e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:12/December-2024 Impact Factor- 8.187 www.irjmets.com

SENTIMENT ANALYSIS USING LSTM

Dr. V. Vijayakumar*1, Deepak*2

*1Professor & Head, Department Of CSE, AVS Engineering College, Salem, Tamilnadu, India.

*2Student, Computer Science Engineering, AVS Engineering College, Salem, Tamilnadu, India.

DOI : https://www.doi.org/10.56726/IRJMETS65428

ABSTRACT

Sentiment Analysis, the process of identifying and categorizing opinions expressed in text to determine the sentiment conveyed, has gained immense significance in the era of data-driven decision-making. Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN), has proven to be a powerful tool for tackling sentiment analysis tasks due to its ability to capture long-term dependencies and context in sequential data. This project explores the application of LSTM networks for sentiment analysis on textual data. The study emphasizes preprocessing techniques like tokenization, stop-word removal, stemming, and vectorization using embeddings such as Word2Vec or GloVe, which transform raw text into numerical representations. These representations are then fed into an LSTM model designed to understand and predict sentiments such as positive or negative. The proposed LSTM-based approach addresses challenges like handling large vocabularies, dealing with contextually rich sentences, and understanding sentiment shifts within a sequence. By leveraging the memory cell structure of LSTM, the model effectively retains relevant information while discarding irrelevant details, resulting in a robust sentiment classification mechanism.

I. INTRODUCTION

1.1 OVERVIEW

This project focuses on performing sentiment analysis using Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN) known for its ability to process sequential data efficiently. Sentiment analysis, also referred to as opinion mining, involves analyzing text data to classify the sentiment as positive, negative, or neutral. With the exponential growth of text-based data such as product reviews, tweets, and comments, there is a pressing need for effective tools to extract sentiment insights. Traditional RNNs are unable to retain information over long sequences, which limits their performance. LSTM overcomes this limitation by incorporating memory cells and gates that control information flow. This project leverages Python for model development, utilizing TensorFlow/Keras for the LSTM model, and Streamlit for creating an interactive web-based user interface. Sentiment analysis is a branch of Natural Language Processing (NLP) that involves extracting and categorizing the emotions or opinions expressed in a text. This could involve determining if a statement is positive, negative, or neutral. Sentiment analysis is highly valuable for businesses, social media platforms, and even government agencies to understand public opinions, gauge customer satisfaction, and monitor brand perception. For instance, analyzing customer reviews, social media posts, or survey responses can help organizations make data-driven decisions. Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Networks (RNNs) designed to tackle the limitations of traditional RNNs. While RNNs are well-suited to sequence data, such as text, they suffer from issues like the vanishing gradient problem, where long-term dependencies in sequences are hard to learn. LSTM networks solve this problem by using a memory cell to retain information for longer periods and selectively forget irrelevant information, making them highly effective for tasks where context or sequence of events is important, like sentiment analysis.

LSTM models excel at capturing the relationships between words in a sentence, which is essential for understanding the sentiment. For example, in the sentence "I love this movie," the word "love" conveys a positive sentiment, while in "I don’t love this movie," the negation changes the sentiment. LSTM can remember such dependencies and accurately predict the sentiment.

1.2 PROJECT SCOPE

This project seeks to address challenges in traditional sentiment analysis by introducing an LSTM-based deep learning solution. The scope includes the followingaspects:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:12/December-2024 Impact Factor- 8.187 www.irjmets.com

Dataset

Selection: Publicly available datasets such as the IMDB movie reviews dataset are used for training and testing. Diversity: These datasets contain balanced and labeled data, ensuring robust model evaluation for multiple sentiment classes (positive, negative, neutral).

Implementation

Data Preprocessing: Techniques such as removing stopwords, tokenization, lemmatization, and padding sequences are used to clean and prepare text for analysis.

Model Architecture:

- ➢ Embedding layers convert textual data into dense vectors.
- ➢ LSTM layers are designed to process sequences and capture dependencies.
- ➢ Dense layers provide the final classification output.
- ➢ Tools and Frameworks: Python is utilized with TensorFlow/Keras for building the model and Streamlit for creating a user-friendly interface.

Deployment

The trained LSTM model is integrated into a Streamlit application, enabling real-time sentiment predictions. Users can input text into the application to receive instant feedback on sentiment polarity.

1.3 PURPOSES

1) Automation of Sentiment Analysis: Reduce manual effort in analyzing text data.Provide a fast and reliable method for sentiment detection.

2) User Accessibility:Develop an intuitive interface that allows non-technical users to easily utilize sentiment analysis tools. Ensure real-time predictions for enhanced user experience.

3) Demonstration of LSTM Capabilities: Highlight the advantages of LSTMs in processing sequence-based data. Showcase how LSTMs effectively handle long-term dependencies and context in text.

4) Practical Insights: Help businesses and researchers extract actionable insights from user-generated content. Enable better decision-making by analyzing sentiment trends.

1.4 STEPS INVOLVED IN LSTM-BASED SENTIMENT ANALYSIS

DATA COLLECTION AND PREPROCESSING

Before training an LSTM model, large datasets of labeled text are required. These datasets often consist of text data, such as customer reviews or tweets, labeled with sentiments (positive, negative, neutral). The text data needs to be preprocessed to remove noise, such as special characters or stop words, and standardized. Text preprocessing might involve lowercasing, tokenization (splitting text into individual words), and padding sequences to ensure they all have the same length for feeding into the LSTM.

TEXT TOKENIZATION AND SEQUENCE PADDING

Tokenization is the process of converting text into numerical representations, usually done by creating a vocabulary from the dataset. Words or sub-words in the text are mapped to integers, which the LSTM can process. However, since LSTMs work with sequences of fixed length, padding is applied to shorter sequences, ensuring that all input sequences to the model are the same length.

BUILDING THE LSTM MODEL

The LSTM model architecture for sentiment analysis typically consists of several layers. The model begins with an embedding layer, which converts word indices into dense vectors representing words in a lower-dimensional space, capturing their semantic meanings. The LSTM layer then processes these word embeddings, learning contextual dependencies within the sequence. A dense layer follows, which outputs the sentiment prediction (positive, negative, or neutral).

TRAINING THE MODEL

Once the model architecture is defined, it is trained on the labeled text data. During training, the model adjusts its internal parameters (weights) to minimize the error in its predictions. The loss function typically used in sentiment analysis tasks is **binary cross-entropy** (for binary classification like positive or negative sentiment)

EVALUATION AND PERFORMANCE METRICS

After training the model, its performance needs to be evaluated. Common evaluation metrics for sentiment analysis models include **accuracy**, **precision**, **recall**, and **F1 score**. These metrics help assess how well the model can predict sentiment correctly, especially in cases where data is imbalanced (for instance, more positive reviews than negative ones).

DEPLOYING THE MODEL

Once the model is trained and evaluated, it can be deployed to predict sentiment in real-time on new, unseen data, such as customer reviews or tweets. The LSTM model will then classify the sentiment of new text into one of the predefined categories (positive, negative, or neutral).

1.5 BACKGROUND OF THE STUDY

The advent of social media platforms, online forums, and e-commerce sites has led to an unprecedented explosion of user-generated content. This content contains valuable insights into public opinion, customer satisfaction, and social trends. Analyzing this data manually is impractical due to its volume and complexity, paving the way for sentiment analysis systems.

TRADITIONAL APPROACHES:

Traditional sentiment analysis relied heavily on statistical and rule-based techniques:

➢ Bag of Words (BoW): This approach represents text as a collection of words, focusing solely on word frequency without considering context or word order. While easy to implement, BoW fails to capture semantic relationships betweenwords.

➢ TF-IDF (Term Frequency-Inverse Document Frequency): While TF-IDF improves text representation by assigning weights to words based on their importance in a document, it still struggles to maintain the contextual flow andtemporal dependencies of words within a sentence

ADVANCES IN NLP

Recent advancements in NLP have transformed sentiment analysis by introducing models that leverage the sequence and context of text. These include:

➢ Word Embeddings: Techniques like Word2Vec and GloVe map words into continuous vector spaces, preserving semantic relationships. For instance, embeddings can capture analogies like "king - man + woman = queen."

➢ Deep Learning Models: Models such as LSTMs are designed to handle sequential data effectively. They account for both the order of words and the context, addressinglimitations in traditional methods.

➢ LSTM-based sentiment analysis represents a significant step forward, as it processes complex sentence structures and long-term dependencies, making it ideal for modernNLP tasks.

II. SYSTEM ANALYSIS

2.1 SYSTEM ANALYSIS

The system comprises several components:

Data Preprocessing:

Text cleaning: Removing stopwords, punctuation, and special characters.
Tokenization: Splitting sentences into individual words.
Padding: Ensuring uniform input length by adding zero-padding.

Model Training:

Building the LSTM model using TensorFlow/Keras.
Optimizing hyperparameters such as learning rate and dropout.
Using validation data to fine-tune the model.

Deployment:

Integrating the trained model with a user-friendly Streamlit interface.

2.2 EXISTING SYSTEM

The existing systems for sentiment analysis have significantly evolved over the years. Traditional methods were rule-based and depended heavily on predefined lexicons. Modern systems leverage advanced machine learning and deep learning techniques to provide more accurate and context-aware sentiment predictions.

2.2.1 DRAWBACKS OF EXISTING SYSTEM

- Rule-based systems cannot interpret complex language structures like sarcasm ornegation.
- Machine learning models relying solely on BoW or TF-IDF ignore word order, leading to loss of contextual meaning.

2.2.2 CHALLENGES

- Difficulty in handling ambiguous sentences.
- Limited scalability for real-world applications.

2.3 PROPOSED SYSTEM

The proposed system for sentiment analysis leverages modern deep learning techniques, specifically Long Short-Term Memory (LSTM) networks, combined with a user-friendly interface and advanced visualization capabilities. This system addresses the limitations of existing models by offering improved accuracy, contextual understanding, and scalability for various applications like customer reviews, social media monitoring, and emotion detection.

ADVANTAGES:

- Utilizes LSTM networks for superior performance in sequence learning.
- Features a scalable web-based interface for real-time sentiment predictions.
- Offers confidence scores and visual insights into model predictions.

ADDITIONAL FEATURES:

- Support for multilingual datasets to expand use cases.
- Potential for integrating additional NLP tasks such as emotion detection or topicmodeling.

2.3 FEASIBILITY STUDY

2.3.1 ECONOMICAL FEASIBILITY

Use of free datasets and open-source tools minimizes costs.

2.3.2 TECHNICAL FEASIBILITY

Modern hardware and software tools ensure smooth implementation. Availability of pre- trained embeddings reduces computational overhead.

2.3.3 OPERATIONAL FEASIBILITY

Streamlit ensures an accessible interface for users with minimal technicalexpertise.

2.3.4 ENVIRONMENTAL FEASIBILITY

This project environment is correct as a admin has developed this system and no expenditure is involved under any head and this process is part of admin document management, this project environment is accessible.

2.4 SYSTEM REQUIREMENTS

2.4.1 HARDWARE REQUIREMENTS

The Hardware of the computer consists of physical component such as Input Devices, Storage Devices, Processing & Control units and Output Devices. Computer includes external storage unit to store data in programs.

The Hardware Configuration involved in this project

- **Processor**: Intel i5 or higher.
- **RAM**: 8 GB or more.
- **Storage**: Minimum 10 GB free space.

2.4.2 SOFTWARE REQUIREMENTS

Software is a group of programs that computers need to do a particular task. It is an essential requirement of Computer System. The Software used to develop the project is

- Programming Language: Python.

- Libraries: TensorFlow/Keras, NumPy, Pandas, Streamlit, NLTK/spacy.
- IDE: Visual Studio Code.
- Web Browser: Google Chrome or equivalent

III. SYSTEM DESIGN AND DEVELOPMENT

DIAGRAM OF LSTM CELL

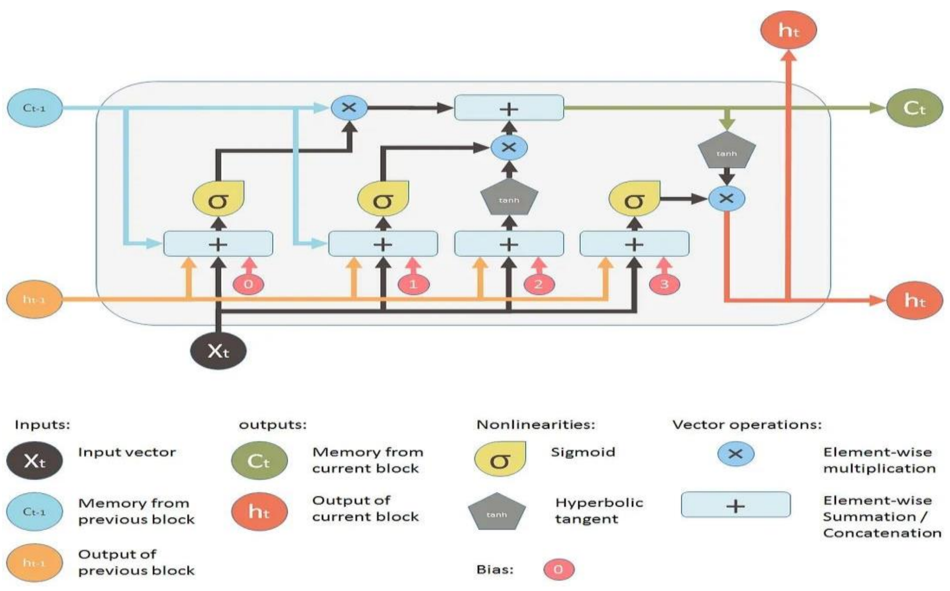

ht
Ct-1
Ct
tanh
σ
σ
σ
tanh
σ
0
1
2
3
ht-1
ht
Xt
Inputs:
Xt Input vector
Ct-1 Memory from previous block
ht-1 Output of previous block
outputs:
Ct Memory from current block
ht Output of current block
Nonlinearities:
σ Sigmoid
tanh Hyperbolic tangent
Bias: 0
Vector operations:
× Element-wise multiplication
+ Element-wise Summation / Concatenation

Fig 3.1

3.2 SYSTEM ARCHICTURE

III. ARCHITECTURE OF PROPOSED NETWORK USED

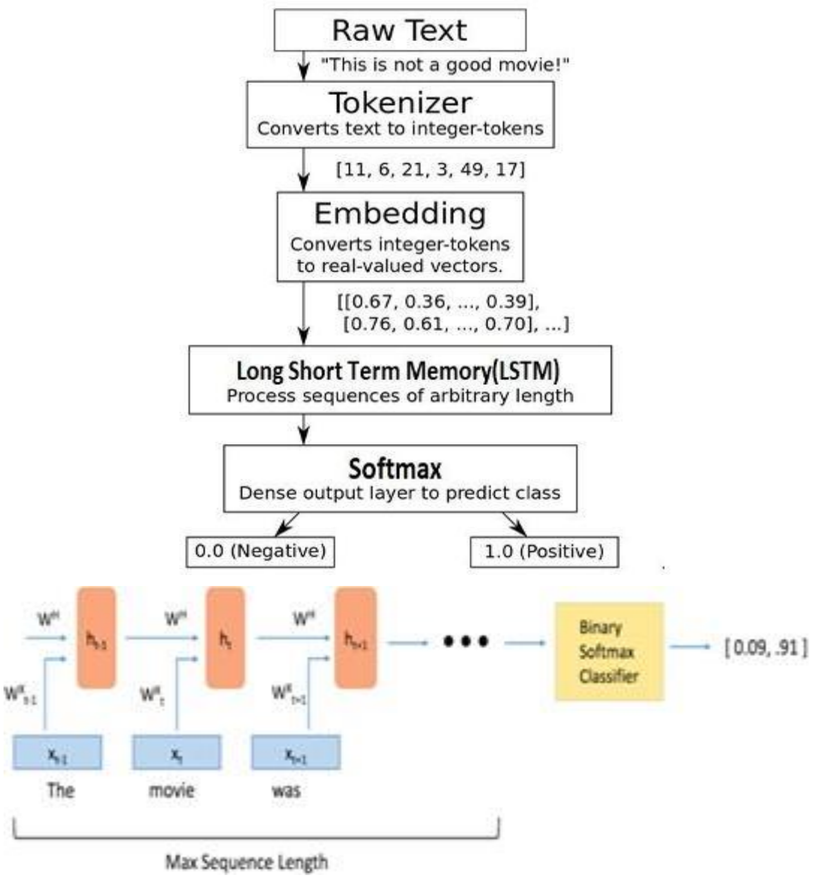

Raw Text
"This is not a good movie!"
Tokenizer
Converts text to integer-tokens
[11, 6, 21, 3, 49, 17]
Embedding
Converts integer-tokens to real-valued vectors.
[[0.67, 0.36, ..., 0.39],
[0.76, 0.61, ..., 0.70], ...]
Long Short Term Memory(LSTM)
Process sequences of arbitrary length
Softmax
Dense output layer to predict class
0.0 (Negative)
1.0 (Positive)
W
h_{t-1}
h_t
h_{t+1}
Binary Softmax Classifier
[0.09, .91]
x_{t-1}
x_t
x_{t+1}
The
movie
was
Max Sequence Length

Fig 3.2

IV. TESTING AND IMPLEMENTATION

4.1 TESTING

Implementation is the stage of the project when the theoretical design is turned into a working system. This is the final and important phase in the system life cycle It is actually the process of converting the new system into a operational one.

TESTING METHODS:

To ensure the reliability and robustness of the system, rigorous testing is performed at variousstages:

1. Unit Testing:

Focuses on testing individual components such as the preprocessing pipeline, LSTM model, and output formatting to ensure they work as intended.

2. Integration Testing:

Verifies the seamless interaction between different modules, such as the preprocessing pipeline, LSTM model, and Streamlit interface.

3. End-to-End Testing:

Simulates real-world scenarios where users input text, and the system processes it through all stages to ensure the final output is accurate and user-friendly.

Evaluation Metrics:

To assess the model's performance, the following metrics are used:

Accuracy:

Measures the proportion of correct predictions out of all predictions made bythe model.

Precision:

Evaluates the accuracy of positive sentiment predictions (i.e., the proportion of true positives to the total predicted positives).

Recall:

Measures how effectively the model identifies all instances of a positivesentiment.

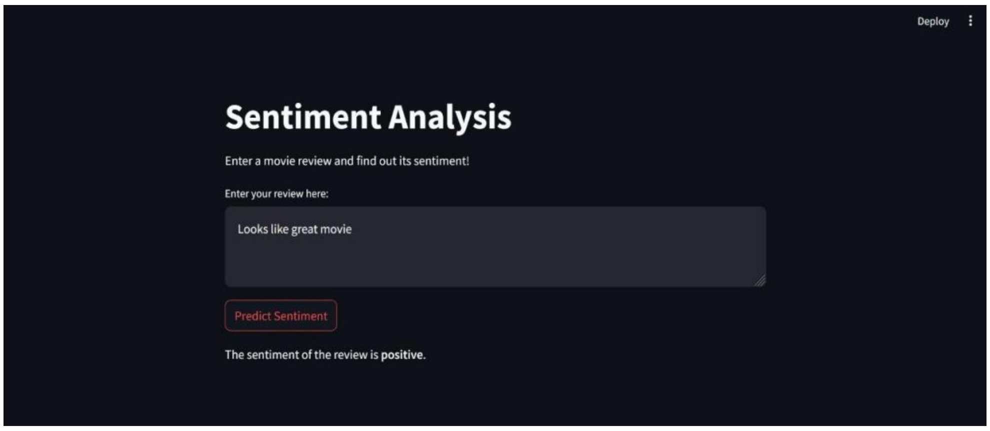
F1-Score:

The harmonic mean of precision and recall, providing a balanced metric for imbalanced datasets.

4.2 SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. This is the final and important phase in the system life cycle It is actually the process of converting the new system into a operational one.

V. RESULT AND OUTPUT



Sentiment Analysis

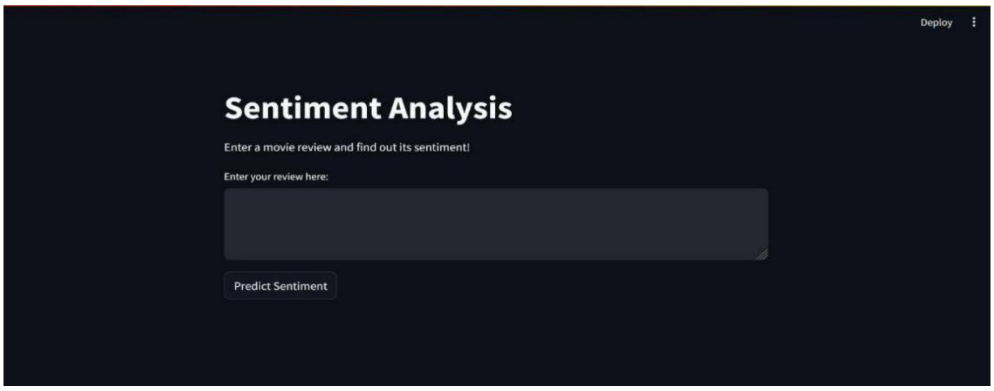
Enter a movie review and find out its sentiment!

Enter your review here:

Looks like great movie

The sentiment of the review is **positive**.

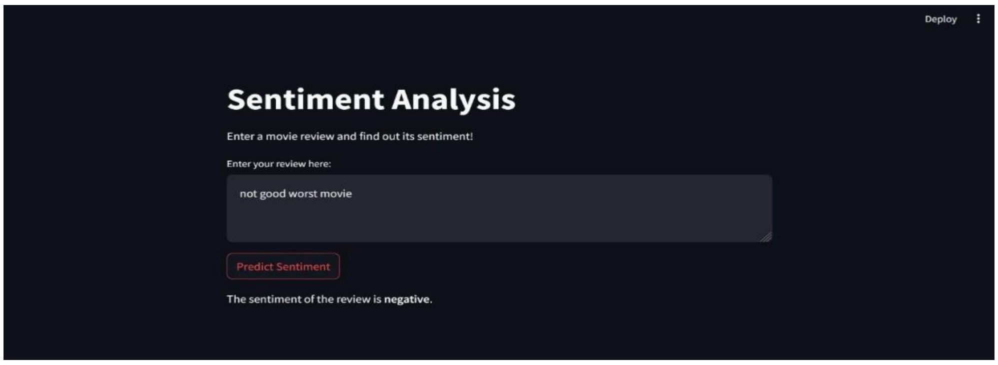
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:12/December-2024 Impact Factor- 8.187 www.irjmets.com



Sentiment Analysis

Enter a movie review and find out its sentiment!

Enter your review here:



Sentiment Analysis

Enter a movie review and find out its sentiment!

Enter your review here: not good worst movie

The sentiment of the review is **negative**.

VI. CONCLUSION

In this paper, In the field of Natural Language Processing (NLP), sentiment analysis plays a pivotal role in understanding user opinions, emotions, and feedback across various domains. This project, "Sentiment Analysis Using LSTM," demonstrates the potential of deep learning techniques, specifically Long Short-Term Memory (LSTM) networks, in addressing the challenges of sentiment classification. By effectively modeling the sequential and contextual nature of text data, this project has provided a robust framework for analyzing sentiments with high accuracy and user-friendliness. The integration of LSTM into sentiment analysis addresses several limitations of traditional methods. Conventional machine learning models like Support Vector Machines (SVM) or Naive Bayes often fail to consider the sequential dependencies of words in a sentence. LSTMs, with their ability to retain long-term memory and selectively forget irrelevant information, provide a significant advantage in capturing both local and global context. This capability has been demonstrated through the implementation of an LSTM model trained on text data, achieving reliable predictions for sentiment classification tasks. In conclusion, this project serves as a comprehensive demonstration of how LSTM networks can be effectively utilized for sentiment analysis. It underscores the importance of combining state-of-the-art deep learning techniques with practical deployment strategies to create impactful solutions. By addressing current challenges and laying the groundwork for future advancements, the project contributes to the growing body of work in NLP and paves the way for more sophisticated and inclusivesentiment analysis systems.

VII. REFERENCE

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. Comprehensive guide on deep learning architectures, including LSTMs andtheir applications in sequence modeling.

[2] Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." Neural Computation, 9(8), 1735-1780. The original paper introducing LSTM networks, providing insights into their architecture and capabilities.

[3] Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing. Pearson. Explains the concepts of natural language processing, sentiment analysis, andword embeddings in detail.

[4] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). "Efficient Estimation of Word Representations in Vector Space." arXiv preprint arXiv:1301.3781. Discusses Word2Vec embeddings, which are foundational for representingwords in sentiment analysis