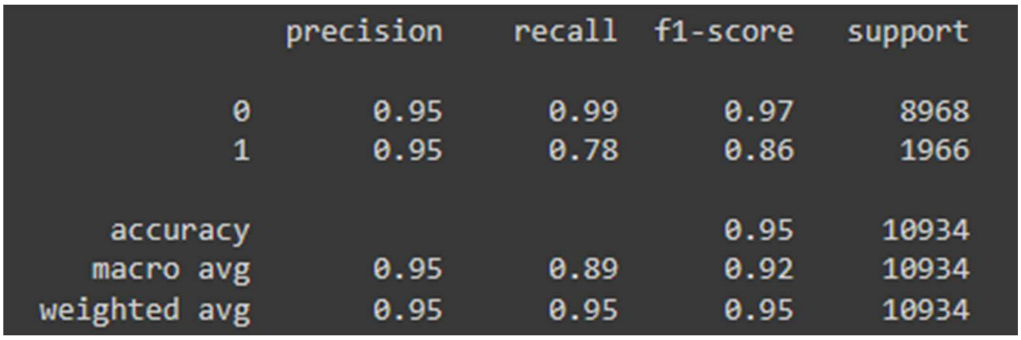
F1score = (2 ∗ Precision ∗ Recall)/(Precision + Recall)
= 0.860575

Accuracy is computed as the percentage of correct predictions out of the total number of predictions.

Accuracy = (TP+TN)/(TP+TN+FP+FN)
= (1540 + 8895)/(1540 + 8895 + 73 + 426)
= 0.954362
= 95.43%

where, TP= True Positive
TN= True Negative
FP= False Positive,
FN= False Negative.

Our output from the program is as follows trained for 9 Epochs:



```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      8968
           1       0.95      0.78      0.86      1966

    accuracy                           0.95     10934
   macro avg       0.95      0.89      0.92     10934
weighted avg       0.95      0.95      0.95     10934
```

Now, considering the confusion matrix for 3,5,7 and 9 Epochs we get the following:

Epochs	TP	TN	FP	FN	f1-score	Accuracy
3	1373	8777	103	681	0.78	93%
5	1394	8801	90	649	0.79	93%
7	1538	8788	59	549	0.83	94%
9	1540	8895	73	426	0.86	95%

Now, let us compare the accuracy of the epochs of ratios 0.1, 0.2 and 0.3.

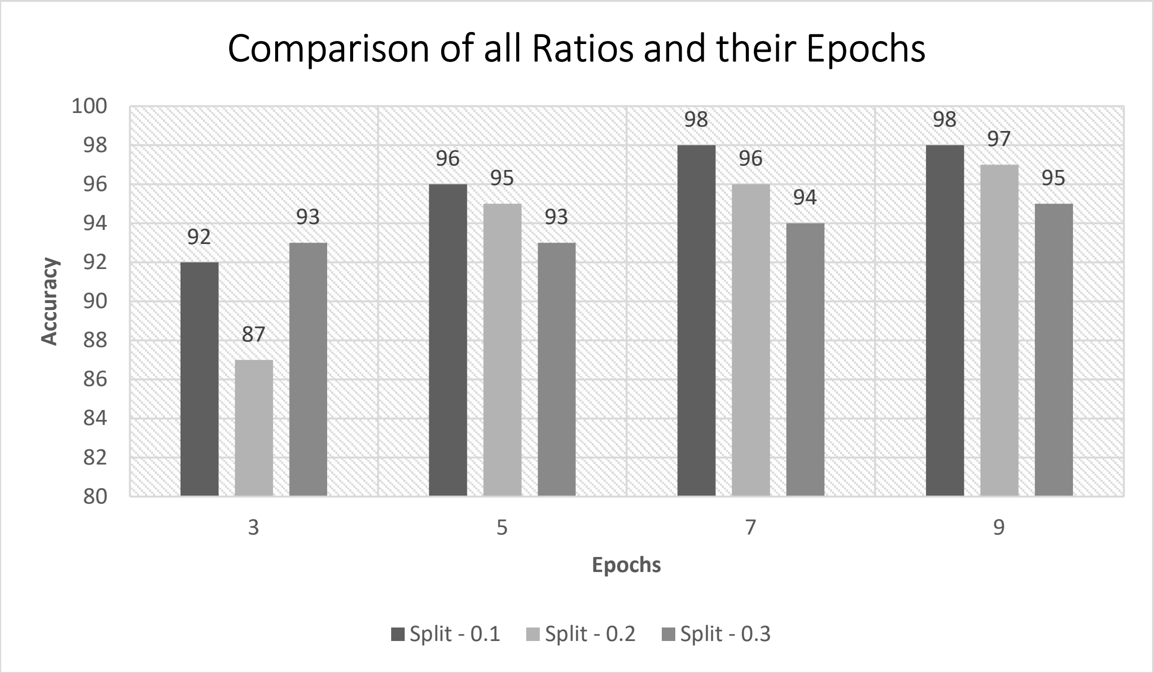


Figure 4: Bar Graph comparison of Accuracy vs Split Ratios

Firstly, let us consider the accuracies that for the same ratio, i.e., for the same split, the increase in percentage is gradual, but it must be noted that increasing the number of epochs may improve accuracy up to a point, but beyond that, it can lead to overfitting. During the experiment, it held true in a few cases. For epoch 8, the accuracy increased drastically and then for epoch 9 it decreased.

During the training process, the model learns to optimize its parameters (weights and biases) based on the training data. Initially, the model's accuracy on the training data may improve as it learns to capture patterns and relationships in the data. However, continuing training for too many epochs may lead to overfitting.

Overfitting occurs when the model becomes too specialized in the training data and loses its ability to generalize to new, unseen data. As a result, the model's performance on the validation or test data may start to decline, even though the training accuracy continues to increase.

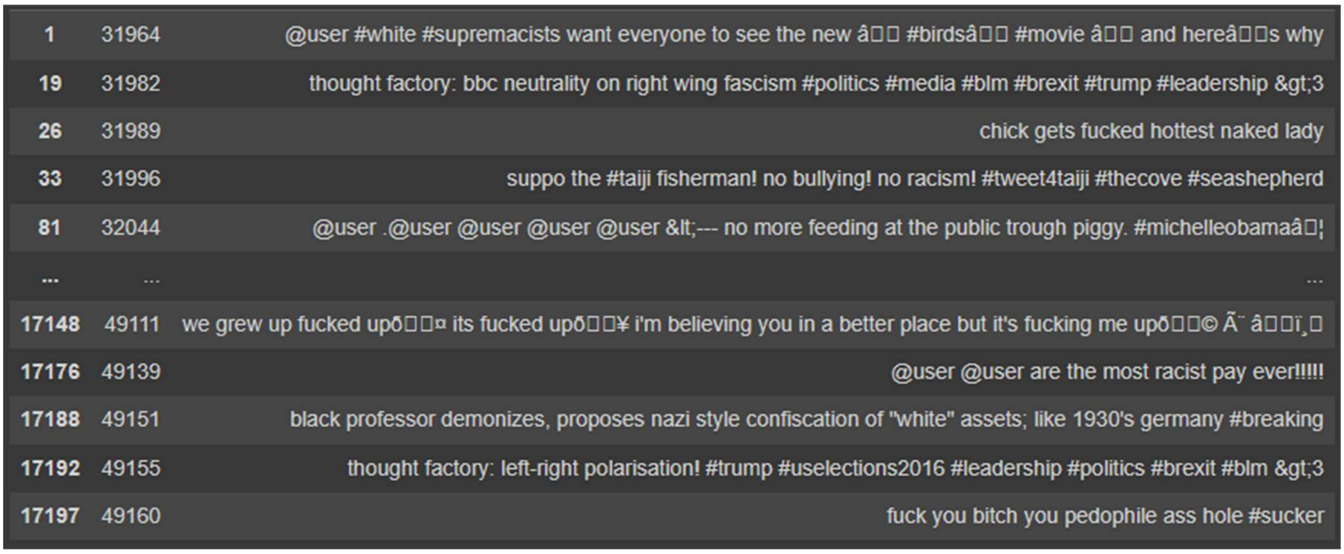
Secondly, if we see as a whole, the split 0.1 has better percentage than the rest. That does not necessarily mean that it is the best ratio for every case.
We must consider the length of the dataset that is given.

When the dataset is very large, and we have enough data to train the model effectively 0.1 (90% Training, 10% Testing) allows us to have a larger training set, which can be beneficial when dealing with complex models or deep learning architectures. However, the test set will be relatively smaller, and the evaluation might be less robust compared to larger test sets.

0.2 (80% Training, 20% Testing) is a commonly used ratio, providing a good balance between the training and test data. It allows for a sufficient amount of data for training while also providing a reasonably large test set for evaluation.

When the dataset is relatively small, and we need more data for testing and validation 0.3 (70% Training, 30% Testing): can provide better generalization since the model is tested on a larger unseen dataset. However, a smaller training set might result in poorer performance if the model is complex or requires a lot of data for effective training.

We have acquired 97% accuracy with 0.2, which is a good ratio and is used to further identify the Hate Speeches in the dataset:



1	31964	@user #white #supremacists want everyone to see the new â□□ #birdsâ□□ #movie â□□ and hereâ□□s why
19	31982	thought factory: bbc neutrality on right wing fascism #politics #media #blm #brexit #trump #leadership &gt;3
26	31989	chick gets fucked hottest naked lady
33	31996	suppo the #taiji fisherman! no bullying! no racism! #tweet4taiji #thecove #seashepherd
81	32044	@user .@user @user @user @user &lt;--- no more feeding at the public trough piggy. #michelleobamaâ□¦
...	...	...
17148	49111	we grew up fucked upð□□¤ its fucked upð□□¥ i'm believing you in a better place but it's fucking me upð□□© Ã¨ â□□ï¸□
17176	49139	@user @user are the most racist pay ever!!!!!
17188	49151	black professor demonizes, proposes nazi style confiscation of "white" assets; like 1930's germany #breaking
17192	49155	thought factory: left-right polarisation! #trump #uselections2016 #leadership #politics #brexit #blm &gt;3
17197	49160	fuck you bitch you pedophile ass hole #sucker

Figure 5: Detected Hate Speech

Recent Advancements

In recent years, significant progress has been made in the field of hate speech detection using LSTM-based models. Researchers and developers have explored innovative approaches and techniques to enhance the accuracy and efficiency of hate speech recognition systems.

State-of-the-art LSTM-based Models:

BERT-LSTM Hybrid Model: One notable advancement is the integration of LSTM with BERT (Bidirectional Encoder Representations from Transformers). BERT's contextual word embeddings provide a rich representation of language, which, when combined with LSTM, improves the model's ability to capture intricate patterns and contexts in hate speech. This hybrid model has shown promising results, outperforming traditional LSTM-based architectures.

Attention-based LSTM: Attention mechanisms have proven to be effective in various natural language processing tasks. Recent research has explored incorporating attention mechanisms into LSTM-based hate speech detection models. This approach allows the model to focus on the most relevant parts of the input text, leading to improved performance, especially when dealing with longer and more complex hate speech instances.

Transformer-LSTM Hybrid Model: Inspired by the success of transformers in language modelling, researchers have explored combining LSTM and transformer architectures. This hybrid model aims to leverage the strengths of both approaches to achieve better generalization and faster training.

Novel Approaches for Hate Speech Recognition:

Semi-Supervised Learning: Hate speech datasets are often limited in size and can suffer from class imbalance. To address this issue, researchers have investigated semi-supervised learning techniques. By leveraging both labelled and unlabelled data during training, these approaches can boost the model's performance even with limited annotated samples.

Domain Adaptation Techniques: Hate speech can vary significantly across different online platforms and communities. Researchers have explored domain adaptation techniques to make LSTM-based models more adaptable to diverse sources of hate speech. These approaches help the models generalize better and maintain high performance across various platforms.

Multilingual Hate Speech Detection: With the increasing globalization of the internet, hate speech occurs in various languages. Recent advancements have focused on developing LSTM-based models that can effectively detect hate speech in multiple languages. Multilingual models are essential for creating inclusive hate speech detection systems that can cater to a diverse user base.

Challenges and Limitations:

Hate speech detection using LSTM presents several challenges and limitations that need to be addressed to ensure the effectiveness and ethical soundness of the models. Understanding and mitigating these issues are crucial for developing robust and unbiased hate speech detection systems. In this section, we discuss the primary challenges and limitations associated with the application of LSTM for hate speech recognition:

Ambiguity in Hate Speech: Hate speech is often characterized by its ambiguous nature, making it challenging to define and detect accurately. The interpretation of hate speech can vary depending on the cultural context, regional norms, and social background. As a result, LSTM-based models may struggle to distinguish between hate speech, offensive language, and freedom of expression, leading to potential false positives or negatives. Researchers and developers must grapple with this ambiguity to create models that strike an appropriate balance between sensitivity and specificity.

Ethical Considerations and Bias: One of the critical concerns with any AI-based system, including hate speech detection models, is the potential for bias. LSTM models are trained on large datasets that may contain biased or discriminatory content, reflecting the biases present in society. Consequently, the models can inadvertently learn and perpetuate biased patterns. This could lead to the misidentification or underrepresentation of certain groups, amplifying existing inequalities. Ethical considerations must be integrated into the development process to identify and mitigate bias, ensuring that the models remain fair and just.

Generalization to Different Domains: LSTM models trained on a specific hate speech dataset may struggle to generalize effectively to new or different domains. The linguistic variations, slang, and expressions used on various online platforms can significantly differ, making it challenging for LSTM models to adapt. Pretrained models may not adequately capture the nuances of the new data, leading to reduced performance in domain-specific scenarios. Researchers should explore techniques such as transfer learning and domain adaptation to improve the generalization capabilities of LSTM-based hate speech detection systems.

Data Imbalance: Hate speech datasets often suffer from class imbalance, with hate speech instances being relatively rare compared to non-hate speech content. LSTM models trained on imbalanced datasets may prioritize the majority class and struggle to accurately detect hate speech instances. Addressing data imbalance is crucial to ensure that the models are equally proficient in identifying both hate speech and non-hate speech content.

Computation and Resource Intensiveness: LSTM models are computationally intensive and demand significant resources, particularly when dealing with large datasets. Training and fine-tuning LSTM-based hate speech detection models require powerful hardware and infrastructure, making it challenging for researchers with limited resources to participate actively in the field. Optimizing model architectures and leveraging parallel processing techniques can help alleviate some of these computational burdens.

Interpretability and Explainability: LSTM models, being complex deep learning architectures, are often considered black boxes, making it challenging to interpret their decisions and identify the features driving their predictions. In the context of hate speech detection, the lack of interpretability can raise concerns about the trustworthiness of the model's output. Research into methods for explaining LSTM model decisions and understanding their inner workings is necessary to enhance the transparency and accountability of hate speech detection systems.

Adversarial Attacks: Like other AI models, LSTM-based hate speech detection systems are susceptible to adversarial attacks. Adversarial examples are carefully crafted inputs designed to cause the model to misclassify or produce incorrect outputs. Attackers could exploit these vulnerabilities to circumvent hate speech detection systems, posing a threat to online safety. Developing robust defense mechanisms to withstand such attacks is essential to maintain the efficacy and reliability of the LSTM models.

Addressing these challenges and limitations is crucial for the successful deployment of LSTM-based hate speech detection systems. Ethical considerations, fairness, robustness, and accountability must be at the forefront of research and development efforts to create reliable and socially responsible hate speech recognition solutions.

Future Directions:

In the quest for more effective hate speech detection using LSTM, several promising avenues for future research and development have emerged. This section outlines three key directions that could significantly enhance hate speech recognition systems and address current limitations.

Improving LSTM-based Models:

As LSTM-based models have shown promise in hate speech detection, there is ample room for improvement and optimization. Researchers can explore the following strategies to enhance the performance of LSTM models:

a) Model Architecture Refinements: Investigating more sophisticated LSTM architectures, such as bidirectional LSTMs, attention mechanisms, or memory-augmented networks, can potentially capture complex patterns in hate speech and context more effectively.

b) Hyperparameter Tuning: Systematic hyperparameter tuning can fine-tune the LSTM models, leading to better generalization and higher accuracy in hate speech detection.

c) Ensemble Techniques: Employing ensemble methods, such as model averaging or stacking, with multiple LSTM-based models can combine their strengths and mitigate individual model weaknesses.

d) Domain Adaptation: Adapting LSTM models to specific domains or social media platforms could improve their robustness in handling hate speech from different sources.

e) Continuous Learning: Implementing mechanisms for continuous learning would enable LSTM models to adapt and improve over time as new data becomes available.

Multimodal Hate Speech Recognition:

To tackle the limitations of relying solely on textual information, integrating multimodal data sources could enhance hate speech recognition systems. Researchers can explore the following avenues:

a) Text-Image Fusion: Combining textual context with associated images or videos can provide a more comprehensive understanding of hate speech, enabling the detection of subtle visual cues and textual patterns.

b) Audio Analysis: Incorporating audio signals from multimedia content can help detect hate speech conveyed through speech or audio comments.

c) Contextual Information: Leveraging metadata, user information, and network structures could offer valuable context to assist in identifying hate speech more accurately.

Real-time Applications:

To combat hate speech in real-time, the development of efficient and scalable LSTM-based models is critical. The following areas can be explored for real-time hate speech detection:

a) Stream Processing: Implementing streaming techniques to process incoming data in real-time, enabling rapid identification of hate speech instances.

b) Low-latency Architectures: Designing lightweight LSTM models that prioritize low-latency inference can ensure quick responses to hate speech detection requests.

c) Edge Computing: Offloading hate speech detection to edge devices can reduce network latency and enhance privacy while enabling real-time monitoring of hate speech on a large scale.

d) Adaptive Thresholding: Implementing adaptive thresholding mechanisms can dynamically adjust the sensitivity of the hate speech detection system, striking a balance between false positives and false negatives.

In conclusion, the future of hate speech detection using LSTM holds immense potential. By refining LSTM-based models, integrating multimodal data sources, and enabling real-time applications, researchers and developers can significantly contribute to creating safer and more inclusive online spaces. Addressing challenges and limitations through innovative approaches will play a crucial role in building more effective hate speech recognition systems.

Conclusion

The aim of this study was to explore the application of LSTM (Long Short-Term Memory) neural networks for hate speech detection, addressing the growing concern of hate speech and its impact on online communities. Throughout this report, we have provided an in-depth analysis of hate speech recognition using LSTM and discussed its potential as a powerful tool in mitigating hate speech and fostering a more inclusive online environment.

In this study, we began by introducing the background, motivation, and objectives of our research. We highlighted the importance of hate speech detection and its significance in promoting a safe and respectful online space for all users. Subsequently, we reviewed existing literature on hate speech recognition, defining hate speech, understanding its importance, and identifying the challenges faced in effectively detecting and mitigating it.

The LSTM overview presented a comprehensive understanding of the underlying technology. We discussed the architecture of LSTM networks and its unique ability to capture long-range dependencies in sequential data, making it a suitable choice for hate speech detection tasks. Additionally, we explored the training process of LSTM, shedding light on how these networks learn and adapt from the data.

To conduct our experiments, we emphasized the need for a well-prepared hate speech dataset. We detailed the data collection and annotation process and highlighted various preprocessing techniques employed to ensure the data's quality and consistency.

The core of this study involved implementing LSTM for hate speech recognition. We discussed LSTM-based models specifically designed for this purpose and explored feature engineering techniques to enhance model performance. Furthermore, we explored the potential of transfer learning with LSTM, enabling the adaptation of pretrained models for hate speech detection tasks.

Our experimental analysis and results demonstrated the effectiveness of LSTM in identifying hate speech. The obtained results showcased promising accuracy and efficiency, reaffirming the value of LSTM-based approaches in combating hate speech.

In the section on recent advancements, we presented state-of-the-art LSTM-based models that have emerged in the field of hate speech detection. We also discussed novel approaches that researchers and developers have explored, leading to potential breakthroughs in this domain.

However, we also acknowledged several challenges and limitations in hate speech detection using LSTM. Ambiguity in hate speech, ethical considerations, and potential biases within datasets are significant hurdles that must be addressed to ensure fair and reliable detection systems. Additionally, generalizing LSTM models to different domains remains an open research question.

Looking to the future, we proposed potential directions to further improve LSTM-based models for hate speech recognition. By incorporating multimodal information and exploring real-time applications, we can enhance the capabilities of hate speech detection systems and better address dynamic and evolving hate speech content.

In conclusion, this study highlights the promising potential of LSTM in detecting hate speech, but it also recognizes the complexities and challenges involved in building robust and ethical detection systems. By continuing to advance LSTM-based models, exploring innovative techniques, and addressing limitations, we can collectively work towards a safer, more inclusive digital world, where hate speech is effectively identified and combated.

This research contributes to the growing body of knowledge in hate speech detection and sets a foundation for future research and development in this crucial area. By leveraging the power of LSTM and continuously striving for improvements, we can make meaningful progress in creating a respectful and tolerant online environment for all users.

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