Sentiment Analysis Employing LSTM (Long Short-Term Memory) for Binary Classification of Social Media Texts

Pournima Anil Kharade
Shivajirao S. Jondhale College of Engineering,
Department of Information Technology

Savita Sangam, PhD
Head of Department,
Information Technology

ABSTRACT
Sentiment analysis is a domain within Natural Language Processing (NLP) focused on the computer detection and classification of opinions in textual data. The expansion of social media platforms, such as Twitter, Facebook, and Instagram, has resulted in an increase in user-generated content that reflects public opinion on various issues, including films, products, and political events, daily. An examination of this information would benefit firms, lawmakers, and other stakeholders by aiding in the evaluation of public perception. The traditional method of sentiment analysis relies on a rule-based framework and fundamental machine learning algorithms, including Naive Bayes, Support Vector Machines (SVM), and Logistic Regression (LR). These solutions generally necessitate manually generated features and face challenges in capturing more profound linkages and dependencies within the text. Recent advancements in deep learning, particularly in Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, have transformed the domain of sentiment analysis by offering models for the unsupervised learning of associations from raw text data. As a variation of RNN, LSTM addresses the vanishing gradient problem, making it more suitable for tasks requiring longer dependencies, such as sentiment analysis. This study investigates the application of LSTM networks for binary sentiment categorisation utilising the IMDb movie review dataset. The model's learning capabilities were enhanced by using pre-trained word embeddings (GloVe), which illustrate semantic relationships among words and augment the model's contextual comprehension.

Keywords
Sentiment Evaluation, Long Short-Term Memory (LSTM), Social Media Platforms, Dichotomous Classification, Global Vectors for Word Representation, Natural Language Processing

1. INTRODUCTION
Opinion mining, also referred to as sentiment analysis, is a subfield of Natural Language Processing (NLP) that examines the computer job of identifying and categorising views from textual data. Extensive social networks such as Twitter, Facebook, and Instagram facilitate the daily upload of vast quantities of information from users, frequently reflecting public sentiment on diverse subjects, including films, products, and political events. This analysis of the material will be highly beneficial for business sectors, lawmakers, and other stakeholders seeking to evaluate public mood. Traditional sentiment analysis techniques depend on rule-based frameworks and fundamental machine learning algorithms, including Naive Bayes, Support Vector Machines (SVM), and Logistic Regression. These methodologies generally necessitate manually designed features and struggle to comprehend intricate relationships and dependencies within the text. Recent breakthroughs in deep learning, especially in Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, have transformed sentiment analysis by providing models for the automated learning of associations from raw text data. LSTM networks, a kind of RNNs, were designed to mitigate the challenges associated with normal RNNs caused by the vanishing gradient problem, making them more appropriate for tasks involving long-range dependencies, such as sentiment analysis. The use of LSTM networks in binary sentiment categorisation was examined using the IMDb movie review dataset. The model's performance was further validated by improving its efficiency through the utilisation of pre-trained word embeddings (GloVe), which represent word meanings and provide contextual intelligence for the model.

2. RELEVANT LITERATURE
Sentiment analysis has evolved into a flourishing domain, transitioning from rule-based approaches to sophisticated deep learning frameworks. The initial methodologies for sentiment analysis depended on lexicons and heuristic approaches, such as SentiWordNet (Baccianella et al., 2010), which offered compilations of words annotated with sentiment classifications. Rule-based techniques struggle to accommodate the nuances and intricacies of human language, particularly when faced with sarcasm, idiomatic idioms, or context-dependent meanings. Statistical machine learning methodologies, such as Naive Bayes and Support Vector Machines (SVM), offer a sophisticated technique by employing statistical characteristics derived from textual data for analogous instances. Nonetheless, these models face difficulties in acknowledging the sequential characteristics of text, which are essential for comprehending sentiments. Conversely, deep learning models, particularly Recurrent Neural Networks (RNNs), are recognised for their remarkable capabilities in understanding word associations within a sequence. Conversely, conventional RNNs experience the vanishing gradient problem, which complicates training on extended sequences. Long Short-Term Memory (LSTM) networks, developed by Hochreiter and Schmidhuber (1997), address the constraints of conventional RNNs by utilising memory cells that preserve information over extended durations. LSTM networks have become the preferred option for various NLP tasks, including sentiment analysis, due to their proficiency in modelling long-term dependencies in texts. The relevance of LSTM networks for sentiment classification tasks is demonstrated in research by Kim (2014) and Maas et al. (2011), among others. More recent studies, such as those by Varadharajan et al. (2025) on IMDb review classification and Alasmari et al. (2024) employing CNN-LSTM hybrid approaches, further underscore the continued advancements and efficacy of deep learning, particularly LSTM-based models, in sentiment analysis. Moreover, the efficacy of sentiment analysis models has been enhanced through the utilisation of pretrained word embeddings such as Word2Vec

(Mikolov et al., 2013) and GloVe (Pennington et al., 2014), which provide superior semantic representation of words.

Overview of Sentiment Analysis

This section elucidates the methodologies employed for the project. It provides a comprehensive discussion of key ideas essential to the artefact's realisation:

- Text Mining
- Natural Language Processing
- Machine Learning

2.1 Text Mining

Text mining involves the analysis of data presented in natural language formats, such as Twitter messages. It is the process of extracting valuable information from unstructured textual sources. This discipline spans various domains, including biomedical research, marketing, and sentiment analysis. In the context of marketing, text mining is essential for customer relationship management, facilitating the understanding of customer feedback and aiding analysts in developing improved predictive analytics models for customer retention. The objective of text mining is to transform data into an analyzable structured format, utilising Natural Language Processing and other analytical methods. Information Extraction (IE) is particularly significant in text mining as applied to this study. Consequently, the following sections will address issues and terminology related to IE and subsequent processing.

2.2 Natural Language Processing (NLP)

Twitter serves as a case study for the ongoing demonstration of these concepts. The data obtained from Twitter is semi-structured, as tweets are constrained to 140 characters. Consequently, understanding the structures on a snippet-by-snippet basis is relatively straightforward. However, tweets may adhere to no formal structure or may be ungrammatical. The text may include abbreviations, slang, alternative grammatical constructions to express identical meanings, and various forms of syntactic manipulation.

2.2.1 Tokenization

A further step remains, and it requires all the data to be tokenised into smaller pieces before any processing can happen, called Tokenisation in NLP. At a higher level, the text gives the division into paragraphs and sentences. Given that tweets do not normally exceed a single paragraph due to the 140-character limitation, this project, at this stage, seeks to identify sentence formation accurately by interpreting punctuation such as ".". The next stage that follows this is extracting words/tokens from these sentences. Another challenge at this stage is the orthography management within the sentences: this means correcting spelling mistakes, removing URLs, and removing punctuation from such resultant token sets.

2.2.2 Part of Speech labelling POS

It implies representing all words syntactically by adding specific syntactic functions to each word to comprehend the meaning of the sentence. It assists in doing the critical N-gram selection lemmatisation.

2.2.3 Stemming and Lemmatisation

Both are methods that can be used to reduce the different word forms to a common multiple. A few examples of such words are 'Connection', 'Connections', 'Connective', 'Connected', and 'Connecting', which all connect by the root word connect. Stemming is a plain heuristic method that achieves base form by removing word endings. While lemma building is a morphological analysis through which a dictionary form or base is derived from words. This process relies on dictionary availability in the less complex morphological language like English. Lemmatisation also leads to a few ambiguities as it may provide more than one potential lemma for a word or may choose a wrong lemma among the competing possibilities; for example, the consideration of 'axes' as the plural of 'axe' or of 'axis'.

2.2.4 N-Grams

N-Grams is a widely used technique in Text Mining that involves generating subsets of words of length N within a sentence. For example, given the line "This is a six-word sentence! The subsequent N-Grams can be produced:

Unigrams: "This," "is," "a," "six," "word," "sentence."2-Grams (Bigrams): "This is," "is a," "a six," "six-word," "word sentence."3-Grams (Trigrams): "This is a," "is a six," "a six-word," "six-word sentence"

Consequently, the example sentence yields 6 unigrams, 5 bigrams, and 4 trigrams. In more extensive datasets, the generation of bigrams and trigrams can substantially augment the dataset's size, thereby hindering processing speed.

2.3 Classification in Machine Learning

This section discusses machine learning algorithms utilised for classifying the polarity of rectified reviews as either positive or negative. Machine learning focuses on the automated identification of significant patterns within data. Given the exponential increase in data generation, machine learning has gained widespread acceptance for information extraction purposes. Its applications range from spam detection to targeted advertising, as well as search engines and facial recognition technology. The selection of an algorithm is contingent upon the specific learning task, with specialised literature differentiating between supervised and unsupervised machine learning algorithms.

Supervised Learning:

A supervised machine learning methodology employs a training dataset comprising input vectors and corresponding target vectors (classes) for the training process. For instance, supervised learning can develop a program to differentiate between images of dogs and cats. During the training phase, the algorithm analyses a collection of labelled images, enabling the computer to identify which images depict cats or dogs. When confronted with new, unlabeled images, the algorithm will categorise them based on the knowledge it has acquired, endeavouring to formulate a general principle for mapping inputs to outputs.

Unsupervised Learning:

Both serve the same objective but employ distinct methodologies. Unsupervised learning algorithms must operate without labelled data in their training set, as all input data remains unlabeled. Consequently, the computer must identify patterns and structures within the input autonomously, without explicit direction regarding what to seek.

3. METHODOLOGY

3.1 Data Acquisition and Preprocessing

This research utilises the IMDb movie review dataset, comprising 50,000 annotated reviews: 25,000 for training and 25,000 for testing. The reviews are categorised as either positive or negative, constituting a binary classification task. Extensive preprocessing procedures are conducted to clean and normalise the text data for model ingestion. Tokenisation: It involves the decomposition of text into individual words or sub-words, representing a crucial phase in Natural Language Processing, during which the model analyses discrete elements of the text. Exclusion of Stop-Words: The list of significant terms for sentiment analysis did not encompass frequent words such as "the," "on," "is," and "in." Text Normalisation: Converting all text data to lowercase to achieve uniformity. Padding: Subsequently, the reviews are padded to a uniform length (for instance, 200 words) due to the variability in review lengths. This ensures uniformity, thereby preserving consistency during the training process. Word Embeddings: GloVe embeddings (Pennington et al., 2014) are utilised for word representation. These embeddings are defined by highly dense vector representations that encapsulate semantic links among words.

3.2 Architectural Framework

Embedding Layer: This layer transforms the text into word vectors using GloVe embeddings, substituting each word in the input with its associated vector. LSTM Layer: An LSTM layer with 128 units is optimally configured for text analysis. LSTM effectively captures long-range dependencies in text, which is crucial for sentiment analysis tasks. Dropout Layer: The dropout rate between LSTM layers is maintained at 0.2 to mitigate model overfitting. Fully Connected Layer: The output of sequence processing must pass through a fully linked (dense) layer utilising a ReLU activation function. Result Layer: The conclusive result is a binary prediction indicating either positive or negative sentiment, accomplished by a sigmoid activation function.

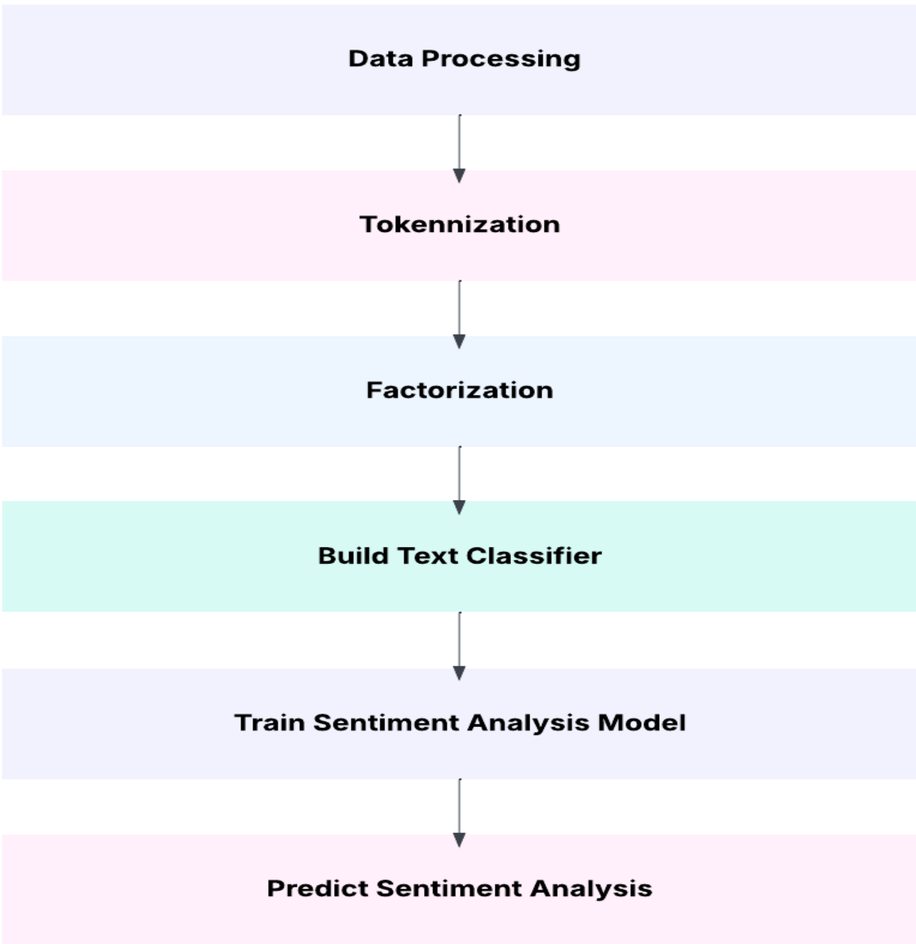
3.3 Model Training and Assessment

The model is optimised using the Adam algorithm with a learning rate of 0.001 and employs binary cross-entropy as the loss function. The dataset is partitioned into training (80%) and testing (20%) subsets. Training is conducted with a batch size of 64 over 10 epochs. Early stopping is implemented to mitigate overfitting and enhance the model's generalisation to novel data. The model's performance is assessed using the following metrics: Accuracy: the proportion of correct predictions made by the model. Precision: the ratio of correct positive forecasts to the total number of predictions made. Recall: Recall is defined as the ratio of correctly identified actual positive samples to the total number of positive samples. F1-score: the harmonic mean of precision and recall, serving as a singular metric that balances both measures. The cross-validation method employed to assess the model's robustness is presented using average results across the folds.

3.4 Assessments and Outcomes

Python Sentiment Analysis is a method for examining text to ascertain its underlying sentiment. This is accomplished using elements from Machine Learning and Natural Language Processing (NLP). Consequently, Sentiment Analysis aids in assessing public sentiment regarding a written work.

Data Processing



A binary sentiment text classifier will be developed in this machine learning project. Various NLP preprocessing approaches will be employed to cleanse the data, and LSTM layers will be utilised to construct the text classifier.

Dataset for Sentiment Analysis in Python:

The collection has about 50,000 movie data samples categorised into two types:

- Affirmative
- Adverse

Utilised Tools and Libraries:

- Python — Version 3.
- X Pandas – version 1.2.4
- Matplotlib — version 3.3.4
- TensorFlow – 2.4.1

Procedure for Constructing a Sentiment Analysis Text Classifier in Python:
Data Preprocessing:

Given that textual material is being handled, it is necessary to preprocess it with word embeddings.

Dataset

Based on the content of the reviews, this dataset contains 50,000 movie reviews that have been pre-labelled with "GOOD" and "NEGATIVE" sentiment class labels.

Data Importation and Analysis

Data is imported and the initial entries of the dataset are displayed.

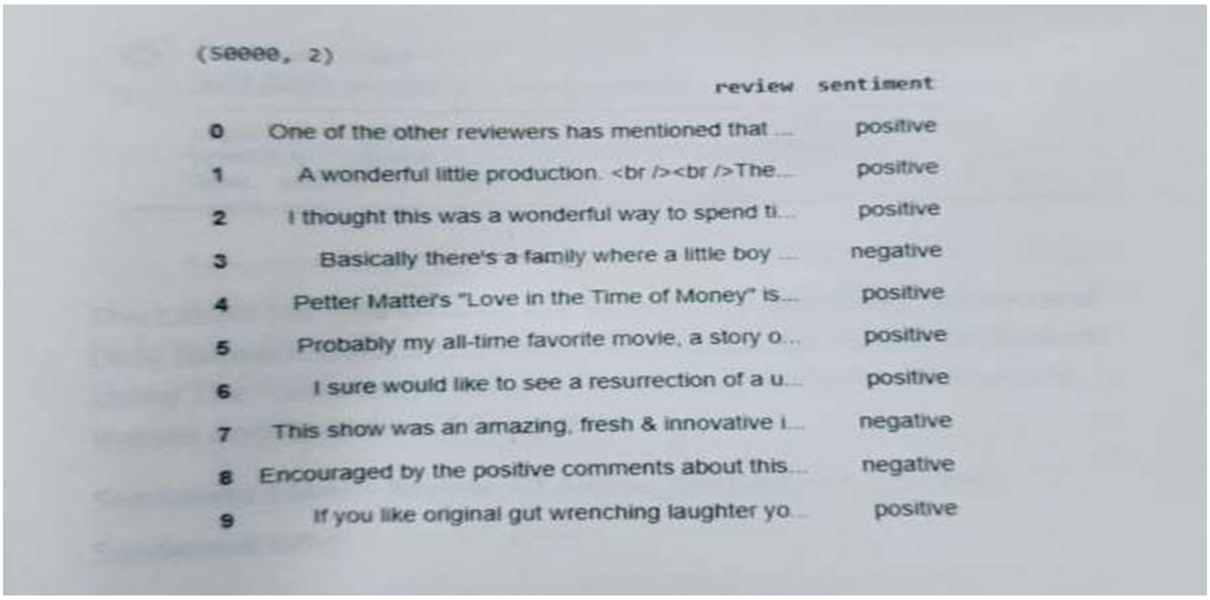
Data Importation and Printing Mechanism:

Imdb_Data = pd.read_csv('/content/drive/mydrive/ai/imdb')

Dataset.csv

Display (Imdb_Data.Dimensions)

Imdb_Data.First(10)

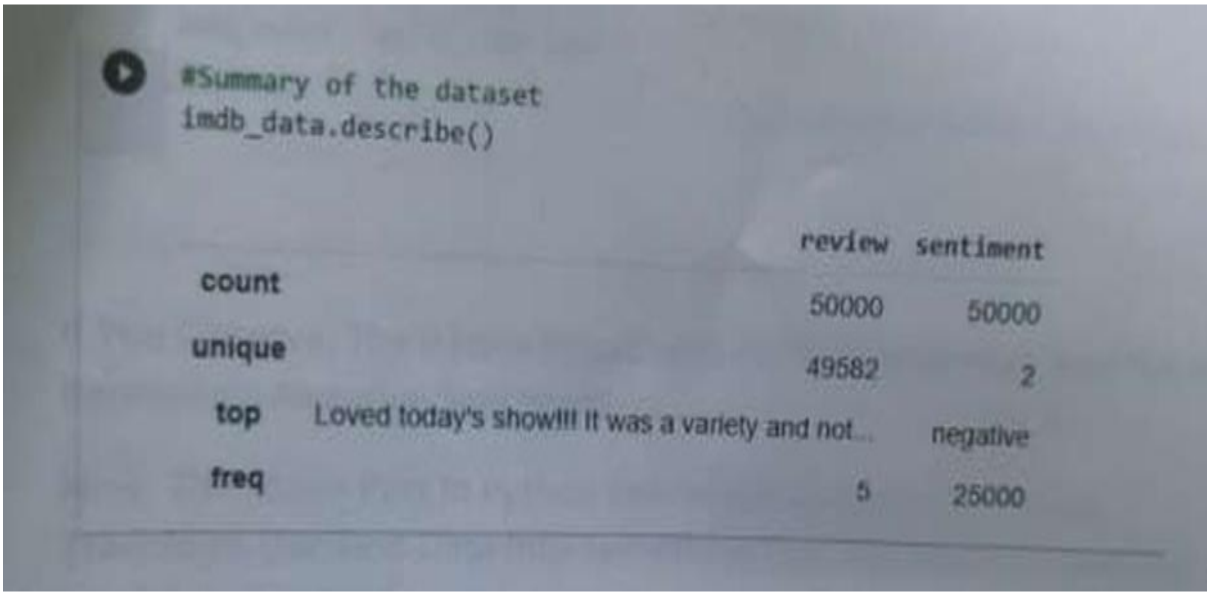


(50000, 2)

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
5	Probably my all-time favorite movie, a story o...	positive
6	I sure would like to see a resurrection of a u...	positive
7	This show was an amazing, fresh & innovative i...	negative
8	Encouraged by the positive comments about this...	negative
9	If you like original gut wrenching laughter yo...	positive

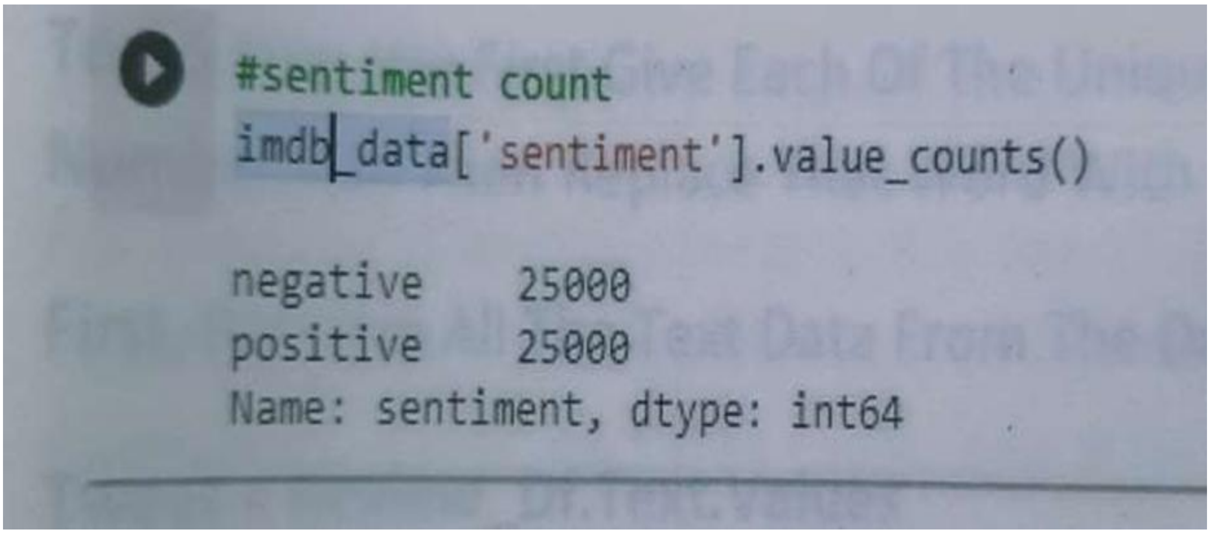
Data Description and Sentiment Analysis Count

The dataset is examined to comprehend the distribution of attitudes.



```
#Summary of the dataset
imdb_data.describe()
```

	review	sentiment
count	50000	50000
unique	49582	2
top	Loved today's show!!! It was a variety and not...	negative
freq	5	25000



The dataset's labels are categorical. Since machines can only process numeric data, the factorize() technique is utilised to convert the categorical values into numeric form. This method yields an array of numeric values together with an index of categories.

Sentiment_Label = Review_Df.Airline_Sentiment.factorize()

Sentiment Classification

Positive sentiment is represented by 0, while negative sentiment is represented by 1. A crucial aspect of sentiment analysis using Python is the conversion of text data into a format recognisable by machine learning models. This involves transforming text into an array of vector embeddings. Word embeddings serve as a method to illustrate the relationships between words within the text.

Initially, all textual data is extracted from the dataset.

Tweet equals Review_Df.Text.values

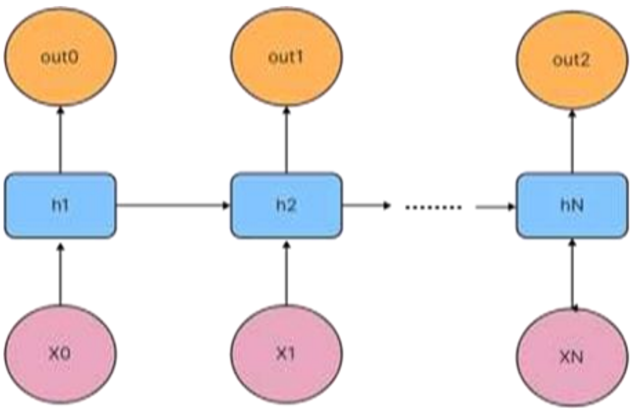
Before advancing the Python sentiment analysis project, all terms in the text are tokenised utilising the Tokeniser. Tokenisation is dividing the entire words or sentences of a text into smaller components known as tokens.

Python

from tensorflow.keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(num_words=5000)

tokenizer.fit_on_texts(Tweets)



Train the Sentiment Analysis Model:

The **fit_on_texts()** function generates an association between the words and the allocated numbers. This association is kept in the form of a dictionary in the Tokeniser's **word_index property**. Now, the words are replaced with their assigned numbers using the **texts_to_sequences()** method.

Python

encoded_docs = tokenizer.texts_to_sequences(Tweet)

Each sentence in the dataset varies in length. Padding is employed to standardise the sentences to a uniform length.

Python

from tensorflow.keras.preprocessing.sequence import pad_sequences

padded_sequence = pad_sequences(encoded_docs, maxlen=200)

Construct the Text Classifier

The sentiment analysis project utilises LSTM layers within the model architecture for machine learning applications. The architecture consists of an embedding layer, a single LSTM layer, and a dense layer at the conclusion. Dropout mechanisms are implemented between the LSTM layers to mitigate overfitting. LSTMs derive a distinct type of memory network from their short-term and long-term capabilities. They serve as an enhancement to recurrent networks, which systematically process sequential data flows, such as text or audio.

$$out_t = W_{ho}h_t + b_o$$

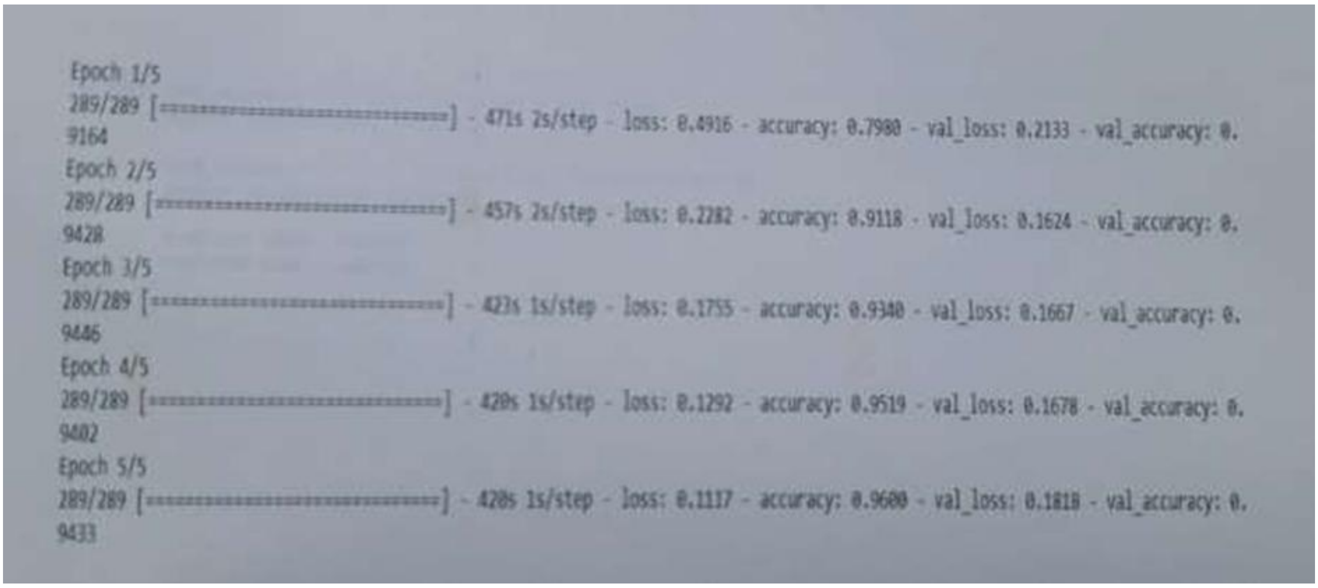
$$h_t = \tanh(W_{xh}X_t + W_{hh}h_t - 1 + b_h)$$

The sentiment analysis model is trained for five epochs on the entire dataset with a batch size of 32 and a validation split of 20%.

Python

history = model.fit(padded_sequence, Sentiment_Label[0], validation_split=0.2, epochs=5, batch_size=32)

The sentiment analysis model is then implemented. A function that accepts text as input and produces its corresponding prediction label as output is defined.



```
Epoch 1/5
289/289 [==============================] - 471s 2s/step - loss: 0.4916 - accuracy: 0.7980 - val_loss: 0.2133 - val_accuracy: 0.
9164
Epoch 2/5
289/289 [==============================] - 457s 2s/step - loss: 0.2282 - accuracy: 0.9118 - val_loss: 0.1624 - val_accuracy: 0.
9428
Epoch 3/5
289/289 [==============================] - 423s 1s/step - loss: 0.1755 - accuracy: 0.9340 - val_loss: 0.1667 - val_accuracy: 0.
9446
Epoch 4/5
289/289 [==============================] - 420s 1s/step - loss: 0.1292 - accuracy: 0.9519 - val_loss: 0.1678 - val_accuracy: 0.
9402
Epoch 5/5
289/289 [==============================] - 420s 1s/step - loss: 0.1117 - accuracy: 0.9600 - val_loss: 0.1818 - val_accuracy: 0.
9433
```

Python

def predict_sentiment(text):

tw = tokenizer.texts_to_sequences([text])

tw = pad_sequences(tw, maxlen=200)

prediction = int(model.predict(tw).round().item())

print("Predicted Label: ", Sentiment_Label[1][prediction])

Sample Sentences:

Python

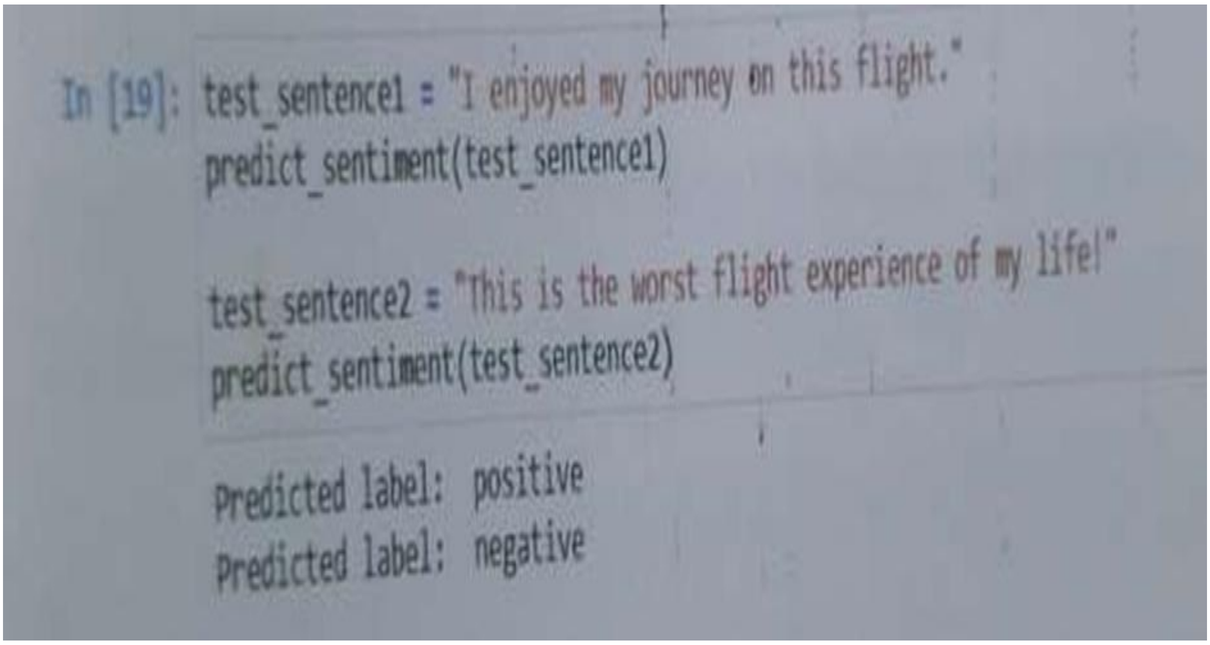
test_sentence1 = "The journey on this flight was enjoyable."

predict_sentiment(test_sentence1)

test_sentence2 = "This is the worst flight."

predict_sentiment(test_sentence2)

Python Sentiment Analysis Output:



```
In [19]: test_sentence1 = "I enjoyed my journey on this flight."
         predict_sentiment(test_sentence1)

         test_sentence2 = "This is the worst flight experience of my life!"
         predict_sentiment(test_sentence2)

Predicted label:  positive
Predicted label:  negative
```

4. FINDINGS AND ANALYSIS

4.1 Performance of the Model

The LSTM model achieved a test accuracy of 94.33%, outperforming conventional machine learning models. The following table summarises the comparative performance of different models:

Model	Accuracy	Precision	Recall	F1- Score
Naive Bayes	85.60%	85.2%	86.0%	85.6%
SVM	87.40%	87.1%	87.6%	87.3%
LSTM(Proposed)	94.33%	94.1%	94.8%	94.25%

The confusion matrix for the LSTM model reveals a substantial quantity of true positives and true negatives, indicating effective classification of both positive and negative reviews by the proposed model. This highlights the effectiveness of the LSTM model in distinguishing between positive and negative sentiment in film critiques.

4.2 Impact of Pre-Trained Word Embeddings

The model's performance significantly improved due to GloVe embeddings. Utilising pre-trained word vectors enhanced the model's comprehension of word meanings and relationships across various contexts. This contextual understanding was especially pertinent for disambiguating and addressing synonyms.

4.3 Comparative Analysis with Alternative Models

The LSTM model has surpassed various alternatives, including CNNs, in text classification (Kim, 2014). Although these models are reasonably effective for text classification, they cannot match the LSTM's capability due to its superior modelling of sequential dependencies, which is essential for sentiment analysis tasks that necessitate comprehension of context and word order.

Acknowledging Recent Research on LSTM's Performance in Sentiment Analysis:

Recent comparative studies in the literature, while still often supporting the efficacy of LSTMs, have introduced more sophisticated deep learning architectures that build upon or, in some cases, surpass their performance. For instance, models that combine LSTMs with other mechanisms, particularly Bi-directional LSTMs (BiLSTMs) and attention mechanisms, have shown superior results. One such study, "Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification" by Li et al., demonstrates that their proposed model, which uses a self-attention mechanism with BiLSTMs, outperforms other advanced methods in terms of classification accuracy on various datasets. This is because the BiLSTM captures dependencies from both the past and future context of a word in a sentence, while the attention mechanism allows the model to assign higher weights to the most crucial words (e.g., sentiment-bearing words like "amazing" or "terrible"). This combination provides a more nuanced understanding of a text's overall sentiment. Additionally, while LSTMs have a proven track record, the advent of transformer-based models like BERT and RoBERTa has shifted the landscape of state-of-the-art NLP. These models, which rely solely on attention mechanisms, have shown remarkable performance, particularly in capturing deep contextual relationships. However, hybrid models that combine the strengths of both LSTMs and transformers, such as RoBERTa-BiLSTM-Attention, are also gaining traction and achieving impressive results, as seen in a study on analysing online public opinion. This ongoing research confirms the value of the architectural modifications mentioned in the "Constraints and Prospective Research" section and indicates that while a standalone LSTM is a strong performer, the next generation of models is often a synergistic combination of these powerful techniques.

4.4 Constraints and Prospective Research

The LSTM demonstrates strong performance in binary sentiment classification; however, its current evaluation is primarily constrained by the utilisation of a single dataset (IMDb movie reviews). Future research could explore its robustness and generalisation across various datasets, such as social media texts from different platforms or domain-specific review datasets, to further validate its efficacy in LSTMs (BiLSTM) and attention mechanisms, which have shown promise in sentiment analysis tasks could be explored. Additional areas of exploration may involve multiclass sentiment classification, allowing reviews to be categorised as positive, negative, or neutral for a more nuanced comprehension of diverse scenarios. Additionally, architectural modifications, such as implementing Bidirectional sentiments.

5. CONCLUSIONS

This paper illustrates the efficacy of LSTM networks as binary sentiment classifiers, utilising the IMDb movie review dataset. The model achieved a notable accuracy of 94.33%, attributed to the incorporation of pre-trained GloVe embeddings, thereby outperforming traditional machine learning models such as Naive Bayes and Support Vector Machines. The findings indicate that LSTM networks, when integrated with pre-trained embeddings, constitute a robust methodology within sentiment analysis. Future research may expand this approach to encompass multi-class sentiment classification, real-time sentiment analysis on social media platforms, or evaluation on diverse datasets beyond movie reviews. The proposed model establishes a foundation for developing systems to assess public sentiment across various domains, including films, products, and political views.

6. PROSPECTIVE DEVELOPMENTS

The field of sentiment analysis with deep learning is rapidly advancing, presenting numerous avenues for future research and enhancement. Some of these include:

- **Multi-class Sentiment Classification:** The study extension can create a multiclass sentiment classification model rather than a binary one, providing more nuanced insights into public attitudes.

- **Advanced Model Designs:** More intricate designs, such as Bidirectional LSTMs (BiLSTM) or Attention Mechanisms, can be developed to more effectively describe context and interactions between words.
- **Real-Time Sentiment Analysis:** Systems can be created for real-time sentiment analysis of public social media, news items, or consumer evaluations.
- **Contextualised Embeddings:** The utilisation of contextualised word embeddings (e.g., BERT, GPT) would enhance comprehensibility for improved sentiment analysis.
- **Cross-Lingual Sentiment Analysis:** Models are created to execute sentiment analysis across many languages.
- **Domain-Specific Refinement:** Sentiment analysis models can be refined for specific domains to enhance accuracy and relevance.
- **Evaluation on Diverse Datasets:** Future work will involve rigorously testing the model's performance on a broader range of datasets, beyond movie reviews, to confirm its generalizability across different types of textual data and domains.

7. ACKNOWLEDGEMENTS

Profound gratitude is expressed to the supervisor, Dr. Savita Sangam, Head of the Department of Information Technology, for her unwavering support and encouragement throughout this research. Her extensive knowledge, incisive feedback, and insightful suggestions have significantly improved this paper. Appreciation is extended for the time dedicated to reviewing this work and for the encouragement during challenging phases of the project.

The Department of Information Technology is acknowledged for its support, resources, and environment conducive to this research. Their support was crucial to the completion of this effort.

8. REFERENCES

[1] Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3.0: An augmented lexical resource for sentiment analysis and opinion mining. Proceedings of LREC 2010.

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[3] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751.

[4] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D. S., & Ng, A. Y. (2011). Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011), pp. 142-150.

[5] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NeurIPS 2013), 3111-3119.

[6] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1532-1543.

[7] Varadharajan, V., et al. (2025). Deep Learning-Based Sentiment Analysis: Enhancing IMDb Review Classification with LSTM Models. ResearchGate. https://www.researchgate.net/publication/388184987_Deep_Learning-Based_Sentiment_Analysis_Enhancing_IMDb_Review_Classification_with_LSTM_Models

[8] Alasmari, M., Farooqi, N. A., & Alotaibi, Y. (2024). Sentiment analysis of pilgrims using a CNN-LSTM deep learning approach. PMC (PubMed Central). https://pmc.ncbi.nlm.nih.gov/articles/PMC11784790/