SOMAIYA
VIDYAVIHAR UNIVERSITY

Sentiment AI: Twitter Sentiment Analysis

Submitted In Partial Fulfillment of Requirements
For the Degree Of

Bachelor of Technology
(Electronics Engineering)

By

Ehsaas Sam

Roll No: 16010220050

Khushi Choudhary

Roll No: 16010220052

Guide :

Prof. Sushma Kadge

Co-Guide :

Dr. Sudha Gupta

SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya Vidyavihar University
Vidyavihar, Mumbai - 400 077
2020-24

Somaiya Vidyavihar University

K. J. Somaiya College of Engineering

Certificate

This is to certify that the dissertation report entitled **Sentiment AI: Twitter Sentiment Analysis** submitted by Ehsaas Sam and Khushi Choudhary at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree Bachelor of Technology (Electronics Engineering) of Somaiya Vidyavihar University

____________________ _________________________

Guide Head of the Department

____

Principal

Date:

Place: Mumbai-77

Somaiya Vidyavihar University

K. J. Somaiya College of Engineering

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

______________________ **Signature of the Student** **16010220050** **Roll No.**	______________________ **Signature of the Student** **16010220052** **Roll No.**

Date:

Place: Mumbai-77

Abstract

The Twitter Sentiment Analysis project seeks to understand popular opinion, trends, and attitudes on a range of topics by analyzing sentiments expressed in tweets. By leveraging natural language processing and machine learning techniques, this project extracts sentiment polarity (positive, negative, or neutral) from Twitter data to understand user sentiments towards specific subjects, brands, or events.

The main goal of the project is to create and put into use a sentiment analysis system that can instantly evaluate massive amounts of Twitter data. The procedure entails gathering data from the Twitter API, cleaning and tokenizing text data through preprocessing, extracting features to numerically represent tweets, and training models with deep learning algorithms including long short-term memory (LSTM) networks and convolutional neural networks (CNN).

Preprocessing of the data, model architecture design, training and assessment processes, and result interpretation are important parts of the sentiment analysis system. The system's objectives are to give businesses, marketers, researchers, and legislators useful information so they may monitor brand reputation, make educated decisions, and gauge public opinion on certain topics or events.

The Twitter Sentiment Analysis project delivers a scalable, accurate, and efficient solution for sentiment analysis on Twitter data, which advances the field of social media analytics. It encourages the use of social media data in a variety of fields, including public opinion research, crisis monitoring, brand management, and market research.

Key words: Twitter, Sentiment Analysis, Natural Language Processing, Machine Learning, Social Media Analytics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter represents the basic and knowledge of Twitter Sentiment Analysis and provides us with the overview of our report.

Social media sites such as Twitter, which were once only tools for communication, have evolved into dynamic ecosystems in the digital age where people interact in a variety of ways with each other, their organizations, and communities. Particularly on Twitter, people have made a name for themselves as global forums for opinion sharing, experience sharing, and public discourse influence. Twitter functions as a virtual agora where ideas are shared, debates develop, and trends emerge quickly due to its real-time nature and large user base.

Twitter's enormous amount of user-generated content creates a mosaic of human sentiments and emotions that provides a broad glimpse into society's collective psyche. Twitter captures the zeitgeist, reflecting the dominant mood, attitudes, and perceptions of individuals and communities alike, from the joyous celebrations of triumph to the somber reflections in times of crisis. Tweets are a powerful tool for expressing a wide range of emotions, from happiness and excitement to frustration and hopelessness.

More than just a platform for personal expression, Twitter is a virtual agora where ideas collide, public opinion is formed, and stories are told in real time. Retweets amplify voices that resonate across continents, memes transcend language barriers to convey cultural phenomena, and hashtags become rallying cries for social movements. The lines separating the private and public spheres become more hazy in this digital agora as users traverse the terrain of virtual social interaction, building communities around common identities and interests.

One cannot stress the importance of Twitter as a gauge of public opinion. Twitter data provides invaluable insights that inform decision-making processes across sectors, from predicting market fluctuations and gauging political sentiment to monitoring brand perception and tracking consumer trends. Companies use sentiment analysis on Twitter to improve marketing tactics, reduce risks to their reputation, and foster customer loyalty. Scholars utilize Twitter data to examine social phenomena, investigate group dynamics, and investigate the dynamics of information dissemination in the digital era.

Twitter, which enables people to amplify their voices, shape narratives, and join in on the global conversation, embodies the democratization of discourse in the digital age. We dive into the depths of this digital agora as we set out to investigate sentiment analysis on Twitter, figuring out the subtleties of sentiment and the complexities of human expression as well as the insights that lurk beneath the surface of the tweetstream.

1.1 The Importance of Twitter Sentiment Analysis:

Twitter is a global platform that reflects varied opinions on a wide range of topics, from politics and entertainment to companies and social issues. Its real-time nature makes it an ideal microcosm of public mood. Understanding public opinion dynamics, spotting new trends, and evaluating the success or failure of advertising campaigns, product launches, or political movements all depend heavily on sentiment analysis on Twitter. Businesses can learn about customer happiness, identify possible crises, and adjust their strategy to suit client preferences by examining the sentiments conveyed in tweets. In a similar vein, decision-makers can use public opinion research to guide their choices and successfully handle social issues. Moreover, sentiment analysis on Twitter contributes to academic research by enabling the study of collective behavior, sentiment contagion, and the impact of events on public sentiment.

1.2 Difficulties in Interpreting Sentiment on Twitter:

Even with the abundance of data on Twitter, sentiment analysis is not without its difficulties. Sentiment identification is a challenging task because of the informal and concise style of tweets, which are commonly marked by emoticons, slang, and acronyms. It could be difficult for conventional natural language processing (NLP) methods to extract the complex context and meanings that are buried in tweets. Sentiment research is further complicated by the dynamic nature of language and the changing usage of memes and emojis. In addition, scalable and effective algorithms for sentiment analysis are required due to the huge amount of tweets that are generated every second. To properly handle the inflow of data, real-time processing and classification of tweets require a strong computational infrastructure and optimization approaches.

1.3 Applying Deep Learning to Sentiment Analysis on Twitter:

Deep learning models have transformed natural language processing in recent years by providing previously unheard-of capacities for comprehending and analyzing textual input. Particularly, Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) have drawn interest due to their superior performance in sentiment analysis applications. CNNs are excellent at identifying spatial linkages and local patterns in text, which makes them a good choice for sentiment analysis feature extraction. However, long-range relationships and sequential data modeling are strong points of LSTMs, which makes it possible for them to extract temporal dynamics and contextual information from tweets. Researchers and practitioners can create strong sentiment analysis models that can reliably identify sentiments from tweets by utilizing the advantages of CNNs and LSTMs.

1.4 Objectives of the Report:

In light of this, this paper sets out to investigate Twitter sentiment analysis through the application of cutting-edge deep learning algorithms. It seeks to explore the nuances of CNNs and LSTMs, how they are used in sentiment analysis, and how well they can extract sentiment from tweets. The research looks at case studies, techniques, and performance evaluations in an effort to offer insightful information about the benefits and drawbacks of using deep learning models for Twitter sentiment analysis. In addition, the report will investigate methods for improving the resilience and accuracy of sentiment analysis on Twitter data through feature engineering, data preprocessing, and model evaluation. Ultimately, the report endeavors to equip stakeholders with the knowledge and tools necessary to harness the power of Twitter data in understanding and leveraging public sentiment effectively for various applications.

CHAPTER 2

Overview of Sentiment Analysis

This chapter represents the overview of Sentiment Analysis including methods, challenges and it's applications.

2.1 Introduction to Sentiment Analysis:

In the twenty-first century, textual data is being generated and distributed on an unprecedented scale across several internet platforms in a fast-paced, highly interconnected digital ecosystem. Every second, millions of text-based exchanges take place, influencing consumer behavior, public discourse, and market dynamics with everything from product evaluations and social media posts to news stories and customer feedback. In this setting, it is becoming more and more important for people, businesses, and organizations in a variety of industries to be able to understand and analyze the underlying feelings, emotions, and attitudes portrayed within this enormous sea of textual information.

Sentiment analysis, or opinion mining, is the vanguard of this effort. It is a subfield of natural language processing (NLP) that is specifically concerned with automating the extraction, measurement, and understanding of opinions that are contained in textual data. Sentiment analysis employs a blend of computer approaches, machine learning algorithms, and linguistic analysis to identify the underlying subjective opinions, attitudes, and emotional tones expressed in written language.

2.2 Importance of Sentiment Analysis:

Sentiment analysis is a potent technique that has broad applications in business, marketing, politics, finance, healthcare, and social sciences. It has become more and more prominent in recent years. Several important aspects contribute to its significance:

1. Consumer Insights and Market Intelligence: Businesses may learn a great deal about the preferences, attitudes, and satisfaction levels of their customers by examining customer reviews, social media chats, and online forums. As a result, they are more equipped to customize their goods, services, and marketing plans to their target market's requirements and expectations.

2. Brand Reputation Management: By tracking mentions, sentiment trends, and consumer feedback in real-time, sentiment analysis enables businesses to keep an eye on and manage their online reputation. Through swift resolution of negative feedback and amplification of positive comments, companies may protect their reputation, build consumer confidence, and cultivate brand advocacy.

3. Identification of Risks and Crisis Handling: Companies can take proactive steps, reduce risks, and maintain stakeholder confidence when they identify unfavorable sentiment, new difficulties, or impending crises early on. Organizations can detect potential reputational risks and take appropriate

action before they become major issues by keeping an eye on social media mood, news headlines, and public opinion.

4. Political and Social Analysis: Sentiment analysis helps governments, policymakers, and scholars understand public opinion, political discourse, and society trends. It also helps them determine how the public feels about certain policies, candidates, and social concerns. This makes it possible to make well-informed decisions, practice efficient governance, and develop evidence-based policies that meet the needs and desires of the general public.

2.3 Methods and Techniques:

Sentiment analysis is a broad field with many different approaches and strategies, all of which have their own benefits and capacities for identifying and analyzing the underlying feelings included in textual data. Sentiment analysis uses a broad range of techniques to extract complex insights from text, from conventional lexicon-based approaches to cutting-edge deep learning systems. These are some of the main approaches and strategies that are frequently used in sentiment analysis:

1. Approaches Based on Lexicology:

Lexicon-based techniques rely on pre-established dictionaries or sentiment lexicons that have lists of words marked with the positive, negative, or neutral polarity that corresponds to each word. These lexicons are used as reference materials for sentiment analysis activities and are either manually or automatically curated. The overall sentiment orientation of the text is then ascertained by analyzing the presence, frequency, and context of words that convey sentiment in the textual data. Lexicon-based techniques are easy to understand and relatively simple, but they may have trouble with complex expressions, meanings that change depending on the context, and concepts that are not part of common language.

2. Machine Learning Models:

Machine learning approaches are essential to sentiment analysis because they provide strong instruments for automatically classifying, regressing, and clustering text data according to sentiment labels. Support vector machines (SVM), Naive Bayes, Decision Trees, and logistic regression are supervised learning methods that are frequently used to train models using labeled datasets, in which each text occurrence is assigned a sentiment category (e.g., positive, negative, or neutral). By learning to identify linguistic signals, patterns, and other characteristics that indicate sentiment, these models are able to generalize and predict outcomes based on data that has not yet been observed. Supervised learning techniques are flexible, scalable, and capable of capturing intricate relationships in text, but they do require annotated training data.

3. Deep Learning topologies:

By enabling the creation of complex neural network topologies that can capture nuanced semantics and contextual relationships in text data, deep learning has completely changed the field of sentiment research. Sentiment analysis tasks have shown remarkable performance from Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term

Memory (LSTM) networks, and Transformer-based models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer). By automatically learning hierarchical representations of text, these models make use of deep neural networks to capture sentiment-related features, grammatical structures, and semantic linkages. Deep learning architectures are excellent at handling large-scale datasets, noisy text, and domain-specific language because they can handle text at various levels of abstraction. This makes them well-suited for a variety of sentiment analysis applications.

2.4 Challenges in Sentiment Analysis:

Despite its many uses and benefits, sentiment analysis is not without its difficulties. Improving the precision, dependability, and resilience of sentiment analysis systems requires addressing these issues. The following are some of the main difficulties in sentiment analysis:

1. <u>Subjectivity and Ambiguity</u>:

Language is by its very nature subjective, and context, cultural background, and individual viewpoints can all have a significant impact on how feeling is understood. Sentences with irony, sarcasm, or other nuanced language may be misconstrued by sentiment analysis algorithms, producing unreliable findings. Sentiment analysis still faces significant challenges in handling linguistic ambiguity and determining the underlying purpose behind text.

2. <u>Data Quality and Bias</u>:

The effectiveness of sentiment analysis models is greatly influenced by the caliber of the training data. An effective prediction and generalization of the model can be severely hampered by biases, errors, and noise in the data. Furthermore, models that are biased in favor of the majority class might be created by imbalances in the dataset's sentiment class distribution, which can distort the results and lower performance. In sentiment analysis research, addressing problems with data quality and reducing biases in training data are ongoing concerns.

3. <u>Contextual Understanding</u>:

Since the same words or phrases may have multiple meanings depending on the context, context is crucial for proper sentiment analysis. Slang, colloquial idioms, domain-specific vocabulary, and cultural allusions make sentiment analysis even more difficult. Effectively capturing and modeling context is still quite difficult, especially in dynamic or changing situations where language usage and mood standards are subject to change over time.

4. <u>Multimodal Sentiment Analysis</u>:

As multimedia content on social media and the internet continues to grow, sentiment analysis is becoming more and more necessary to examine sentiment conveyed in a variety of modalities, such as text, photos, videos, and audio. Cross-modal semantics, feature fusion, and alignment are some of the particular difficulties that arise when integrating data from many modalities. Research on efficient methods for multimodal sentiment analysis is ongoing, and the results have important practical consequences.

5. Domain Adaptation and Transfer Learning:

Due to variations in language usage, topic-specific terminology, and sentiment distributions, sentiment analysis models trained on one domain or dataset may not translate effectively to different domains or contexts. In order to address this issue, domain adaptation and transfer learning strategies make use of source domain knowledge to enhance performance in target domains with a dearth of labeled data. Sentiment analysis still faces a challenging issue when it comes to transferring models to new domains while maintaining their efficacy and preventing negative transfer.

2.5 Applications of Sentiment Analysis:

Sentiment analysis finds several applications in a wide range of areas due to its capacity to extract, measure, and evaluate sentiments represented in text data. The following are some important uses for sentiment analysis:

1. Social Media Monitoring:

Social media sites have become indispensable conduits for engagement, expression, and communication. Businesses, organizations, and people can watch brand sentiment, detect new trends, keep an eye on social media conversations, and assess the effectiveness of marketing activities by using sentiment analysis. Sentiment analysis, which examines user-generated content like tweets, postings, comments, and reviews, offers insightful information on consumer sentiment, public opinion, and market perception.

2. Customer Reviews and Feedback:

In the age of online shopping and e-commerce, consumer reviews and feedback are extremely important in influencing consumers' perceptions of brands and their decisions to buy. Businesses can use sentiment analysis to evaluate and classify consumer reviews, ratings, and feedback in order to better understand the sentiment behind the product, pinpoint its advantages and disadvantages, and obtain useful information for raising customer satisfaction and improving the product. Sentiment analysis makes data-driven decision-making easier and expedites the feedback analysis process by automatically processing massive amounts of textual data.

3. Financial Market Analysis:

For financial market analysis and making investment decisions, sentiment analysis has proven to be a useful tool. Sentiment analysis assists traders, investors, and financial institutions in determining market sentiment, forecasting trends, and evaluating investment opportunities by examining sentiment expressed in news stories, social media, analyst reports, and financial statements. Sentiment analysis is used by sentiment-driven algorithms, sentiment indexes, and sentiment-based trading methods to take advantage of market sentiment signals and obtain an advantage over competitors in the financial markets.

4. Political Opinion Mining:

Sentiment analysis is a tool used by governments, political analysts, and policymakers to assess public opinion, sentiment toward political candidates, policy concerns, and election results. Sentiment analysis gives information on voter sentiment, political discourse, and electoral dynamics through examining social media conversations, news articles, and public speeches. Sentiment analysis is a tool used by political campaigns to determine public opinion, customize content, and create focused campaign tactics that will sway voters and increase their chances of winning.

5. Brand Reputation Management:

It's critical for companies and organizations to keep a positive brand image and reputation. Sentiment analysis assists brands in tracking online conversations, locating brand mentions, sentiment, and sentiment drivers, and quickly handling complaints, concerns, and feedback from customers. Through proactive brand sentiment management and customer service, firms may foster brand loyalty, establish credibility, and reduce reputational threats.

2.6 Conclusion:

Sentiment analysis has become indispensable for extracting valuable insights from textual data, empowering stakeholders across various domains to make informed decisions and drive positive outcomes. Despite challenges like subjectivity and data quality issues, ongoing advancements in NLP and machine learning are enhancing the accuracy and scalability of sentiment analysis models.

Looking ahead, sentiment analysis holds immense promise, with further innovations expected to revolutionize decision-making processes. By leveraging sentiment analysis alongside other AI-driven technologies, organizations can navigate the complexities of the digital landscape with confidence and drive success in an increasingly data-driven world.

Chapter 3

Literature Survey

This chapter represents various survey, research, development, outcomes and findings of different papers done on Twitter Sentiment Analysis.

Title	Authors	Publishers	Year	Abstract
Feature-Based Twitter Sentiment Analysis With Improved Negation Handling	Itisha Gupta, Nisheeth Joshi	IEEE Transactions on Computational Social Systems	2021	This article introduces a feature-based approach to Twitter sentiment analysis (TSA), incorporating enhanced negation handling. Leveraging various features such as lexicon-based, morphological, POS-based, and n-gram features, our proposed system impacts polarity determination significantly. We experiment with three state-of-the-art classifiers—support vector machine (SVM), Naive Bayesian, and decision tree—to identify optimal classifier-feature group combinations. Additionally, we investigate negation, developing an algorithm to handle negation tweets effectively. Evaluation on the SemEval-2013 Task 2 benchmark dataset shows SVM's superiority and the effectiveness of our negation handling strategy.
Real Time Sentiment Analysis Of Twitter Posts	Prakruthi V, Sindhu D, Dr. S Anupama Kumar	IEEE Conference on Computational Systems	2018	Sentiment analysis involves processing natural language to recognize affective states and subjective information. Twitter, a popular social media platform, serves as a hub for expressing

				opinions and sentiments. This paper evaluates sentiment about individuals, trends, products, or brands by analyzing tweets. Utilizing the Twitter API, tweets are directly accessed to build sentiment classification. Results are visualized using techniques like histograms and pie charts. Sentiment Analysis of Lockdown in India During COVID-19: A Case Study on Twitter Prasoon Gupta, Sanjay Kumar, R. R. Suman, and Vinay Kumar IEEE Transactions on Computational Social Systems August 2021 This study delves into sentiment analysis of tweets regarding the nationwide lockdown in India during the COVID-19 pandemic. Utilizing natural language processing (NLP) and machine learning classifiers, sentiments from tweets containing the keyword "Indialockdown" were analyzed. A total of 12,741 tweets from April 5, 2020 to April 17, 2020 were extracted and processed using TextBlob, VADER lexicons, and various classifiers. The highest accuracy of 84.4% was achieved with the LinearSVC classifier and unigrams, indicating widespread support for the government's lockdown decision.
The Effect of Preprocessing Techniques on	Akrivi Krouska, Christos	IEEE Transactions on Computational	2022	Twitter serves as a rich source for expressing diverse thoughts and opinions, making it valuable for

Twitter Sentiment Analysis	Troussas, Maria Virvou	Social Systems		sentiment analysis. However, effective data preprocessing is crucial for accurate sentiment classification. This paper explores various preprocessing methods to preprocess reviews for sentiment analysis, discussing the impact of preprocessing on sentiment polarity classification methods for Twitter text. Through experiments on manually annotated Twitter datasets, the study evaluates different preprocessing combinations and their efficiency. The results demonstrate the positive influence of feature selection and representation on classification performance.
Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter	Meylan Wongkar, Apriandy Angdresey	ICIC Conference	2019	This study conducts sentiment analysis on tweets related to the 2019 Republic of Indonesia presidential candidates using the Naive Bayes algorithm. The sentiment analysis application, implemented in Python, collects data, processes text, trains and tests data, and classifies sentiments. Results reveal positive and negative sentiment scores for each presidential candidate, with an overall accuracy of approximately 80.1%. Additionally, the study compares the Naive Bayes algorithm with SVM and K-Nearest Neighbor (K-NN) methods, demonstrating superior accuracy with Naive Bayes.
SentiDiff:	Lei Wang,	IEEE	2020	This paper addresses the challenge

Combining Textual Information and Sentiment Diffusion Patterns for Twitter Sentiment Analysis	Jianwei Niu, Shui Yu	Transactions on Knowledge and Data Engineering		of Twitter sentiment analysis by integrating textual information with sentiment diffusion patterns. Leveraging sentiment diffusion patterns, particularly sentiment reversals, we propose an iterative algorithm called SentiDiff to predict sentiment polarities expressed in Twitter messages. Our approach represents the first attempt to incorporate sentiment diffusion patterns for enhancing Twitter sentiment analysis. Experimental results on real-world datasets demonstrate significant improvements in sentiment classification compared to existing textual information-based methods.
Sentiment Analysis of Lockdown in India During COVID-19: A Case Study on Twitter	Prasoon Gupta, Sanjay Kumar, R. R. Suman, and Vinay Kumar	IEEE Transactions on Computational Social Systems	2021	This study delves into sentiment analysis of tweets regarding the nationwide lockdown in India during the COVID-19 pandemic. Utilizing natural language processing (NLP) and machine learning classifiers, sentiments from tweets containing the keyword "Indialockdown" were analyzed. A total of 12,741 tweets from April 5, 2020 to April 17, 2020 were extracted and processed using TextBlob, VADER lexicons, and various classifiers. The highest accuracy of 84.4% was achieved with the LinearSVC classifier and unigrams, indicating widespread support for the government's lockdown decision.

Table 3.1 Literature Survey

Chapter 4

Components of Sentiment Analysis

This chapter represents the components used in Sentiment Analysis Models.

Sentiment analysis, often referred to as opinion mining, stands at the intersection of various disciplines, offering a comprehensive approach to extracting and deciphering sentiments, emotions, and viewpoints embedded within textual data. This multifaceted field is pivotal in understanding the nuanced expressions of individuals across diverse contexts. To adeptly unravel the sentiments encapsulated within textual content, sentiment analysis leverages a series of pivotal components, each meticulously crafted to contribute towards the overarching objective with precision and efficacy. These components synergistically collaborate, forming a robust framework that enables the systematic exploration and interpretation of sentiment, thereby empowering analysts to glean invaluable insights from the vast ocean of textual information.

4.1 Data Preprocessing

Data preprocessing is crucial for cleaning and preparing textual data before analysis. Here's a detailed explanation of each step:

1. Special Character Removal:

Special characters, such as punctuation marks, symbols, and emoticons, are often irrelevant for sentiment analysis and can introduce noise into the data. Removing these characters ensures that only meaningful text remains.

2. Lowercasing:

Standardizing the text to lowercase helps in treating words with different cases (e.g., "Good" vs. "good") as the same, reducing the vocabulary size and ensuring consistency in analysis.

3. Numerical Digit Removal:

Numerical digits are typically not indicative of sentiment and can be safely removed. However, in some cases, such as sentiment analysis of product reviews containing ratings, numerical digits might carry sentiment information and may be retained.

4. Link Removal:

URLs or hyperlinks present in the text are usually irrelevant for sentiment analysis and can be removed to focus on the textual content.

5. Tokenization:

Tokenization involves breaking down the text into individual words or tokens. This process facilitates further analysis by treating each word as a separate entity. Common tokenization techniques include word-level tokenization and subword-level tokenization.

6. Stopword Removal:

Stopwords are commonly occurring words (e.g., "the," "is," "and") that do not carry significant meaning in sentiment analysis. Removing stopwords helps in reducing noise and focusing on sentiment-bearing words.

7. Lemmatization/Stemming:

Lemmatization and stemming are techniques used to reduce words to their base or root form. For example, "running" and "ran" would both be reduced to the base form "run." This process helps in standardizing the text and reducing the vocabulary size.

4.2. Feature Extraction

Feature extraction involves transforming the preprocessed textual data into numerical representations that can be fed into machine learning models.

1. Word Embeddings:

Word embeddings are dense vector representations of words in a continuous vector space. They capture semantic relationships between words, allowing the model to understand the contextual meaning of words based on their distribution in the text corpus. Techniques like Word2Vec, GloVe, and FastText are commonly used for generating word embeddings.

2. Bag-of-Words (BoW):

BoW representation represents text as a vector of word frequency counts. It disregards word order and context but is effective in capturing the presence of words in the text.

3. TF-IDF (Term Frequency-Inverse Document Frequency):

TF-IDF measures the importance of a word in a document relative to a corpus. It assigns higher weights to words that are frequent in a document but rare in the corpus, thus emphasizing words that are discriminative for sentiment analysis.

4.3. Model Architecture

The model architecture defines the structure of the neural network used for sentiment analysis. Here's a detailed explanation:

1. Embedding Layer:

The embedding layer learns word embeddings from the input text data. It maps each word to a dense vector representation in a continuous vector space, capturing semantic relationships between words.

2. Convolutional Neural Network (CNN):

CNNs are effective in capturing local patterns and spatial relationships within text data. They consist of convolutional layers followed by pooling layers, enabling the model to extract hierarchical features from the text.

3. Long Short-Term Memory (LSTM) Network:

LSTMs are a type of recurrent neural network (RNN) capable of capturing long-range dependencies and sequential information in text data. They are well-suited for modeling sequential data and are effective in tasks involving time-series prediction and natural language processing.

4.4. Model Training and Evaluation

Model training involves feeding the preprocessed and feature-extracted data into the neural network model to learn patterns and relationships. Here's a detailed explanation:

1. Training Data:

The dataset is split into training and validation sets, with the training set used for model training and the validation set used for model evaluation.

2. Loss Function:

The loss function measures the difference between the predicted and actual sentiment labels. Common loss functions for binary classification tasks include binary cross-entropy and hinge loss.

3. Optimizer:

The optimizer updates the model parameters during training to minimize the loss function. Popular optimizers include Adam, RMSprop, and Stochastic Gradient Descent (SGD).

4. Metrics:

Metrics such as accuracy, precision, recall, and F1-score are used to evaluate the performance of the model on the validation set. These metrics provide insights into the model's ability to correctly classify sentiment labels.

4.5. Model Optimization and Callbacks

Model optimization techniques are applied to improve model performance and prevent overfitting. Here's a breakdown:

1. Regularization:

Techniques like dropout regularization and batch normalization are applied to prevent overfitting by introducing noise or normalizing the activations of the network layers.

2. Learning Rate Schedule:

The learning rate schedule adjusts the learning rate during training to prevent the model from getting stuck in local minima and to facilitate faster convergence.

3. Callbacks:

Callbacks like ModelCheckpoint and EarlyStopping are used to save the best-performing model weights during training and stop training when the validation loss stops improving, respectively.

4.6. Model Architecture for CNN Model:

The sentiment analysis model is built using Keras, a high-level neural networks API running on top of TensorFlow. The model architecture consists of several layers designed to process and extract features from the input data for sentiment classification.

1. Embedding Layer:

Configuration: An embedding layer is added to the model to learn word embeddings from the input sequences.

Input Dimension: The input dimension of the embedding layer matches the vocabulary size plus one to accommodate out-of-vocabulary words, with dimensions set to match the Word2Vec vector size (W2V_SIZE).

Pretrained Weights: Pretrained word embeddings are initialized using the embedding matrix derived from the Word2Vec model, ensuring compatibility and leveraging semantic information captured during feature extraction.

Trainable: The embedding layer weights are set to non-trainable to prevent further training and preserve the learned word embeddings.

2. Dropout Layer:

Regularization: A dropout layer with a dropout rate of 0.3 is added to mitigate overfitting by randomly dropping 30% of the input units during training.

3. Convolutional Layer (Conv1D):

Configuration: A one-dimensional convolutional layer is introduced to capture local patterns and spatial relationships within the text data.

Filter Size and Activation: The convolutional layer consists of 300 filters with a filter size of 3 and ReLU activation function to introduce non-linearity and extract relevant features.

4. Global Max Pooling Layer:

Pooling Operation: Global max pooling is applied to the output of the convolutional layer to extract the most salient features across the entire sequence, aggregating information for subsequent processing.

5. Dense Layers:

Configuration: Two fully connected dense layers are added to the model to further process the extracted features and perform sentiment classification.

Activation Functions: The first dense layer consists of 200 neurons with a sigmoid activation function, followed by a dropout layer with a dropout rate of 0.3 to prevent overfitting.

Output Layer: The final dense layer with a single neuron and sigmoid activation function produces the binary sentiment classification output (positive or negative).

6. Model Compilation:

Loss Function: The binary cross-entropy loss function is selected for binary classification tasks, measuring the difference between predicted and actual sentiment labels.

Optimizer: The Adam optimizer is chosen for model optimization, efficiently updating model weights during training.

Metrics: Model performance is evaluated based on accuracy, measuring the proportion of correctly classified samples.

7. Model Summary:

The model summary provides an overview of the model architecture, including the number of parameters in each layer and the output shape at each stage, facilitating model inspection and debugging.

4.7 Model Architecture for LSTM:

The LSTM model is another powerful architecture for sequential data processing, particularly effective for tasks like sentiment analysis. Here's a breakdown of its components:

1. Embedding Layer:

Configuration: Similar to the CNN model, an embedding layer is added to the LSTM model to learn word embeddings from the input sequences.

Input Dimension: The input dimension of the embedding layer matches the vocabulary size plus one to accommodate out-of-vocabulary words. It also aligns with the Word2Vec vector size (W2V_SIZE) for compatibility.

Pretrained Weights: The embedding layer initializes pretrained word embeddings using the embedding matrix derived from the Word2Vec model, preserving semantic information captured during feature extraction.

Trainable: In this model, the embedding layer weights are set to trainable, allowing them to be updated during model training to improve performance.

2. Dropout Layer:

Regularization: A dropout layer with a dropout rate of 0.5 is added to mitigate overfitting by randomly dropping 50% of the input units during training.

3. LSTM Layer:

Configuration: The LSTM layer is the core component of the model, responsible for processing sequential data and capturing long-range dependencies.

Number of Units: The LSTM layer consists of 128 memory units, controlling the number of memory cells or blocks in the network. Higher values may capture more complex patterns but may also increase computational complexity.

Dropout Rate: Dropout regularization is applied within the LSTM layer itself, with a dropout rate of 0.5 for input units and recurrent connections, enhancing model robustness.

4. Dense Layer:

Configuration: A dense layer is added to the model to perform sentiment classification based on the features extracted by the LSTM layer.

Activation Function: The dense layer utilizes a sigmoid activation function, producing a binary sentiment classification output (positive or negative).

5. Model Compilation:

Loss Function: Similar to the CNN model, the binary cross-entropy loss function is chosen for binary classification tasks, quantifying the difference between predicted and actual sentiment labels.

Optimizer: The Adam optimizer is selected for model optimization, efficiently updating model weights based on gradient descent during training.

Metrics: Model performance is evaluated using accuracy, measuring the proportion of correctly classified samples, and potentially other metrics like precision, recall, and F1-score for a comprehensive assessment.

Model Summary:

The model summary provides a concise overview of the LSTM model architecture, detailing the number of parameters in each layer, the output shape at each stage, and the total number of trainable parameters. This summary aids in model inspection, debugging, and optimization

Chapter 5

Case Study on Sentiment Analysis Systems

This chapter represents various case studies in which sentiment analysis is currently being used..

5.1 Introduction

With the use of sentiment analysis, also known as opinion mining, organizations, scholars, and decision-makers can gain valuable insights from textual data by examining the sentiment, emotions, and attitudes that are expressed in the text. With an emphasis on three different types of sentiment analysis systems—Twitter Sentiment Analysis, Product Review Sentiment Analysis, and Social Media Sentiment Analysis—this case study investigates the use of sentiment analysis across many platforms.

5.2 Amazon Product Review Sentiment Analysis:

Platform Overview:

One of the biggest online retailers in the world, Amazon offers a wide selection of goods in a number of categories, such as electronics, books, clothes, and home goods. An abundant supply of data for sentiment analysis is offered by Amazon, which has millions of product listings and customer reviews.

Use Case:

Businesses that sell goods on Amazon can use sentiment analysis to learn more about the preferences, views, and levels of satisfaction of their customers. Sellers can improve sales and customer retention by addressing negative feedback or complaints, identifying characteristics that appeal to buyers, and optimizing product listings by examining Amazon product reviews.

Methodology:

Sentiment analysis of Amazon product reviews entails using web scraping methods or the Product Advertising API to gather review data from Amazon product pages. Preprocessing operations are performed on the gathered text data, including the removal of HTML tags, punctuation, and stopwords. Word embeddings, sentiment lexicons, and review ratings are a few examples of feature extraction techniques. To categorize reviews into positive, negative, or neutral sentiment categories, machine learning models—such as Naive Bayes, Support Vector Machines (SVM), or deep learning architectures like Recurrent Neural Networks (RNNs) or Transformer-based models—are trained on labeled review datasets.

Challenges:

Numerous obstacles stand in the way of sentiment analysis of Amazon product reviews: the volume of reviews, the existence of phony or paid reviews, and the requirement to manage vocabulary unique to the product and domain. Furthermore, the availability and caliber of review data for research may be impacted by Amazon's continuously changing review regulations and algorithms.

Conclusions:

Sentiment analysis of Amazon product reviews provides sellers with useful information about customer sentiment, how to enhance product quality, and how to best use marketing tactics. Sellers can use client feedback to make data-driven decisions and maintain their competitiveness in the e-commerce market by utilizing sentiment analysis tools.

5.3 Instagram Sentiment Analysis:

Overview:

Instagram is a well-known social networking site where users can share photos and videos. Instagram offers a visually stimulating environment for users to express themselves through photos, videos, stories, and captions. There are currently over 1 billion monthly active users on the platform.

Use Case:

Instagram sentiment analysis is a useful tool for businesses and marketers to gauge consumer opinion on their brands, goods, and advertising efforts. Businesses may learn about consumer preferences, spot brand evangelists and critics, and assess the success of their Instagram marketing campaigns by examining user-generated content, such as posts, comments, and stories.

Methodology:

Using the Instagram Graph API or web scraping methods, data from public Instagram accounts, hashtags, or location tags are gathered for Instagram sentiment analysis. Preprocessing is done on the gathered data, which includes image analysis with computer vision algorithms to extract features from photos and videos, such as object recognition, scene detection, and emotion detection. Natural language processing (NLP) approaches are used to tokenize, sanitize, and analyze textual data, such as comments and captions, in order to extract sentiment-related information. Sentiment classifications, such positive, negative, or neutral, are assigned to posts or comments by machine learning models or deep learning architectures that have been trained on labeled Instagram data.

Challenges:

The majority of Instagram material is visual, which makes sentiment analysis on the platform difficult and necessitates the use of advanced image processing algorithms. Additionally, text preprocessing and sentiment categorization are complicated by the variety of languages, emojis, and slang used on Instagram. When gathering and evaluating Instagram data for sentiment analysis, it's crucial to take user privacy and Instagram's data usage guidelines into account.

Conclusion:

Instagram sentiment analysis offers valuable insights into user sentiment, preferences, and behaviors on the platform, enabling businesses to enhance their marketing strategies, engage with their audience effectively, and build stronger brand relationships. By leveraging image analysis and NLP techniques, organizations can unlock the potential of Instagram data to drive actionable insights and make informed decisions in a visually driven social media landscape.

5.4 Healthcare Sentiment Analysis:

Overview:

A plethora of user-generated content about healthcare experiences, treatment outcomes, and medical advice may be found on healthcare platforms, such as social media groups, patient forums, and websites that offer medical reviews.

Use Case:

Sentiment analysis is a useful tool that healthcare organizations, pharmaceutical businesses, and medical experts can use to assess patient happiness, sentiment toward treatments, and input from patients. Healthcare professionals can better care for patients, customize treatment programs, and effectively handle problems by knowing how they feel.

Methodology:

In order to classify patient feedback into positive, negative, or neutral sentiment categories, healthcare sentiment analysis involves gathering data from healthcare platforms, preprocessing the text by anonymizing patient information and removing personal identifiers, extracting features like medical terms, treatment names, and sentiment indicators, and using machine learning or deep learning models.

Challenges:

Ensuring patient privacy and confidentiality, managing medical jargon and terminology, and resolving biases in patient feedback are challenges faced by sentiment analysis in the healthcare industry. Additionally, data security and regulatory compliance must be carefully considered when combining sentiment analysis with electronic health records (EHRs) and healthcare analytics platforms.

Conclusion:

Sentiment analysis in healthcare has the potential to significantly improve patient care, raise the caliber of healthcare services, and increase patient participation. Healthcare firms may measure patient happiness, pinpoint areas for development, and streamline their delivery procedures by monitoring patient feedback and sentiment. The use of sentiment analysis in healthcare is developing despite obstacles pertaining to data privacy, medical language, and regulatory compliance. This is providing insightful information and chances for innovation in the healthcare sector.

5.5 Best Practices:

Opinion mining, another name for sentiment analysis, is a potent method for drawing conclusions from textual data. However, adhering to sentiment analysis standard practices is crucial to guaranteeing accurate data and insightful interpretation. The following are some essential rules:

1. Establish Clear Objectives:

Establish the analysis's objectives in detail before beginning any sentiment analysis. Ascertain the precise queries you wish to address or the knowledge you hope to get from the sentiment analysis procedure.

2. Choose Relevant Data Sources:

Select reliable sources of information that support your goals. Depending on the context of your investigation, take into account sources including news stories, customer reviews, surveys, and social media platforms.

3. Effective Preprocessing of Data:

Prior to sentiment analysis, the textual data should be cleaned and standardized by preprocessing. Eliminating punctuation, stopwords, special characters, and superfluous metadata are some possible steps. To further standardize the text, take into account lemmatization, tokenization, and stemming procedures.

4. Handle Negation and Context:

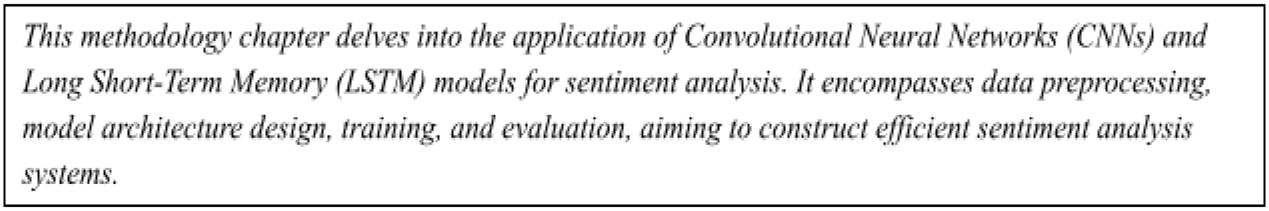
When assessing sentiment, consider negations and context. Changes in sentiment polarity can be captured with the aid of negative handling strategies, such as appending "not" to words that convey sentiment. Accurately assessing sentiment requires contextual information, particularly when dealing with sarcasm, irony, or cultural nuances.

5. Select the Appropriate Model:

Depending on the features of your data and the intricacy of the sentiment analysis task, choose the right sentiment analysis models. Deep learning architectures (e.g., CNNs, RNNs), machine learning algorithms (e.g., SVM, Naive Bayes), and lexicon-based techniques are among the options.

CHAPTER 6

Methodology



This methodology chapter delves into the application of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models for sentiment analysis. It encompasses data preprocessing, model architecture design, training, and evaluation, aiming to construct efficient sentiment analysis systems.

6.1 Dataset Overview

The foundation of our Twitter Sentiment Analysis project is a carefully selected dataset that we obtained from Sentiment140, a repository that is well-known for its extensive and varied set of tweets that have been labeled with sentiments. Including an astounding 1,600,000 tweets, this large dataset is carefully acquired through the powerful capabilities of the Twitter API, guaranteeing a thorough portrayal of opinions shared on the Twitter network.

Sentiment labels are carefully applied to every tweet in the dataset, identifying positive feelings (labeled as 4) as opposed to neutral sentiments (labeled as 2) and negative sentiments (labeled as 0). This careful annotation makes it possible to perform sophisticated sentiment analysis and identify minute differences in the emotional expressions that people on Twitter display.

In addition to emotion labels, the dataset contains a wealth of critical metadata, such as timestamps, tweet IDs, and user details, which offer crucial context for the temporal dynamics and place of origin of each individual tweet. Additionally, the dataset captures the unadulterated spirit of Twitter conversation by maintaining the original wording of every tweet, acting as a rich tapestry of feelings, viewpoints, and ideas that are shared throughout the Twitter network.

This carefully selected dataset is the lifeblood of our sentiment analysis project; it provides us with a solid base from which to start our investigation. This dataset, with its wide range of sentiments and detailed information, enables us to get deep insights into the complexities of human emotions as they appear on Twitter.

6.2 About CNN Model

In this section, we explain the detailed approach used to build and train the sentiment analysis Convolutional Neural Network (CNN) model. The approach includes preparing the data, designing the model architecture, fine-tuning the hyperparameters, training the system, and calculating evaluation metrics in order to create a sentiment analysis tool that is reliable and accurate and meets our goals.

It is a type of deep learning architecture that is largely used for processing and evaluating visual data. They are particularly good at identifying patterns and relationships in sequential data, such as text, however their adaptability also extends to this type of data. Convolutional operations, which enable the network to automatically learn hierarchical feature representations, are at the core of CNNs.

CNNs use activation functions like the Rectified Linear Unit (ReLU) in conjunction with the convolutional operation to add non-linearity to the network. By ensuring that the model can capture complicated interactions between features, ReLU activation improves the model's capacity to learn from complex input. Because ReLU is non-linear, CNNs can represent a wide range of patterns and subtleties seen in text data, which enhances their performance in sentiment analysis applications.

Another crucial part of CNNs is the pooling layer, which is usually positioned after the convolutional layers. By downsampling the feature maps through pooling layers, translation invariance is improved and computational complexity is decreased. A popular pooling method called max pooling keeps the maximum value in each local area of the feature map, protecting significant features and eliminating unimportant data.

CNNs use fully connected layers for classification, which are similar to standard neural networks, after convolutional and pooling layers. These fully linked layers get the flattened feature maps and learn how to mix features from various spatial locations in order to generate predictions. Sentiment categorization is made easier by applying activation functions like sigmoid or softmax to the output layer, which result in probability distributions over many classes.

CNNs use gradient descent and backpropagation to iteratively modify the weights of filters and fully connected layers during training. The model gains the ability to identify pertinent features and patterns in the input data by decreasing the discrepancy between the anticipated and real labels. CNNs employ learnt representations to iteratively improve their capacity to categorize text data into sentiment categories.

CNNs' ability to automatically learn hierarchical data representations is one of their unique advantages. Higher layers of the network often learn more abstract and complicated properties, while lower levels typically capture low-level data like edges or simple patterns. CNNs are capable of accurately classifying text data in sentiment analysis by identifying relevant linguistic patterns, sentiment indicators, and contextual signals at different levels of abstraction.

Convolutional operations, non-linear activation functions, and hierarchical feature learning are three powerful tools that CNNs use to effectively capture complex patterns and structures in text data. This makes them ideal for jobs involving sentiment analysis, because precise categorization depends on the subtle interpretation of language.

In the convolutional procedure, the input data is passed through tiny filters, sometimes referred to as kernels. Every filter generates feature maps that highlight pertinent patterns by computing a dot product with local regions of the input. In text analysis, for example, filters might identify particular word sequences or linguistic structures that convey sentiment. Convolutional operations are applied repeatedly by CNNs, teaching them to extract ever complicated and abstract characteristics from

i. Data Preprocessing:

A carefully selected dataset from Sentiment140 forms the basis of our CNN-based sentiment analysis algorithm. This dataset is made up of 1.6 million tweets that have been annotated with labels indicating positive, neutral, and negative sentiments. The collection also contains crucial metadata, which gives useful context for the temporal dynamics and origin of each tweet, such as timestamps, user information, and tweet IDs.

ii. Data cleaning:

Special characters, punctuation, URLs, and other noise are eliminated from the tweet content as part of the preprocessing stage. Additionally, the text data is standardized through the use of tokenization and lemmatization procedures, which guarantee consistency and make feature extraction easier later on.

iii. Text Vectorization:

Using the Tokenizer class from the Keras library, the preprocessed text data is tokenized and turned into sequences of integers. Then, in order to guarantee uniform length for the input to the CNN model, these sequences are padded.

The CNN model architecture has been carefully crafted to fully utilize convolutional layers for the purpose of extracting features from textual data. Convolutional layers, pooling layers, fully connected layers, and activation functions are some of the main parts of the design.

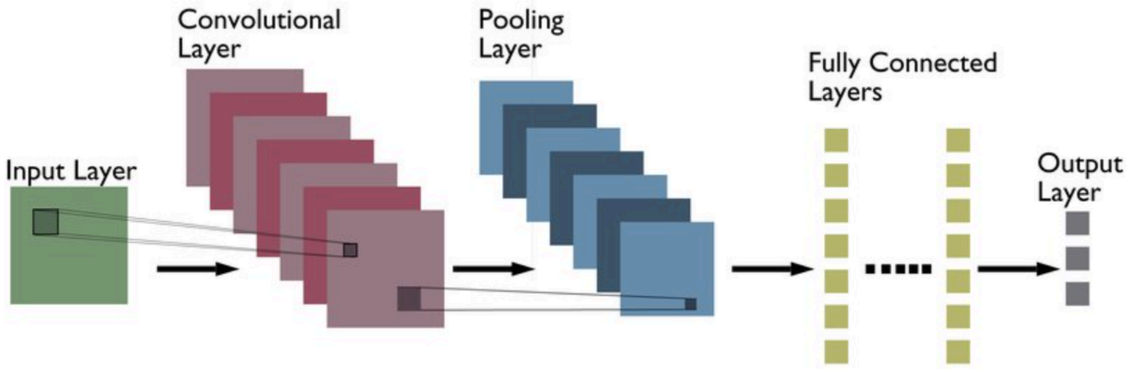


Fig 6.1 Block diagram of CNN Model

iv. Convolutional Layers:

To extract local features and patterns from the text, the input data is subjected to convolution operations using learnable filters. ReLU activation functions are used by these convolutional layers to add non-linearity and aid in feature learning.

v. Pooling Layers:

After the convolutional layers, maximum pooling layers are added to the feature maps in order to downsample them and save computational complexity while keeping significant features. By calculating the greatest value inside each local region of the feature map, max pooling improves the feature map's resilience to spatial changes.

vi. Fully Connected Layers:

To integrate extracted features for sentiment classification, the pooling layers' output is flattened and routed via fully connected layers. To predict sentiment, these fully linked layers use sigmoid activation algorithms.

vii. Model Compilation:

The binary cross-entropy loss function, which calculates the discrepancy between the sentiment labels that are predicted and those that are actual, is used to compile the CNN model. Model optimization is done with the help of the Adam optimizer, which effectively updates model weights during training. Accuracy is the evaluation metric that counts the percentage of instances that are accurately classified.

viii. Training Process:

To ensure a balanced distribution of sentiment labels in both sets, the dataset is divided into training and testing sets using a standard ratio. With the training data, mini-batch stochastic gradient descent is used to train the CNN model. To avoid overfitting and maximize model performance, early halting and learning rate reduction callbacks are incorporated into the training process.

ix. Hyperparameter tuning:

To attain optimal performance, hyperparameters including batch size, number of epochs, filter sizes, and dropout rates are empirically adjusted. Effective exploration of the hyperparameter space can be achieved by using grid search or random search approaches.

x. Evaluation measures:

A number of measures, such as accuracy, precision, recall, and F1-score, are used to assess the CNN model's performance. The distribution of true positive, true negative, false positive, and false negative predictions is also visualized using a confusion matrix, which sheds light on the model's advantages and disadvantages.

xi. Mathematical Formulas used:

Convolutional Neural Networks (CNNs) utilize several mathematical formulas to perform operations such as convolution, pooling, and activation functions. Here are some of the key mathematical formulas used in CNNs:

1. Convolution Operation:

The convolution operation between an input image X and a filter W at position (i, j) is calculated as:

$$(X * W)(i, j) = \sum_{m} \sum_{n} X(m, n) \cdot W(i - m, j - n)$$

where (X(m, n)) represents the pixel intensity of the input image at position (m, n) and (W(i - m, j - n)) represents the weight of the filter at position (i - m, j - n).

2. Pooling Operation:

Max Pooling: The max pooling operation takes the maximum value within a local region of the input feature map. For a \(2 \times 2 \) pooling window, the operation is defined as:

$$\text{MaxPooling}(X)_{i,j} = \max(X(2i, 2j), X(2i, 2j + 1), X(2i + 1, 2j), X(2i + 1, 2j + 1))$$

3. Activation Functions:

ReLU (Rectified Linear Unit): The ReLU activation function is commonly used in CNNs and is defined as:

$$\mathrm{ReLU}(x) = \max(0, x)$$

4. Fully Connected Layer:

The output of the convolutional and pooling layers is flattened before being passed through a fully connected layer. If X is the flattened input vector and W is the weight matrix of the fully connected layer, the output Y is calculated as:

$$Y = \mathrm{ReLU}(X \cdot W + b)$$

where b is the bias vector.

By following this meticulously crafted methodology, we aim to develop a highly accurate and robust CNN-based sentiment analysis model capable of effectively classifying text data into positive and negative sentiment categories, thereby facilitating deeper insights into the sentiments expressed on social media platforms like Twitter.

6.3 About LSTM Model

In this section, we explain the detailed approach used to build and train the sentiment analysis Long Short Term Memory (LSTM) model. The approach includes preparing the data, designing the model architecture, fine-tuning the hyperparameters, training the system, and calculating evaluation metrics in order to create a sentiment analysis tool that is reliable and accurate and meets our goals.

i. Working of LSTM:

An input gate, an output gate, a forget gate, and a cell state make up the fundamental parts of an LSTM unit. Together, these parts enable the LSTM to maintain pertinent information over extended periods by selectively updating and propagating information over time steps.

ii. Cell State:

The cell state is the LSTM's "memory" and it spans the whole sequence. It can be compared to a conveyor belt that moves data between several time increments. The LSTM is well-suited for applications that require modeling long-range dependencies because of its ability to retain information over lengthy sequences because of its cell state.

iii. Input Gate:

Information entering the cell state is managed by the input gate. It is composed of a pointwise multiplication operation after a sigmoid activation function. Values between 0 and 1, which indicate how much of each component should be updated, are output by the sigmoid function. After that, the sigmoid gate's input and output are combined using a pointwise multiplication operation to determine which data is necessary to update the cell state.

iv. Forget Gate:

The forget gate selects which data from the cell state to remove. It has a sigmoid activation function and a pointwise multiplication operation, just like an input gate. The forget gate determines which data is no longer relevant and ought to be forgotten by taking as inputs the current input and the prior cell state.

v. Output Gate:

This device regulates the information transfer from the cell state to the output. It is composed of a pointwise multiplication operation after a hyperbolic tangent (tanh) activation function and a sigmoid activation function. The tanh function squashes the values to the range [-1, 1], making them appropriate for the following layer in the network, while the sigmoid function controls the amount of information to output.

vi. LSTM architecture:

An LSTM is usually made up of several LSTM units layered one after the other to create different network levels. After processing input sequences over time steps, each LSTM unit updates its internal state and outputs pertinent data to layers or time steps below it.

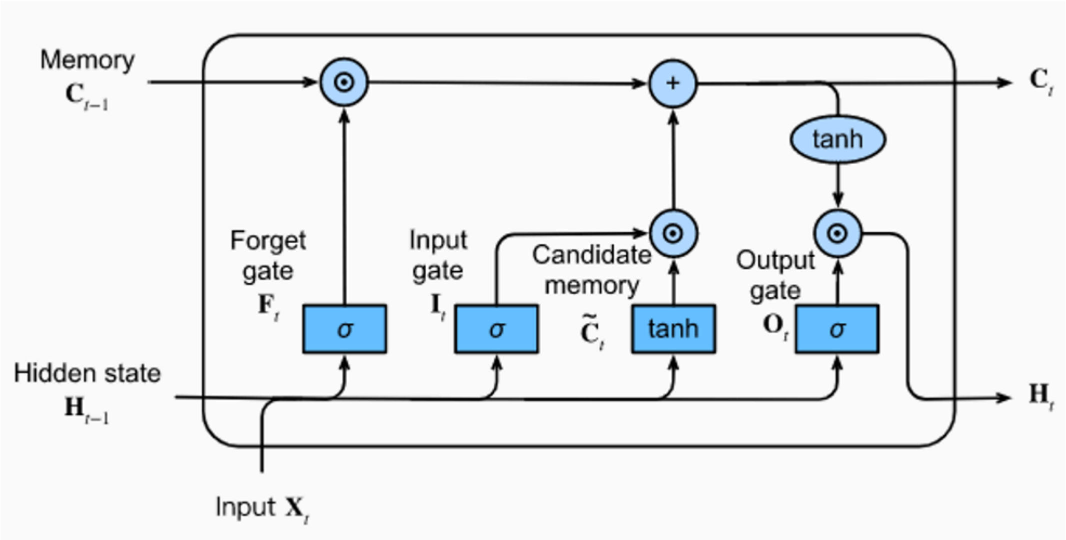


Fig 6.2 Block Diagram of LSTM Model

vii. Data Preprocessing:

Sentiment polarity-labeled 1.6 million tweets make up the Sentiment140 dataset, which was selected for sentiment analysis. This dataset is ideal for supervised sentiment analysis applications since it offers a wide range of tweets on different subjects.

viii. Data Cleaning:

The text data underwent a number of preprocessing processes in order to enable analysis. This involved eliminating punctuation, URLs, special characters, and other non-alphanumeric characters. Tokenization was also used to divide the text into discrete words or tokens, and lemmatization was used to return the words to their most basic form.

ix. Text Vectorization:

The Tokenizer class from the Keras library was used to tokenize the preprocessed text. This required turning the text into sequences of integers, with an integer index assigned to each distinct word. After that, these sequences were padded to guarantee that the length of each input to the LSTM model was the same. In order to match the length of the longest sequence in the dataset, padding involved appending zeros to the start or finish of sequences.

x. Model Architecture Design:

To understand long-range dependencies within the input sequences and efficiently capture the sequential nature of text data, the LSTM model architecture was created.

xi. Embedding Layer:

The input sequences that have been integer-encoded are mapped to dense vectors of a fixed size by the model's embedding layer. Based on the contextual similarity of the input words, this layer develops a dense representation of the words.

xii. Layers of LSTM:

For the purpose of processing the sequential input data, stacked LSTM layers were used. Multiple memory cells in each LSTM layer store information throughout time and transfer it to the subsequent time step. The model may learn hierarchical representations of the input data thanks to the employment of numerous LSTM layers.

xiii.Dense Layers:

To do classification based on the learnt representations, dense fully connected layers were added to the model after the LSTM layers. One neuron with a sigmoid activation function makes up the output layer. It generates a probability score that shows how likely it is that each input will fall into a specific sentiment class.

xiv. Training Process:

The Adam optimizer, which works well for binary classification problems like sentiment analysis, was used to train the LSTM model using a binary cross-entropy loss function.

xv. Batch Size and Epochs:

A predetermined number of epochs and a batch size of 512 were used to train the model. In order to minimize the loss function, the model iteratively changed its weights during training based on the training data.

xvi. Callbacks:

To keep an eye on the training process and modify the learning rate or end training early if performance metrics on the validation set did not improve, callbacks like ReduceLROnPlateau and EarlyStopping were employed.

xvii. Evaluation Metrics:

Accuracy, precision, recall, and F1-score were among the metrics used to assess the LSTM model's performance.While precision quantifies the percentage of successfully predicted positive cases among all positively predicted instances, accuracy assesses the overall soundness of the model's predictions. The percentage of accurately anticipated positive instances among all actual positive instances is determined by recall, which is sometimes referred to as sensitivity. The model's performance can be fairly assessed using the F1-score, which is the harmonic mean of precision and recall. This is especially useful when there is an imbalance in the classes. In order to provide insight into the model's performance across several sentiment classes, confusion matrices were also created to display the distribution of true positive, true negative, false positive, and false negative predictions.

xviii. Mathematical Formulae used:

a. Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

b. Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

c. Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

d. Candidate Cell State

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

e. Hidden State

$$h_t = o_t \odot \tanh(C_t)$$

f. Sigmoid Activation function: The sigmoid activation function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

g. Hyperbolic Tangent Activation Function: The hyperbolic tangent activation function is defined as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

To summarize, long-term dependencies in sequential data are well captured by LSTM networks, which propagate and update information via time steps in a selective manner. They are a valuable tool for a variety of NLP applications due to their unique architecture, which combines gated units and a memory cell to successfully model complex sequential patterns.

Chapter 7

Design Considerations

This chapter represent the design consideration for implementing twitter sentiment analysis using different models.

7.1 Introduction

The design factors that go into choosing the right model architecture in the fields of machine learning and deep learning are essential to attaining peak performance and efficiency. Design considerations cover a wide range of topics, such as the type of problem, dataset properties, computing capabilities, and performance indicators. We discuss the reasoning behind the selection of model architectures in this section for sentiment analysis, with particular attention to the Long Short-Term Memory (LSTM) model created for this test and a Convolutional Neural Network (CNN) model created by another person.

Design considerations play a pivotal role in guiding the selection process between different model architectures, as they influence the model's ability to effectively capture and represent complex relationships within the data. By carefully evaluating the strengths and weaknesses of each architecture and aligning them with the requirements of the sentiment analysis task, researchers and practitioners can make informed decisions to optimize model performance and achieve desired outcomes.

7.2 Model Selection

Sentiment analysis tasks involving text data with different sequence lengths are a good fit for the LSTM model because of its capacity to capture long-term dependencies in sequential data. Because LSTMs are so good at modeling intricate relationships and phrase context, they can accurately capture sentiment nuances and semantic meaning.

But in the CNN model created by someone else, local characteristics are extracted from text data using convolutional layers, and then the most pertinent features are captured using max-pooling layers. Due to its propensity for seeing spatial patterns, CNNs have been utilized extensively in picture recognition applications; however, they can also be customized for text classification tasks such as sentiment analysis.

It's critical to weigh the advantages and disadvantages of the two architectures in relation to the sentiment analysis task. Because LSTMs can effectively model sequential data with long-range dependencies, they are a good choice for capturing the finer points of sentiment that are conveyed over longer texts. They are less sensitive to the word order in a sentence and especially useful for handling sequences of varying length.

CNNs, on the other hand, are skilled in identifying spatial correlations and local patterns in the input data. Even while they might not be as good at capturing long-range relationships as LSTMs are, they can still be quite useful in sentiment analysis tasks, particularly when the main goal is to pinpoint specific traits or phrases that convey a particular mood. Large-scale text classification problems are suited for CNNs because they can be trained more quickly and more computationally efficiently than LSTMs.

In general, the particular requirements of the sentiment analysis task—such as the type of text input, the required interpretability level, computational resources, and performance metrics—determine whether to use CNN or LSTM architectures. Researchers and practitioners can choose the best model architecture to get the best results by carefully weighing these variables.

7.3 Strengths and Weakness

The LSTM model has various advantages, such as:

i. Ability to model long-range dependencies:

Long Short-Term Memory Banks (LSTMs) are well-suited for jobs that necessitate the understanding of sentiment nuances and phrase meanings because they can capture contextual information and dependencies across numerous time steps.

ii. Efficient management of sequential data:

Long short-term memory and context preservation are made possible by LSTMs' sequential processing of input sequences, which is crucial for sentiment analysis tasks requiring text material with varying lengths.

But the LSTM model also has several drawbacks, like:

i. Computational complexity:

LSTMs require a lot of computation, particularly when working with deep architectures and big datasets. LSTM model training can be time-consuming and computationally demanding, especially for workloads requiring a lot of hyperparameter tweaking.

ii. Vulnerability to vanishing gradients:

LSTMs are susceptible to vanishing gradient issues, which can impair model performance and training convergence, particularly in deep architectures or lengthy sequences.

The CNN model, however, has the following advantages:

i. Effectiveness in extracting local patterns and features from input data:

CNNs are very good at extracting local patterns and features from text data, which makes them useful for tasks like sentiment analysis that require feature extraction.

ii. Parallel processing:

CNNs can achieve quicker training and inference times than sequential models like LSTMs by utilizing parallel processing across several convolutional filters to provide effective feature extraction and computation.

The CNN model does, however, have certain drawbacks, such as:

i. Limited capacity to catch long-range dependencies:

CNNs' fixed-size windows and local pattern-capturing architecture may make it difficult for them to effectively capture semantic context and long-range relationships in text input, particularly in tasks involving variable-length sequences.

ii. Sensitivity to hyperparameters:

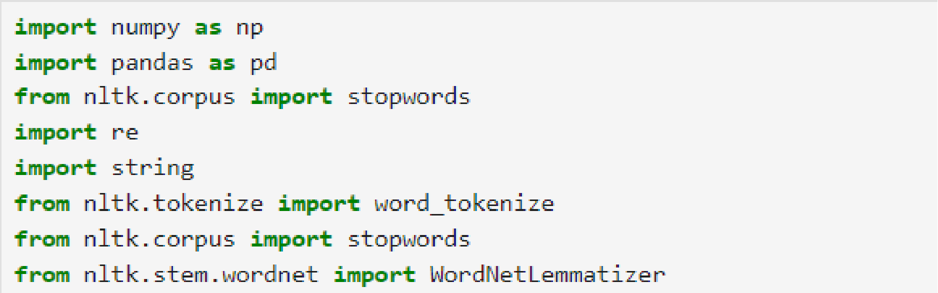
In order to attain the best results, CNN models' performance may be sensitive to hyperparameters such as filter sizes, the number of filters, and pooling algorithms.

Chapter 8

Implementation

This chapter represents the implementation of Twitter Sentiment Analysis using LSTM and CNN models.

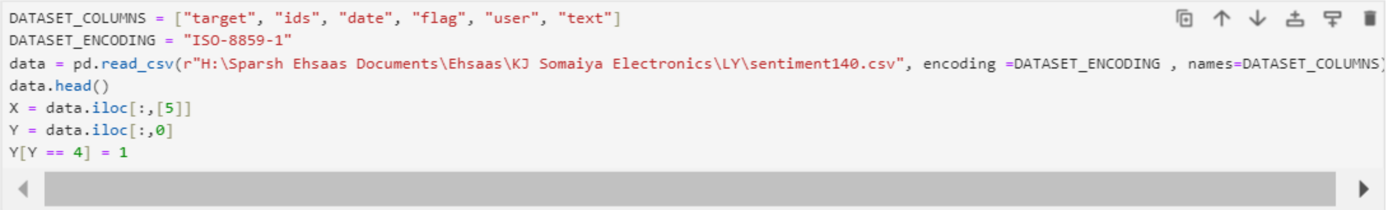
8.1 Importing the necessary modules



```
import numpy as np
import pandas as pd
from nltk.corpus import stopwords
import re
import string
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
```

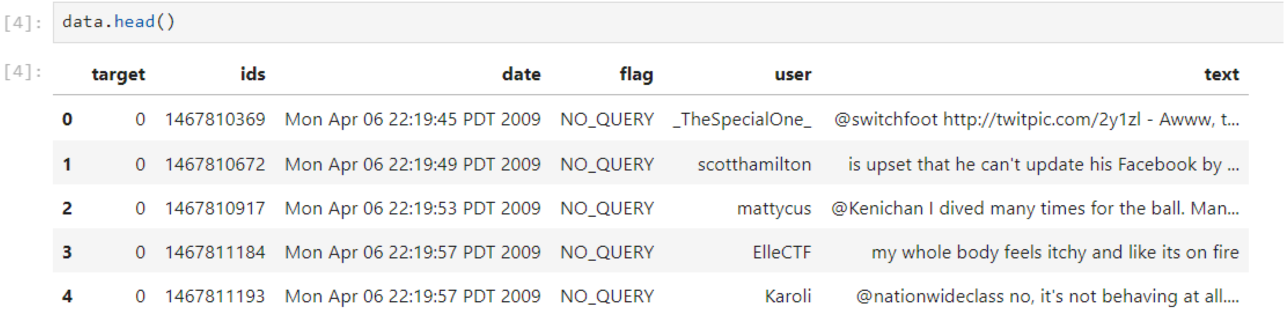
Fig 8.1 Importing modules

8.2 Loading the Dataset



```
DATASET_COLUMNS = ["target", "ids", "date", "flag", "user", "text"]
DATASET_ENCODING = "ISO-8859-1"
data = pd.read_csv(r"H:\Sparsh Ehsaas Documents\Ehsaas\KJ Somaiya Electronics\LY\sentiment140.csv", encoding =DATASET_ENCODING , names=DATASET_COLUMNS)
data.head()
X = data.iloc[:,[5]]
Y = data.iloc[:,0]
Y[Y == 4] = 1
```

Fig 8.2 Loading the dataset



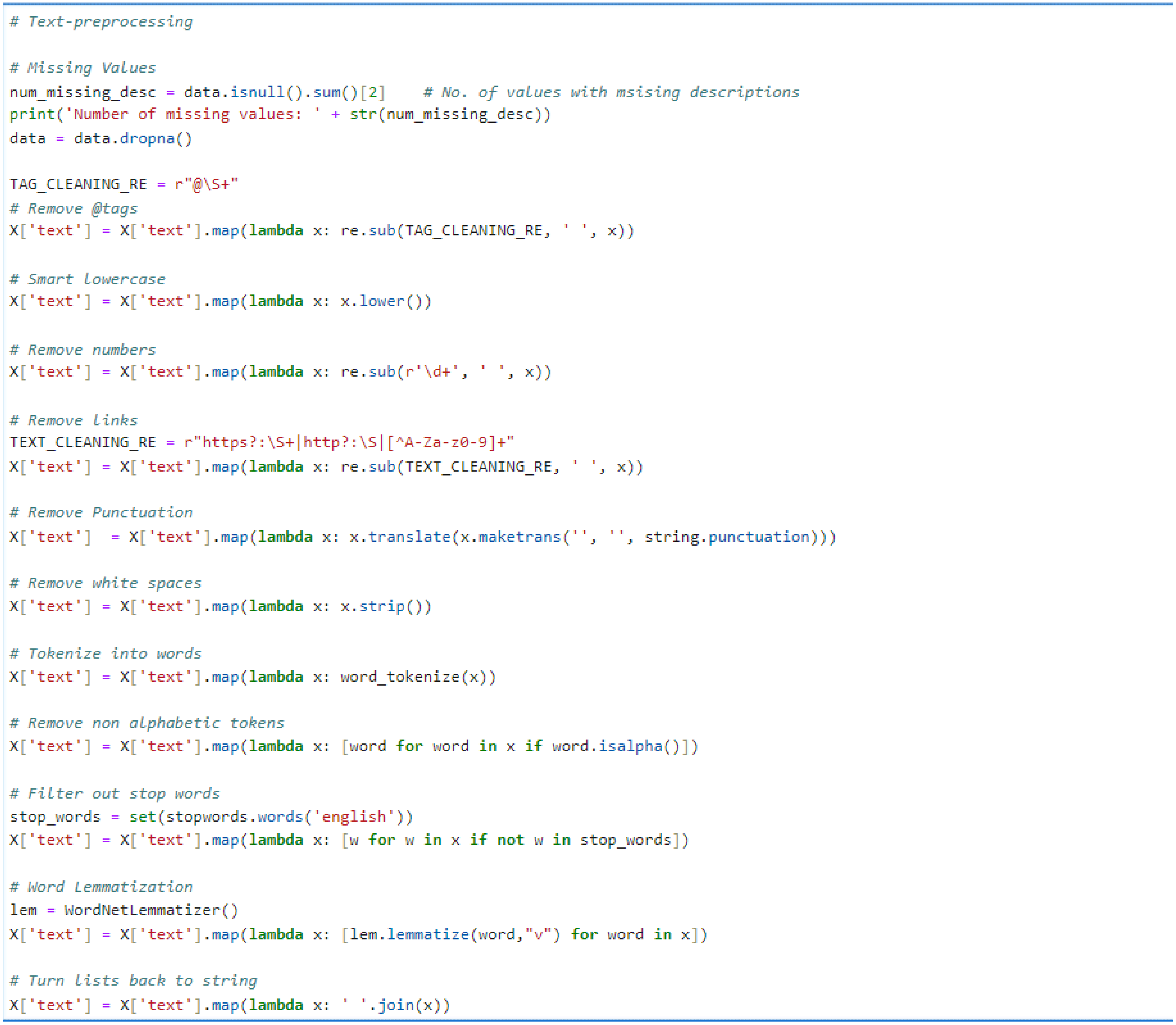
```
[4]: data.head()
```

[4]:

	target	ids	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Fig 8.3 Dataset

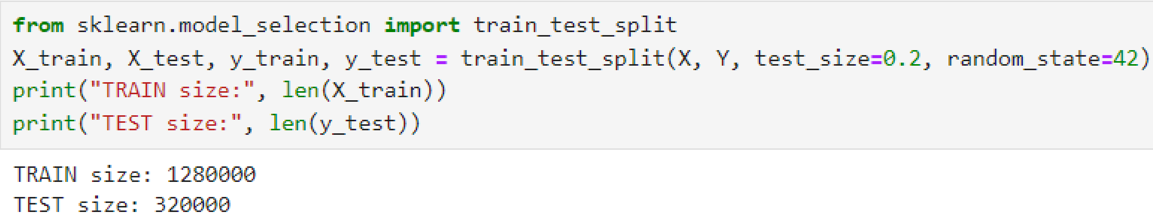
8.3 Text Preprocessing



```python
# Text-preprocessing

# Missing Values
num_missing_desc = data.isnull().sum()[2]    # No. of values with msising descriptions
print('Number of missing values: ' + str(num_missing_desc))
data = data.dropna()

TAG_CLEANING_RE = r"@\S+"
# Remove @tags
X['text'] = X['text'].map(lambda x: re.sub(TAG_CLEANING_RE, ' ', x))

# Smart lowercase
X['text'] = X['text'].map(lambda x: x.lower())

# Remove numbers
X['text'] = X['text'].map(lambda x: re.sub(r'\d+', ' ', x))

# Remove links
TEXT_CLEANING_RE = r"https?:\S+|http?:\S|[^A-Za-z0-9]+"
X['text'] = X['text'].map(lambda x: re.sub(TEXT_CLEANING_RE, ' ', x))

# Remove Punctuation
X['text']  = X['text'].map(lambda x: x.translate(x.maketrans('', '', string.punctuation)))

# Remove white spaces
X['text'] = X['text'].map(lambda x: x.strip())

# Tokenize into words
X['text'] = X['text'].map(lambda x: word_tokenize(x))

# Remove non alphabetic tokens
X['text'] = X['text'].map(lambda x: [word for word in x if word.isalpha()])

# Filter out stop words
stop_words = set(stopwords.words('english'))
X['text'] = X['text'].map(lambda x: [w for w in x if not w in stop_words])

# Word Lemmatization
lem = WordNetLemmatizer()
X['text'] = X['text'].map(lambda x: [lem.lemmatize(word,"v") for word in x])

# Turn lists back to string
X['text'] = X['text'].map(lambda x: ' '.join(x))
```

Fig 8.4 Text Preprocessing

8.4 Splitting the model into training and testing

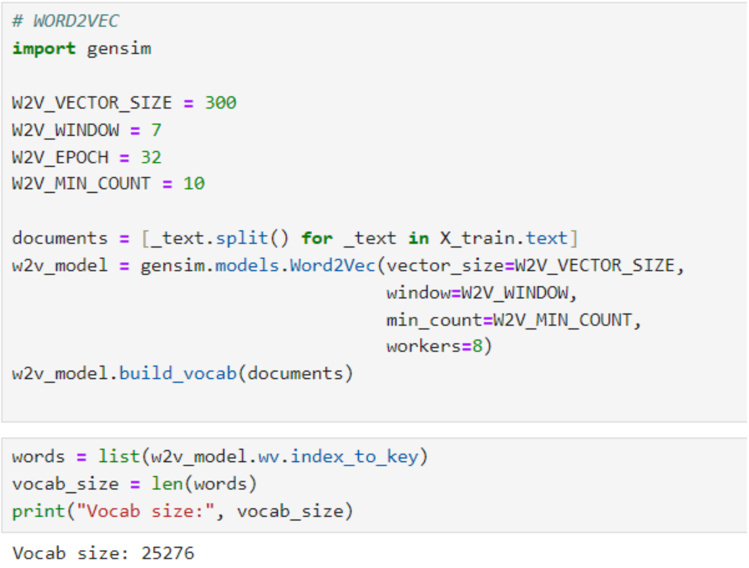


```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
print("TRAIN size:", len(X_train))
print("TEST size:", len(y_test))
```

```
TRAIN size: 1280000
TEST size: 320000
```

Fig 8.5 Train and Test Batch Size

8.5 Building vocab using Word2vec



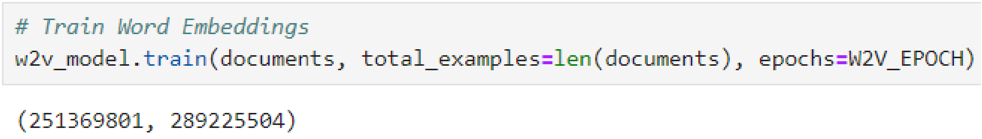
```
# WORD2VEC
import gensim

W2V_VECTOR_SIZE = 300
W2V_WINDOW = 7
W2V_EPOCH = 32
W2V_MIN_COUNT = 10

documents = [_text.split() for _text in X_train.text]
w2v_model = gensim.models.Word2Vec(vector_size=W2V_VECTOR_SIZE,
                                   window=W2V_WINDOW,
                                   min_count=W2V_MIN_COUNT,
                                   workers=8)
w2v_model.build_vocab(documents)
```

```
words = list(w2v_model.wv.index_to_key)
vocab_size = len(words)
print("Vocab size:", vocab_size)
```

```
Vocab size: 25276
```

Fig 8.6 Building Vocab

8.6 Training the word embeddings

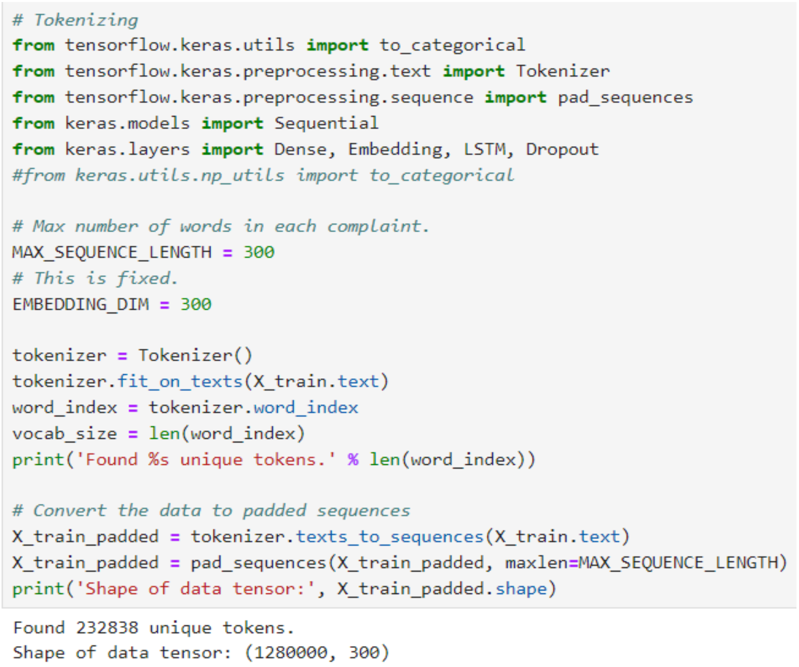


```
# Train Word Embeddings
w2v_model.train(documents, total_examples=len(documents), epochs=W2V_EPOCH)
```

```
(251369801, 289225504)
```

Fig 8.7 Training the word embeddings

8.7 Tokenizing



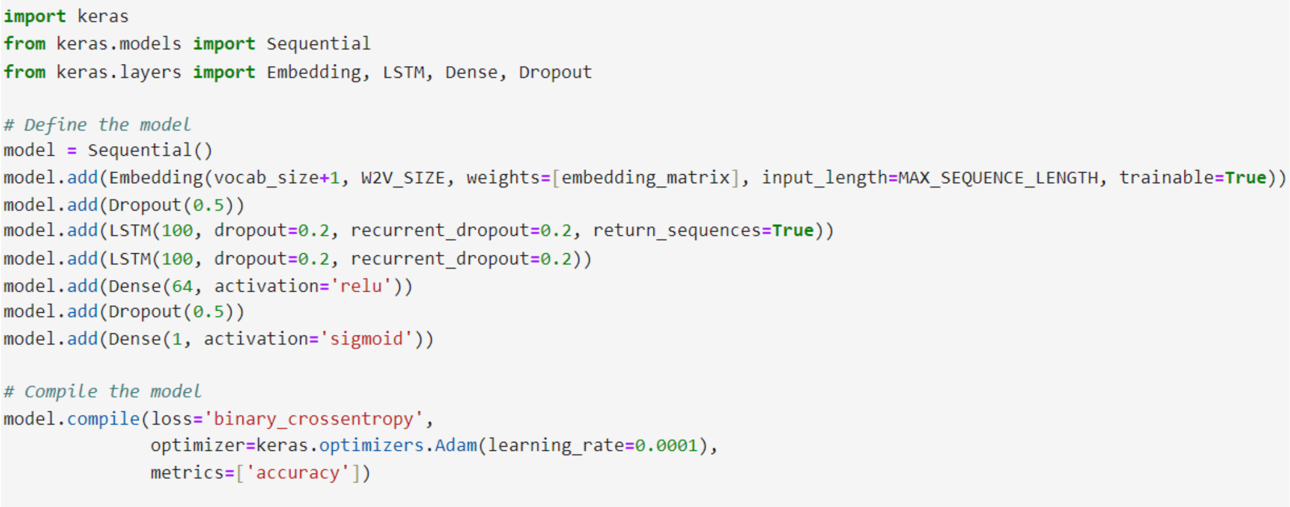
```
# Tokenizing
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, Dropout
#from keras.utils.np_utils import to_categorical

# Max number of words in each complaint.
MAX_SEQUENCE_LENGTH = 300
# This is fixed.
EMBEDDING_DIM = 300

tokenizer = Tokenizer()
tokenizer.fit_on_texts(X_train.text)
word_index = tokenizer.word_index
vocab_size = len(word_index)
print('Found %s unique tokens.' % len(word_index))

# Convert the data to padded sequences
X_train_padded = tokenizer.texts_to_sequences(X_train.text)
X_train_padded = pad_sequences(X_train_padded, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', X_train_padded.shape)
```

```
Found 232838 unique tokens.
Shape of data tensor: (1280000, 300)
```

Fig 8.8 Tokenizing

8.8 Building the Model

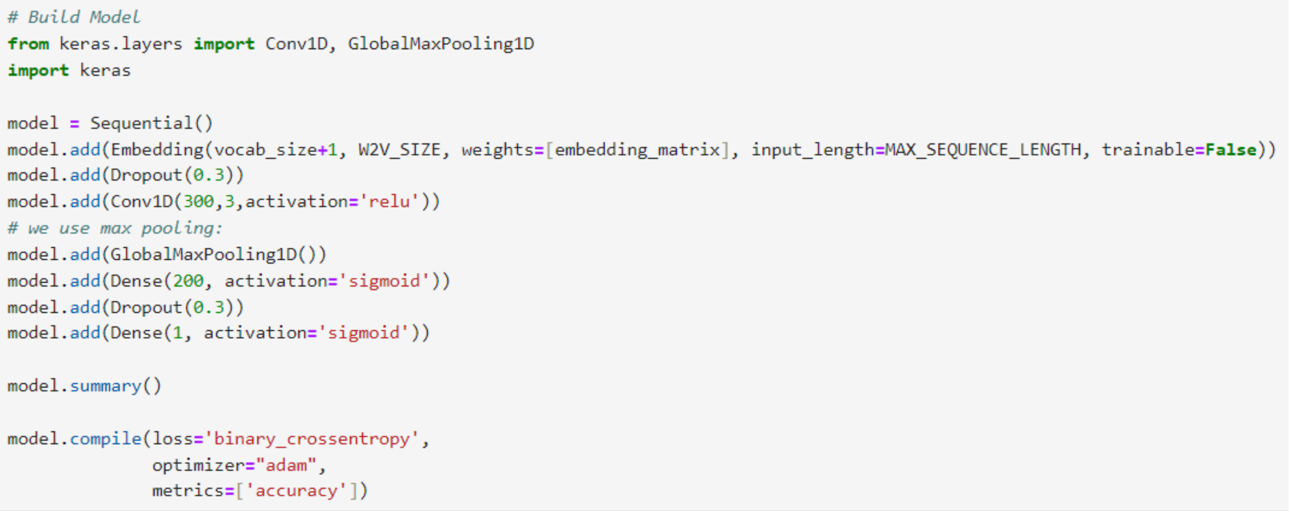
8.8.1 LSTM Model



```python
import keras
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense, Dropout

# Define the model
model = Sequential()
model.add(Embedding(vocab_size+1, W2V_SIZE, weights=[embedding_matrix], input_length=MAX_SEQUENCE_LENGTH, trainable=True))
model.add(Dropout(0.5))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2, return_sequences=True))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(loss='binary_crossentropy',
              optimizer=keras.optimizers.Adam(learning_rate=0.0001),
              metrics=['accuracy'])
```

Fig 8.9 Building the LSTM Model

8.8.2 CNN Model

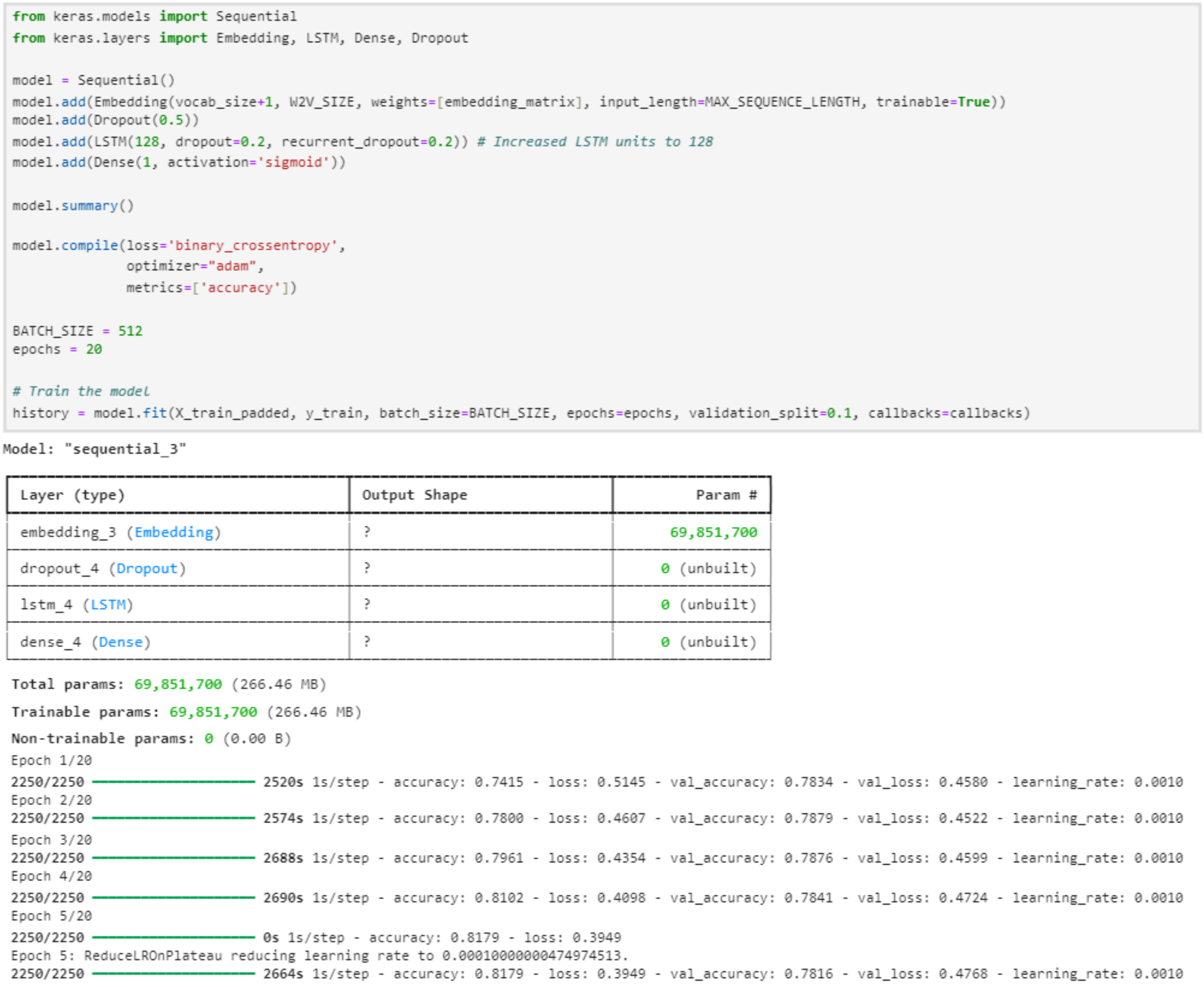


```python
# Build Model
from keras.layers import Conv1D, GlobalMaxPooling1D
import keras

model = Sequential()
model.add(Embedding(vocab_size+1, W2V_SIZE, weights=[embedding_matrix], input_length=MAX_SEQUENCE_LENGTH, trainable=False))
model.add(Dropout(0.3))
model.add(Conv1D(300,3,activation='relu'))
# we use max pooling:
model.add(GlobalMaxPooling1D())
model.add(Dense(200, activation='sigmoid'))
model.add(Dropout(0.3))
model.add(Dense(1, activation='sigmoid'))

model.summary()

model.compile(loss='binary_crossentropy',
              optimizer="adam",
              metrics=['accuracy'])
```

Fig 8.10 Building the CNN Model

8.9 Training the model

8.9.1 LSTM Model

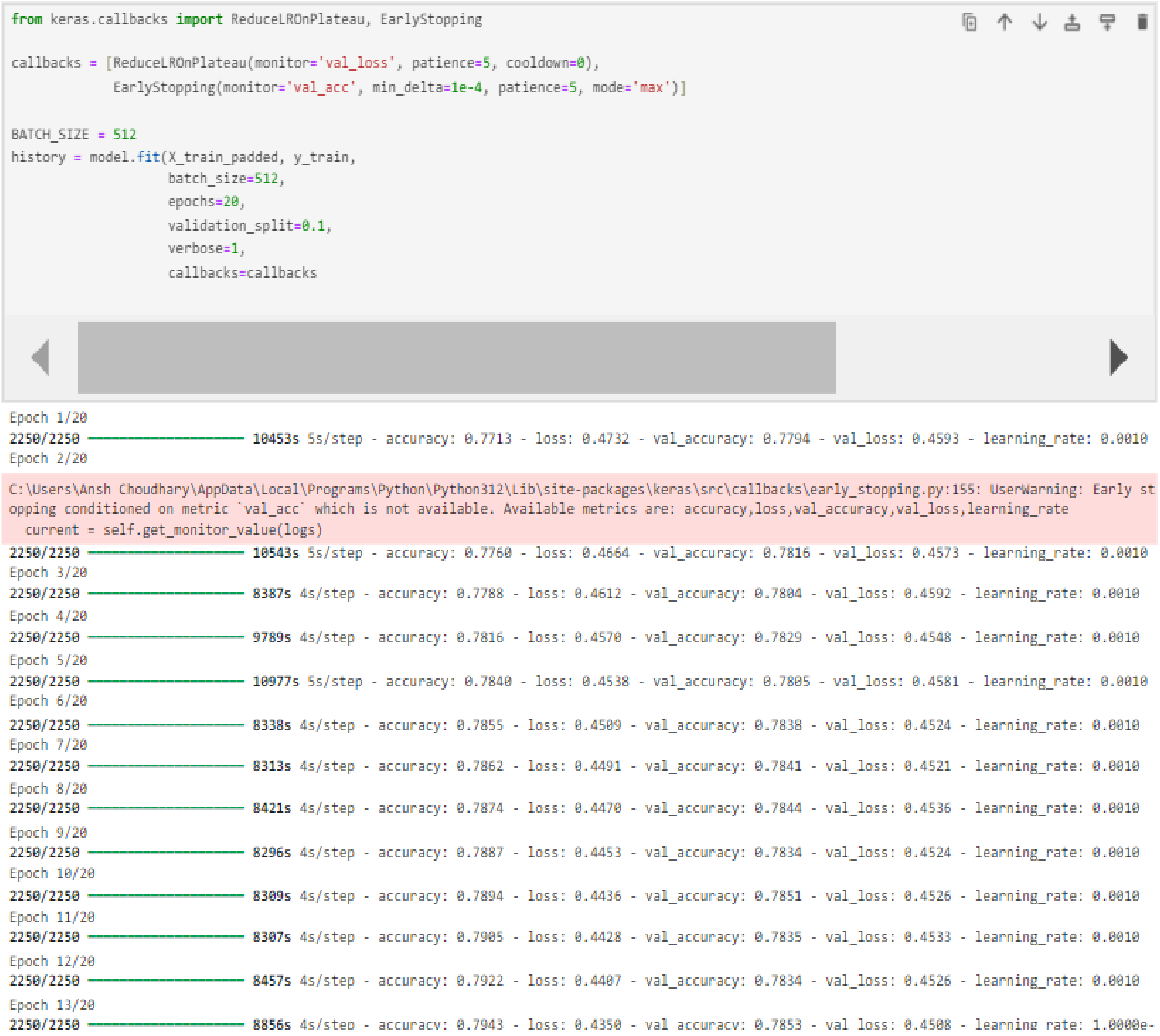


```python
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense, Dropout

model = Sequential()
model.add(Embedding(vocab_size+1, W2V_SIZE, weights=[embedding_matrix], input_length=MAX_SEQUENCE_LENGTH, trainable=True))
model.add(Dropout(0.5))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2)) # Increased LSTM units to 128
model.add(Dense(1, activation='sigmoid'))

model.summary()

model.compile(loss='binary_crossentropy',
              optimizer="adam",
              metrics=['accuracy'])

BATCH_SIZE = 512
epochs = 20

# Train the model
history = model.fit(X_train_padded, y_train, batch_size=BATCH_SIZE, epochs=epochs, validation_split=0.1, callbacks=callbacks)
```

```
Model: "sequential_3"
┌──────────────────────────────────────┬────────────────────────────┬─────────────────┐
│ Layer (type)                         │ Output Shape               │         Param # │
├──────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ embedding_3 (Embedding)              │ ?                          │      69,851,700 │
├──────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ dropout_4 (Dropout)                  │ ?                          │     0 (unbuilt) │
├──────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ lstm_4 (LSTM)                        │ ?                          │     0 (unbuilt) │
├──────────────────────────────────────┼────────────────────────────┼─────────────────┤
│ dense_4 (Dense)                      │ ?                          │     0 (unbuilt) │
└──────────────────────────────────────┴────────────────────────────┴─────────────────┘
 Total params: 69,851,700 (266.46 MB)
 Trainable params: 69,851,700 (266.46 MB)
 Non-trainable params: 0 (0.00 B)
Epoch 1/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 2520s 1s/step - accuracy: 0.7415 - loss: 0.5145 - val_accuracy: 0.7834 - val_loss: 0.4580 - learning_rate: 0.0010
Epoch 2/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 2574s 1s/step - accuracy: 0.7800 - loss: 0.4607 - val_accuracy: 0.7879 - val_loss: 0.4522 - learning_rate: 0.0010
Epoch 3/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 2688s 1s/step - accuracy: 0.7961 - loss: 0.4354 - val_accuracy: 0.7876 - val_loss: 0.4599 - learning_rate: 0.0010
Epoch 4/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 2690s 1s/step - accuracy: 0.8102 - loss: 0.4098 - val_accuracy: 0.7841 - val_loss: 0.4724 - learning_rate: 0.0010
Epoch 5/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.8179 - loss: 0.3949
Epoch 5: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 2664s 1s/step - accuracy: 0.8179 - loss: 0.3949 - val_accuracy: 0.7816 - val_loss: 0.4768 - learning_rate: 0.0010
```

Fig 8.11 Training the LSTM Model

8.9.2 CNN Model



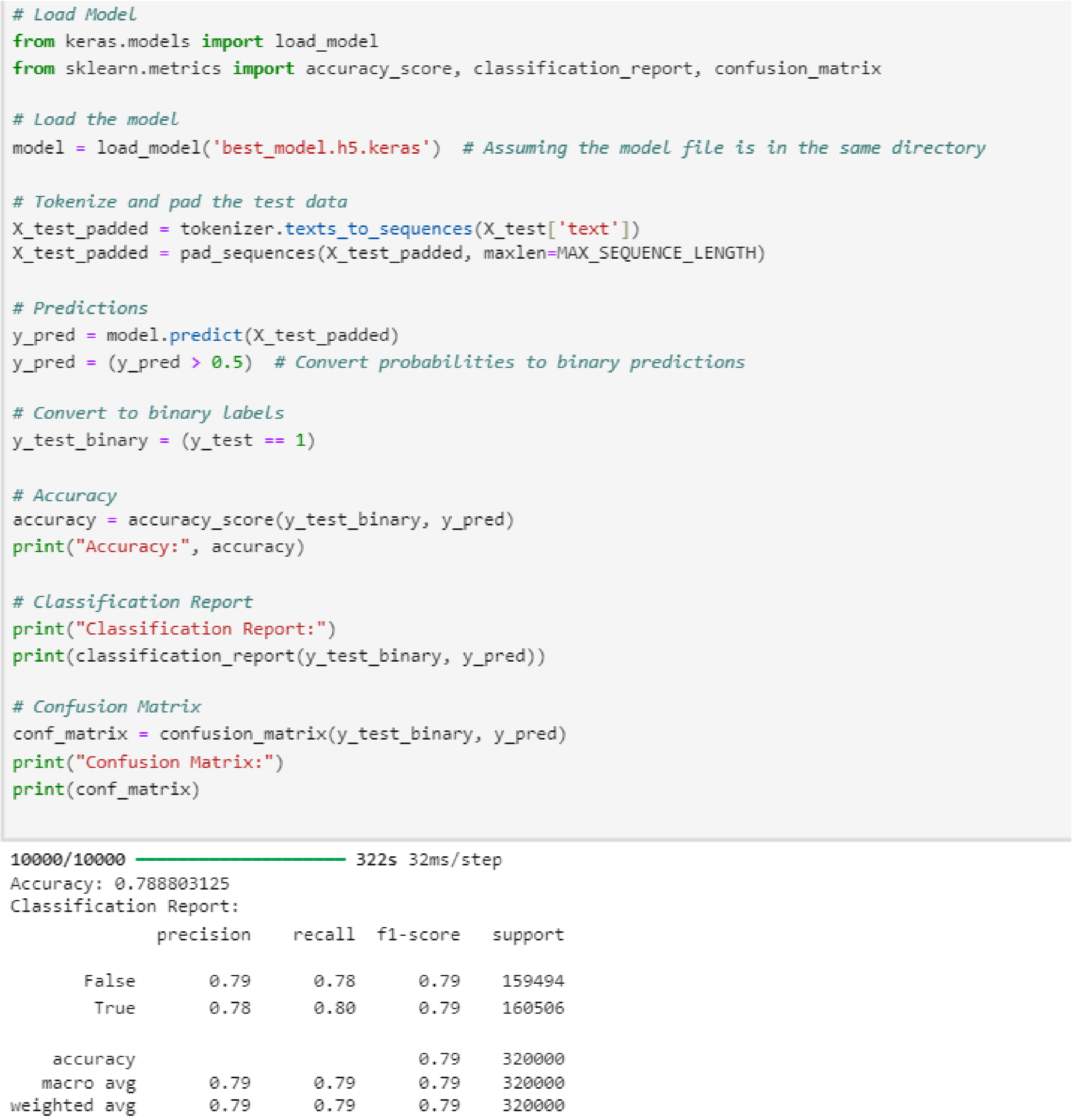
```
from keras.callbacks import ReduceLROnPlateau, EarlyStopping

callbacks = [ReduceLROnPlateau(monitor='val_loss', patience=5, cooldown=0),
             EarlyStopping(monitor='val_acc', min_delta=1e-4, patience=5, mode='max')]

BATCH_SIZE = 512
history = model.fit(X_train_padded, y_train,
                    batch_size=512,
                    epochs=20,
                    validation_split=0.1,
                    verbose=1,
                    callbacks=callbacks
```

```
Epoch 1/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 10453s 5s/step - accuracy: 0.7713 - loss: 0.4732 - val_accuracy: 0.7794 - val_loss: 0.4593 - learning_rate: 0.0010
Epoch 2/20
C:\Users\Ansh Choudhary\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\callbacks\early_stopping.py:155: UserWarning: Early st
opping conditioned on metric `val_acc` which is not available. Available metrics are: accuracy,loss,val_accuracy,val_loss,learning_rate
  current = self.get_monitor_value(logs)
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 10543s 5s/step - accuracy: 0.7760 - loss: 0.4664 - val_accuracy: 0.7816 - val_loss: 0.4573 - learning_rate: 0.0010
Epoch 3/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8387s 4s/step - accuracy: 0.7788 - loss: 0.4612 - val_accuracy: 0.7804 - val_loss: 0.4592 - learning_rate: 0.0010
Epoch 4/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 9789s 4s/step - accuracy: 0.7816 - loss: 0.4570 - val_accuracy: 0.7829 - val_loss: 0.4548 - learning_rate: 0.0010
Epoch 5/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 10977s 5s/step - accuracy: 0.7840 - loss: 0.4538 - val_accuracy: 0.7805 - val_loss: 0.4581 - learning_rate: 0.0010
Epoch 6/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8338s 4s/step - accuracy: 0.7855 - loss: 0.4509 - val_accuracy: 0.7838 - val_loss: 0.4524 - learning_rate: 0.0010
Epoch 7/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8313s 4s/step - accuracy: 0.7862 - loss: 0.4491 - val_accuracy: 0.7841 - val_loss: 0.4521 - learning_rate: 0.0010
Epoch 8/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8421s 4s/step - accuracy: 0.7874 - loss: 0.4470 - val_accuracy: 0.7844 - val_loss: 0.4536 - learning_rate: 0.0010
Epoch 9/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8296s 4s/step - accuracy: 0.7887 - loss: 0.4453 - val_accuracy: 0.7834 - val_loss: 0.4524 - learning_rate: 0.0010
Epoch 10/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8309s 4s/step - accuracy: 0.7894 - loss: 0.4436 - val_accuracy: 0.7851 - val_loss: 0.4526 - learning_rate: 0.0010
Epoch 11/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8307s 4s/step - accuracy: 0.7905 - loss: 0.4428 - val_accuracy: 0.7835 - val_loss: 0.4533 - learning_rate: 0.0010
Epoch 12/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8457s 4s/step - accuracy: 0.7922 - loss: 0.4407 - val_accuracy: 0.7834 - val_loss: 0.4526 - learning_rate: 0.0010
Epoch 13/20
2250/2250 ━━━━━━━━━━━━━━━━━━━━ 8856s 4s/step - accuracy: 0.7943 - loss: 0.4350 - val_accuracy: 0.7853 - val_loss: 0.4508 - learning_rate: 1.0000e-
```

Fig 8.12 Training the CNN Model

8.10 Testing and evaluation of model

8.10.1 LSTM Model

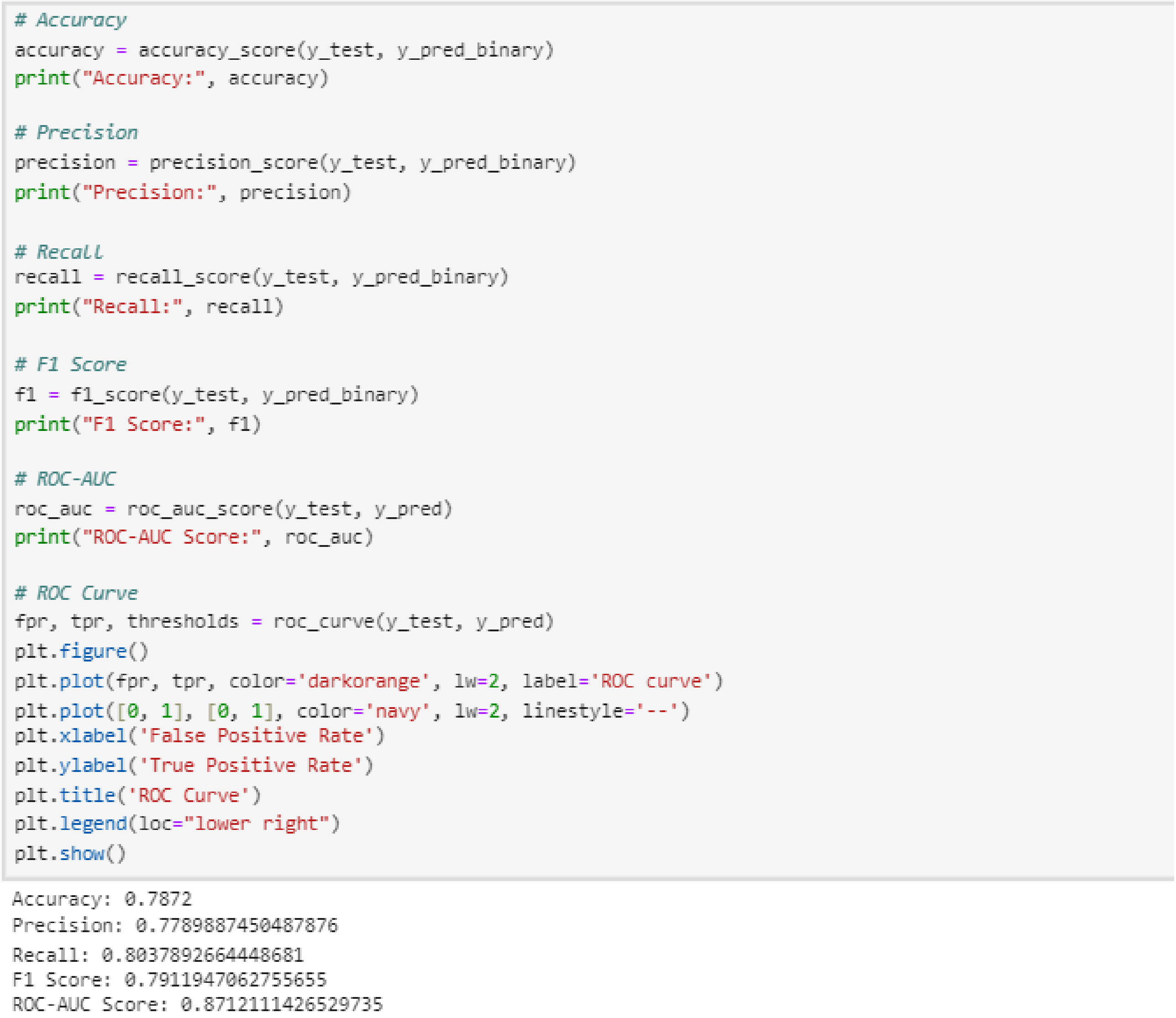


```
# Load Model
from keras.models import load_model
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load the model
model = load_model('best_model.h5.keras')  # Assuming the model file is in the same directory

# Tokenize and pad the test data
X_test_padded = tokenizer.texts_to_sequences(X_test['text'])
X_test_padded = pad_sequences(X_test_padded, maxlen=MAX_SEQUENCE_LENGTH)

# Predictions
y_pred = model.predict(X_test_padded)
y_pred = (y_pred > 0.5)  # Convert probabilities to binary predictions

# Convert to binary labels
y_test_binary = (y_test == 1)

# Accuracy
accuracy = accuracy_score(y_test_binary, y_pred)
print("Accuracy:", accuracy)

# Classification Report
print("Classification Report:")
print(classification_report(y_test_binary, y_pred))

# Confusion Matrix
conf_matrix = confusion_matrix(y_test_binary, y_pred)
print("Confusion Matrix:")
print(conf_matrix)
```

```
10000/10000 ━━━━━━━━━━━━━━━━━━━━ 322s 32ms/step
Accuracy: 0.788803125
Classification Report:
              precision    recall  f1-score   support

       False       0.79      0.78      0.79    159494
        True       0.78      0.80      0.79    160506

    accuracy                           0.79    320000
   macro avg       0.79      0.79      0.79    320000
weighted avg       0.79      0.79      0.79    320000
```

Fig 8.13 Testing and Evaluation of LSTM Model

8.10.2 CNN Model



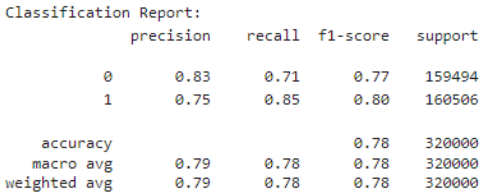
```
# Accuracy
accuracy = accuracy_score(y_test, y_pred_binary)
print("Accuracy:", accuracy)

# Precision
precision = precision_score(y_test, y_pred_binary)
print("Precision:", precision)

# Recall
recall = recall_score(y_test, y_pred_binary)
print("Recall:", recall)

# F1 Score
f1 = f1_score(y_test, y_pred_binary)
print("F1 Score:", f1)

# ROC-AUC
roc_auc = roc_auc_score(y_test, y_pred)
print("ROC-AUC Score:", roc_auc)

# ROC Curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred)
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc="lower right")
plt.show()
```

```
Accuracy: 0.7872
Precision: 0.7789887450487876
Recall: 0.8037892664448681
F1 Score: 0.7911947062755655
ROC-AUC Score: 0.8712111426529735
```

Fig 8.14 Testing and evaluation of CNN Model

8.11 Evaluation of models

8.11.1 LSTM Model



```
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.71      0.77    159494
           1       0.75      0.85      0.80    160506

    accuracy                           0.78    320000
   macro avg       0.79      0.78      0.78    320000
weighted avg       0.79      0.78      0.78    320000
```

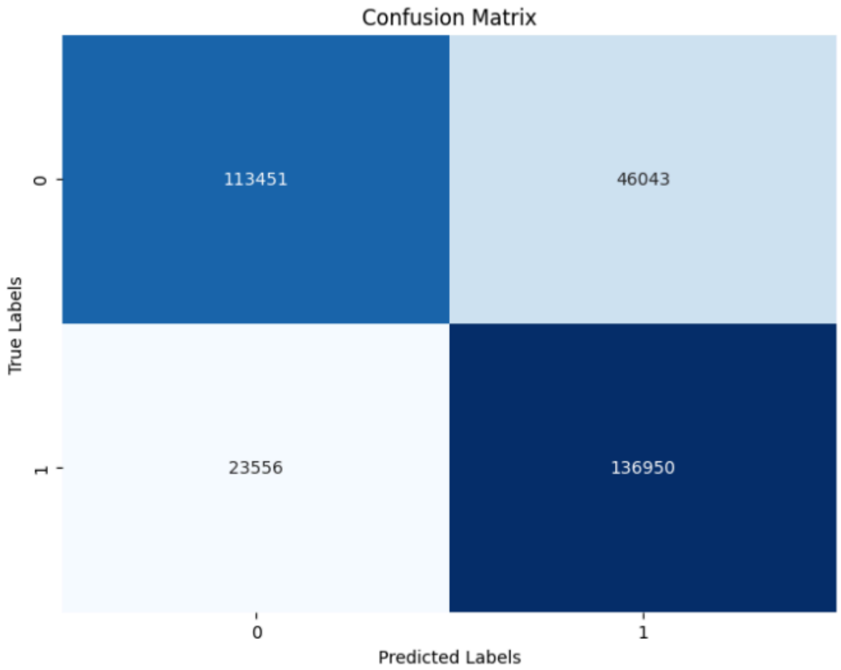


Fig 8.15 Confusion Matrix of LSTM Model

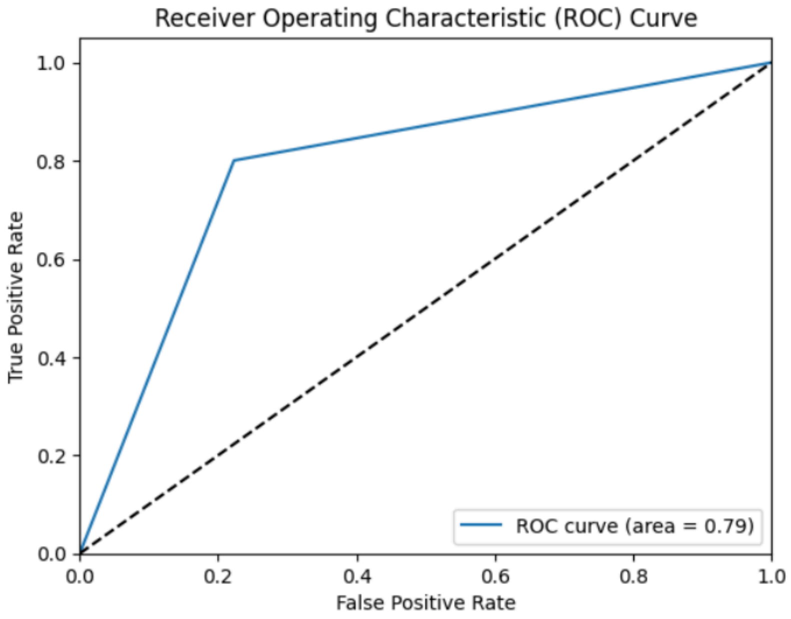


Fig 8.16 ROC-AUC Curve of Trained LSTM Model

8.11.2 CNN Model

Accuracy: 0.7872
Precision: 0.7789887450487876
Recall: 0.8037892664448681
F1 Score: 0.7911947062755655

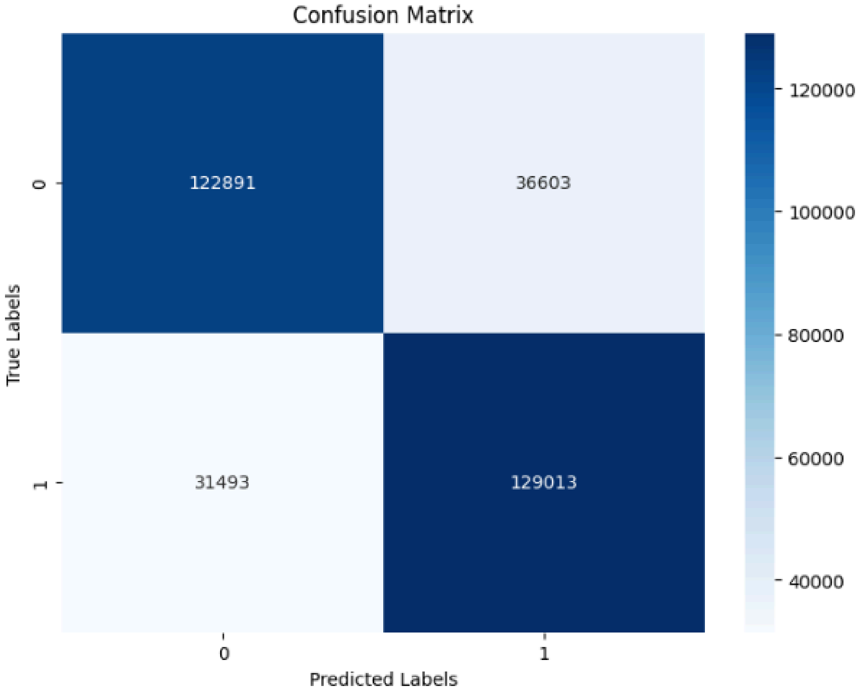


Fig 8.17 Confusion Matrix of CNN Model

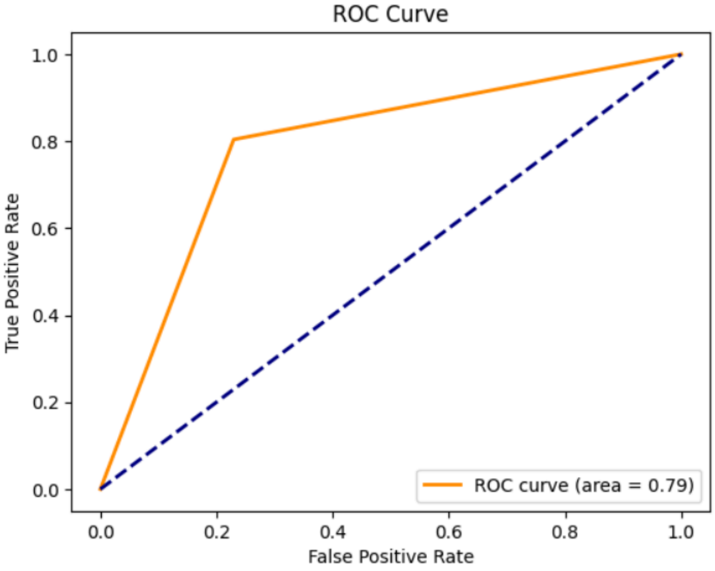


Fig 8.18 ROC-AUC Curve of Trained CNN Model

Chapter 9

Importance of Sentiment Analysis Model

This chapter represents the importance of Sentiment Analysis, Advantages & Disadvantages and its applications..

9.1 Importance of Sentiment Analysis

Sentiment analysis is a fundamental component of data analytics and has a substantial impact on a wide range of industries and applications. Opinion mining is a potent technique that sifts through textual data to reveal the complex web of human mood, emotions, and ideas. Its influence reverberates across a wide range of areas, from political campaign trails and social media platforms to corporate boardrooms and marketing think tanks. It fundamentally transforms how companies view, interact with, and react to their constituents.

Sentiment analysis shines as a light of insight in the busy world of business and marketing, helping decision-makers navigate the maze of customer preferences, market dynamics, and brand perceptions. Through careful analysis of consumer feedback, social media buzz, and online reviews, businesses can obtain important insights into the thoughts and feelings of their target audience. When carefully gleaned and examined, these insights form the basis for well-informed decision-making, which in turn drives product developments, optimizes marketing tactics, and cultivates enduring customer relationships.

Additionally, sentiment analysis protects companies from reputational storms and brand crises by allowing them to gracefully and nimbly withstand the storm of unfavorable sentiment. Companies that actively monitor internet sentiment in real-time can quickly spot developing unrest, spot impending crises, and take preventative action to save their reputations from harm. Organizations can direct their brands towards safer harbors and maintain the trust and loyalty of their stakeholders by navigating the turbulent waters of public opinion with the help of practical insights obtained from sentiment research.

Sentiment analysis, driven by an understanding of human emotions, ushers in a new era of seamless user experiences and tailored engagement in the digital age. Sentiment analysis is used by these intelligent systems—which range from sympathetic chatbots to recommendation engines—to interpret the complex signals hidden in user interactions. These systems create customized experiences that connect with consumers on a very personal level, promoting greater engagement and stronger connections. They do this by distinguishing between delight and frustration, excitement and apathy.

Moreover, sentiment analysis has significant ramifications for political and social debate since it highlights the constantly fluctuating currents of popular opinion and group feeling. Sentiment analysis is a tool used by policymakers, activists, and thought leaders to measure public sentiment and adjust their agendas. It can be used to track sentiment surrounding political campaigns, track social movements, or measure public attitudes toward pressing social issues.

Sentiment analysis is essentially a sentinel at the intersection of digital data and human emotion, providing direction based on insights from the collective awareness of the masses. It is impossible to overestimate the significance of sentiment analysis in interpreting human emotion and producing significant results as societies and organizations work to navigate the turbulent currents of the digital era.

In conclusion, sentiment analysis is essential to data-driven decision-making because it allows businesses to use the strength of human feeling to promote innovation, cultivate client loyalty, and successfully negotiate the challenges posed by the digital world. The significance of sentiment analysis in obtaining actionable knowledge and producing significant results will only increase as businesses depend more and more on data-driven insights to stay competitive.

9.2 Advantages and Disadvantages

Advantages	Disadvantages
Provides actionable insights into customer opinions, preferences, and sentiments, aiding in product development and marketing strategies.	Sentiment analysis may struggle with sarcasm, irony, and nuanced language, leading to inaccuracies in sentiment classification.
Enables organizations to detect and mitigate reputational risks and crises in real-time, safeguarding brand reputation.	Sentiment analysis may face challenges with polysemy where a single word has multiple meanings, leading to misinterpretation.
Helps businesses understand market trends, consumer behavior, and competitive landscape, guiding strategic decision-making.	Privacy concerns may arise due to the analysis of personal data in public forums or social media platforms.

Facilitates personalized customer experiences and targeted marketing campaigns, enhancing customer engagement and loyalty.	Sentiment analysis may be influenced by biases inherent in the data or the algorithms used, leading to skewed results.
Supports sentiment monitoring in diverse domains such as finance, healthcare, politics, and social media, enabling informed decision-making.	Sentiment analysis alone may not provide a comprehensive understanding of context, requiring human oversight and interpretation for nuanced analysis.

Table 9.1 Advantages and Disadvantages

9.3 Application

Sentiment analysis finds applications across various industries and domains, enabling organizations to extract valuable insights from textual data. Some of the key applications of sentiment analysis include:

1. Social Media Monitoring: Sentiment analysis is widely used to track and analyze social media conversations, comments, and posts about brands, products, or events. It helps businesses understand public perception, identify trends, and engage with customers in real-time.

2. Customer Feedback Analysis: Many businesses use sentiment analysis to analyze customer feedback from surveys, reviews, and customer support interactions. By categorizing feedback into positive, negative, or neutral sentiments, organizations can identify areas for improvement, address customer concerns, and enhance customer satisfaction.

3. Brand Reputation Management: Sentiment analysis allows companies to monitor online mentions and discussions related to their brand or products. By identifying positive and negative sentiments, businesses can proactively manage their brand reputation, address negative publicity, and leverage positive feedback for marketing purposes.

4. Market Research: Sentiment analysis helps market researchers gain insights into consumer preferences, behaviors, and trends. By analyzing online discussions, product reviews, and social media conversations, researchers can identify emerging market trends, assess brand perception, and evaluate competitive positioning.

5. Financial Analysis: Sentiment analysis is used in the financial industry to analyze news articles, social media posts, and market sentiment to predict stock price movements, assess investor sentiment, and identify market trends. It helps investors make informed decisions and manage investment portfolios more effectively.

6. Product Development: Sentiment analysis provides valuable feedback to product development teams by analyzing customer opinions and preferences. By understanding customer sentiments towards existing products or features, organizations can identify areas for innovation, prioritize product enhancements, and develop new products that better align with customer needs.

7. Customer Service Optimization: Sentiment analysis is integrated into customer service platforms to analyze customer interactions and sentiment in real-time. By automatically categorizing customer inquiries and sentiment, organizations can prioritize and route inquiries more effectively, identify recurring issues, and improve overall customer service efficiency.

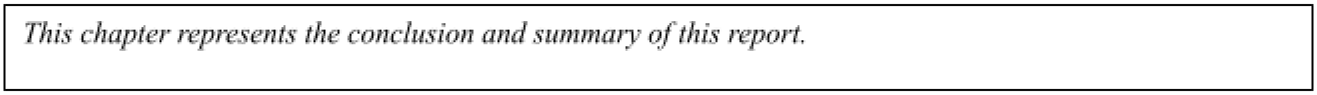
8. Political Analysis: Sentiment analysis is used in political campaigns and governance to analyze public opinion, sentiment, and reactions towards political candidates, policies, and events. It helps political parties understand voter sentiment, tailor their messaging, and gauge public perception to inform campaign strategies.

9. Healthcare: Sentiment analysis is applied in healthcare to analyze patient feedback, reviews, and social media discussions related to healthcare services, medications, and treatments. It helps healthcare providers identify patient satisfaction levels, address concerns, and improve overall patient experience.

10. Voice of the Customer (VoC) Analysis: Sentiment analysis is a key component of Voice of the Customer programs, where organizations collect and analyze customer feedback from various sources. By understanding customer sentiment across different touchpoints, organizations can drive customer-centric improvements and enhance overall customer experience.

Chapter 10

Conclusion


This chapter represents the conclusion and summary of this report.

In Conclusion, our effort to do sentiment analysis on Twitter data with CNN and LSTM models is a noteworthy advancement in the field of artificial intelligence and natural language processing. By combining state-of-the-art deep learning architectures, we have developed a powerful framework that can extract and analyze sentiment from massive textual data sets on social networking sites like Twitter.

Sentiment analysis is important because it can provide priceless insights into customer sentiment, public opinion, and societal trends. Our work advances this subject by introducing a novel method that combines CNN's effective feature extraction capabilities with LSTM's superior temporal understanding capabilities.

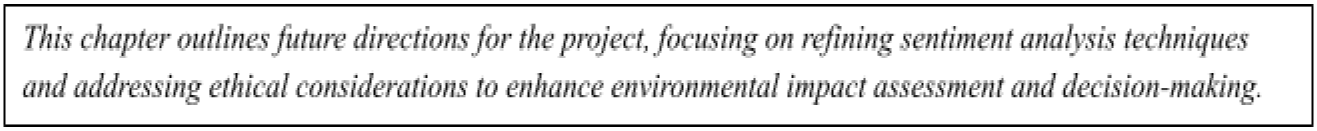
Through the use of LSTM, our model is able to capture contextual subtleties and temporal relationships that are encoded within textual data, allowing for a more sophisticated and perceptive study of sentiment over time. Concurrently, the addition of CNN improves the model's ability to distinguish by effectively removing important characteristics and patterns from the intricate textual inputs.

Furthermore, our project has a wide range of applications in areas including public opinion tracking, brand management, and market research. Organizations and legislators can make educated judgments, customize their communication plans, and quickly adapt to changing trends and attitudes when they have a clear grasp of the sentiment revealed by Twitter data.

In the future, we see sentiment analysis as a vital tool for understanding and impacting public conversation, driving constructive change, and encouraging increased accountability and transparency in the digital sphere. We are unwavering in our resolve to realize this vision and fully utilize sentiment analysis for the benefit of society as a whole through unrelenting innovation, cooperative synergy, and moral stewardship.

Chapter 11

Future Scope



This chapter outlines future directions for the project, focusing on refining sentiment analysis techniques and addressing ethical considerations to enhance environmental impact assessment and decision-making.

The sentiment analysis of Twitter tweets presents a vast field for exploration and improvement. While the current implementation provides valuable insights into the sentiment trends of Twitter users, there are several avenues for future development and enhancement.

1. Fine-grained Sentiment Analysis

Currently, the sentiment analysis model categorizes tweets into broad categories such as positive, negative, or neutral sentiments. However, refining the analysis to recognize more nuanced emotions and sentiments could provide richer insights. This could involve training the model to identify specific emotions like joy, sadness, anger, or fear, enabling a more detailed understanding of users' sentiments.

2. Contextual Analysis

Incorporating contextual information into the sentiment analysis process can significantly enhance the accuracy of sentiment classification. Future iterations of the project could explore techniques to consider the context of tweets, such as the topics being discussed, the sentiment of linked articles or websites, and the user's historical tweets. Contextual analysis could help disambiguate sentiments expressed in sarcasm, irony, or colloquial language, leading to more accurate sentiment classification.

3. Multimodal Sentiment Analysis

With the increasing popularity of multimedia content on Twitter, including images, videos, and emojis, extending the sentiment analysis to incorporate multimodal data could provide a more comprehensive understanding of user sentiments. Integrating techniques from computer vision and natural language processing to analyze visual and textual elements together could yield deeper insights into users' emotions and sentiments expressed through diverse media types.

4. Real-time Analysis and Monitoring

Implementing real-time sentiment analysis capabilities would enable monitoring of sentiment trends on Twitter as they unfold. This could involve developing efficient algorithms and infrastructure to process incoming tweets in real-time, allowing for immediate detection and analysis of emerging sentiment patterns, events, or crises. Real-time monitoring could be particularly valuable for businesses, brands, and policymakers to respond promptly to changes in public sentiment.

5. User-level Analysis and Personalization

Tailoring sentiment analysis to individual Twitter users could offer personalized insights into their sentiments and preferences. By analyzing a user's past tweets, interactions, and profile information, personalized sentiment analysis could provide recommendations, content suggestions, or targeted interventions based on the user's unique sentiment profile. This approach could be especially beneficial for marketing campaigns, customer engagement strategies, and mental health support initiatives.

6. Cross-lingual and Multicultural Analysis

Expanding the sentiment analysis capabilities to support multiple languages and cultures could broaden the applicability and impact of the project. Cross-lingual sentiment analysis would enable the analysis of tweets in languages other than English, facilitating a more global understanding of sentiments and opinions across diverse linguistic communities. Additionally, incorporating cultural nuances and differences in sentiment expression could improve the accuracy and relevance of sentiment analysis for users worldwide.

7. Ethical Considerations and Bias Mitigation

As with any data-driven project, it is essential to address ethical considerations and potential biases in sentiment analysis. Future iterations of the project should prioritize fairness, transparency, and accountability in the analysis process. This could involve implementing techniques to detect and mitigate biases in training data, ensuring representativeness across diverse demographic groups, and providing transparent explanations for the model's predictions. Additionally, incorporating mechanisms for user consent, data privacy, and responsible data usage would uphold ethical standards and promote trust among users and stakeholders.

In conclusion, the sentiment analysis of Twitter tweets holds immense potential for further development and innovation. By exploring the outlined avenues for future research and enhancement, the project can continue to advance the state-of-the-art in sentiment analysis, providing valuable insights into user sentiments, behaviors, and preferences on one of the world's largest social media platforms.

References

Papers:

1) I. Gupta and N. Joshi, &quot;Feature-Based Twitter Sentiment Analysis With Improved Negation Handling,&quot; in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 917-927, Aug. 2021, doi: 10.1109/TCSS.2021.3069413.

2) V. Prakruthi, D. Sindhu and D. S. Anupama Kumar, &quot;Real Time Sentiment Analysis Of Twitter Posts,&quot; 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2018, pp. 29-34, doi: 10.1109/CSITSS.2018.8768774.

3) P. Gupta, S. Kumar, R. R. Suman and V. Kumar, &quot;Sentiment Analysis of Lockdown in India During COVID-19: A Case Study on Twitter,&quot; in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 992-1002, Aug. 2021, doi: 10.1109/TCSS.2020.3042446.

4) A. Krouska, C. Troussas and M. Virvou, &quot;The effect of preprocessing techniques on Twitter sentiment analysis,&quot; 2016 7th International Conference on Information, Intelligence, Systems &amp; Applications (IISA), Chalkidiki, Greece, 2016, pp. 1-5, doi: 10.1109/IISA.2016.7785373.

5) M. Wongkar and A. Angdresey, &quot;Sentiment Analysis Using Naive Bayes Algorithm Of The Data Crawler: Twitter,&quot; 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-5, doi: 10.1109/ICIC47613.2019.8985884.

6) L. Wang, J. Niu and S. Yu, &quot;SentiDiff: Combining Textual Information and

Sentiment Diffusion Patterns for Twitter Sentiment Analysis,&quot; in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 10, pp. 2026-2039, 1 Oct. 2020, doi: 10.1109/TKDE.2019.2913641.