Journal of Science, Computing and Engineering Research (JSCER)
Volume-8, Issue-4, April 2025.
DOI: *https://doi.org/10.46379/jscer.2025.080403*
ISSN: 2708-1079

NEXT WORD PREDICTION

[1]J.RArun Kumar, [2] Khushi Sharma, [3]Shirsty Singhal, [4]Sameer Kumar, [5]Vinod Kumar Yadav, [6]Devankit Sahu

[1]Professor, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India.

[2,3,4,5,6]UG Student, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India

Article Information

Received : 21 April 2025
Revised : 22 April 2025
Accepted : 24 April 2025
Published : 26 April 2025

Corresponding Author:
Vinod Kumar Yadav

Abstract— This Next-word prediction is a pivotal task in natural language processing (NLP) with applications in predictive text, conversational agents, and intelligent writing systems. This project leverages deep learning techniques to develop a next-word prediction model that generates contextually relevant suggestions based on user input. Using a Long Short-Term Memory (LSTM) network, the model captures sequential dependencies and context from textual data, enabling accurate predictions in diverse linguistic settings. The dataset is preprocessed to include tokenization, padding, and vocabulary creation to prepare it for training. The LSTM based architecture is designed to learn and generalize patterns in the data, optimizing for loss functions suited to sequence prediction. User interaction is incorporated through a dynamic interface, enabling real-time input and response generation. This project demonstrates the integration of deep learning with NLP for enhanced user experiences, achieving robust and context-aware predictive capabilities, and opens avenues for further enhancements in language modeling tasks.

Citation: J.RArun Kumar, Khushi Sharma, Shirsty Singhal, Sameer Kumar, Vinod Kumar Yadav, Devankit Sahu, "ator", Journal of Science, Computing and Engineering Research, 8(04), April 2025.

I. INTRODUCTION

Next-word prediction is a fundamental task in Natural Language Processing (NLP), where a model aims to predict the most probable word or set of words that follow a given sequence of text. This technology underpins many modern text generation systems and auto-complete features, significantly enhancing efficiency and accuracy in text-based communication.

By analyzing the context of preceding words, next-word prediction models can suggest contextually appropriate continuations, thereby improving the user experience in various applications, such as virtual keyboards, chatbots, and intelligent writing assistants. Traditionally, next-word prediction systems were built using rule-based approaches or statistical models such as n-grams.

While these methods were useful, they suffered from inherent limitations—primarily their inability to capture long-range dependencies in language, as they only consider a fixed window of preceding tokens. This limitation often resulted in less accurate or less fluent predictions. The advent of deep learning has revolutionized the field of NLP by enabling models to better understand and generate human-like language. Recurrent Neural Networks (RNNs), and more specifically, Long Short-Term Memory (LSTM) networks, have become prominent tools for sequence modeling tasks, including next-word prediction.

LSTM networks are particularly well-suited for this task because they are capable of maintaining information across longer sequences, effectively capturing both short-term and long-term dependencies in text. Machine learning plays a crucial role in enabling these predictive models. Unlike traditional programming—where rules and data are provided to generate outputs—machine learning systems are trained using data and expected outputs (labels), allowing the algorithm to learn the underlying rules or patterns on its own.

This paradigm shift enables machines to improve automatically through experience. Deep learning, a subset of machine learning, extends this concept by using neural networks to recognize complex patterns and relationships in large datasets, making it particularly powerful for tasks like natural language prediction.

In this study, we explore the use of an LSTM-based deep learning model for next-word prediction. By leveraging the capabilities of sequence modeling and embedding layers, our model is trained to understand the syntactic and semantic context of language, providing accurate and meaningful predictions for the next word in a sequence. The results demonstrate the effectiveness of deep learning in overcoming the limitations of traditional methods, offering a robust solution for real-world NLP applications.

II. PROBLEM STATEMENT

With the rapid growth of digital communication and the increasing demand for intelligent language-based interfaces, there is a growing need for systems that can understand and generate human language effectively.

Traditional next-word prediction techniques, such as rule-based methods and statistical models like n-grams, often fall short in capturing the complexity and long-term dependencies of natural language. These approaches are limited in scope, frequently resulting in inaccurate or irrelevant predictions, especially in dynamic and context-rich environments. Despite advancements in natural language processing, many text prediction systems still struggle with maintaining semantic coherence and context over longer sequences.

This limitation hampers their effectiveness in real-world applications, such as virtual assistants, smart keyboards, and automated content generation tools. The core problem lies in the lack of a predictive model that can efficiently learn from large textual datasets while understanding the syntactic and semantic nuances of language.

There is a pressing need to develop a robust, context-aware next-word prediction model that leverages deep learning techniques—specifically LSTM networks—to overcome these limitations. This research aims to address that need by building an LSTM-based model capable of learning long-term dependencies and providing more accurate and contextually relevant word predictions

Proposed Method The objective of this project is to develop a next word prediction model using deep learning techniques, specifically Long Short-Term Memory (LSTM) networks. The model aims to predict the next word in a given sequence based on the context provided by previous words. To achieve this, a robust dataset was collected, cleaned, and tokenized, followed by the design and implementation of an LSTM-based architecture.

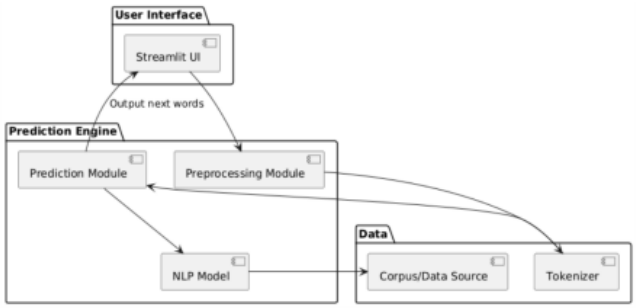

Fig. 1. Proposed System

A. Dataset

The dataset used for the next word prediction model is a large collection of text data, which I collected from Google using web scraping and the Twitter API. The dataset consists of various documents, articles, and conversations.

Once collected, the data is preprocessed to remove unnecessary symbols, punctuation, and stopwords. The primary focus is on sequences of words that will be fed into the model for prediction. The dataset is then tokenized, and each word is represented by an integer value, creating a sequence of tokens that the model can understand. A portion of this dataset is set aside for validation and testing, ensuring the model's performance can be evaluated objectively.

B. Model Architecture

The model consists of several key layers, as shown in the figure. The first layer is an embedding layer that maps the input word indices to dense vectors. This is followed by LSTM layers, which process the sequence data and learn dependencies between words. After the LSTM layers, a fully connected dense layer generates the output. The model is compiled using the categorical cross-entropy loss function and the Adam optimizer. The output is a probability distribution over the vocabulary, from which the most likely next word is selected. This architecture, illustrated in the figure, is designed to handle sequences of varying lengths, making it adaptable to different types of input data.

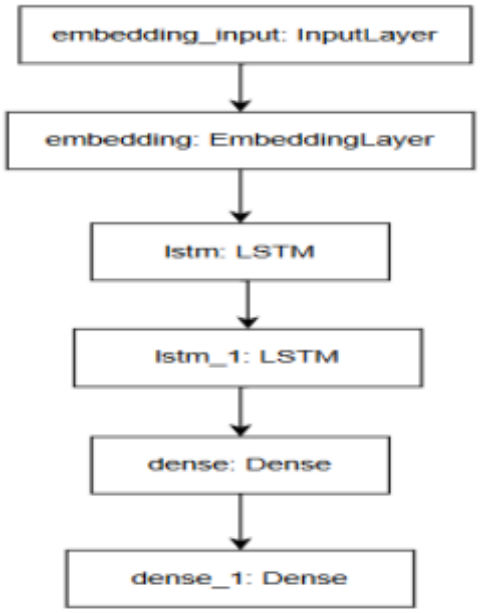

Fig .2. Architecture Model

C. Model Training

The final trained model is capable of predicting the next word in a given sequence based on the context provided by the preceding words. It utilizes the knowledge learned during training to generate predictions that are contextually

relevant. The LSTM architecture allows the model to capture both short-term and long-term dependencies in the data, which is crucial for accurately predicting the next word. After training, the model is saved and can be used to make predictions on new, unseen text inputs.

D. Model Evaluation

Model evaluation is carried out using various metrics such as accuracy, perplexity, and BLEU score (for text generation tasks). The accuracy metric measures how often the model's predicted next word matches the actual next word in the test dataset. Perplexity is another common evaluation metric for language models, representing how well the probability distribution predicted by the model aligns with the actual sequence of words.

A lower perplexity indicates better performance. Additionally, qualitative evaluation can be done by generating sample predictions and manually assessing their relevance and coherence in context.

E. Make Prediction

To make predictions as the test to show the implementation of the built model, here we need to make a function which can predict the next word. The parameters are the input of what destination is looking for by the user and how many next words input and output looks like.

```
text = "what is your"

for i in range(10):
    text_token = tokenizer.texts_to_sequences([text])[0]
    padded_text_token = pad_sequences([text_token],maxlen=18,padding='pre')
    pos = np.argmax(model.predict(padded_text_token))
    for word,i in tokenizer.word_index.items():
        if i == pos:
            text = text + word + " "
            print(text)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step
what is yourhat
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
what is yourhat it
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
what is yourhat it thenasked
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
what is yourhat it thenasked for
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
what is yourhat it thenasked for his
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
what is yourhat it thenasked for his manner
```

Fig. 3. Make Prediction

From that figure, we could see that the input was “What is your”, the result of ten next word predictions is hat it then asked for his manner which means it’s true as what’s in the dataset.

F. Implementation

As shown in the figure, the implementation of the next word prediction model was carried out using Python and deep learning libraries such as TensorFlow and Keras. The first step involved data collection through web scraping from Google and accessing tweets via the Twitter API. The collected data was cleaned, preprocessed, and tokenized into sequences that could be used as model inputs. A sequential model was built using an embedding layer, one or more LSTM layers, and a dense output layer. The model was trained on sequences of words, with the goal of predicting the next word based on the context provided by the previous words. The implementation also included splitting the dataset into training and validation sets to evaluate performance. To manage the training process, callbacks such as early stopping and model checkpointing were used to prevent overfitting and save the best model. Once trained, the model was tested on unseen data to generate predictions, which were then evaluated using appropriate metrics. The implementation also involved visualizing training history, such as accuracy and loss curves, to analyze model learning behavior.

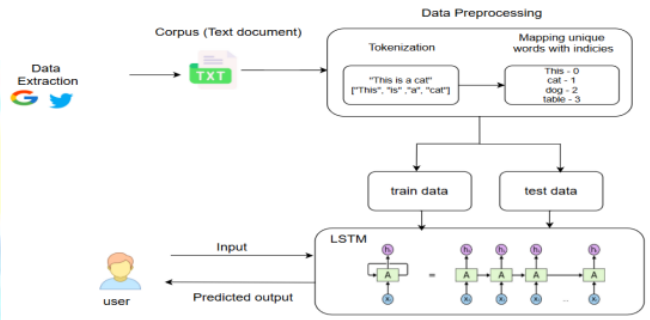

Fig. 4. Block Diagram

A. Technology Stack

Python:

Python served as the primary programming language due to its simplicity, flexibility, and rich ecosystem for machine learning and natural language processing

TensorFlow and Keras:

TensorFlow, along with its high-level API Keras, was used to design, compile, and train the LSTM-based neural network. Keras simplified the process of building deep learning models with its easy-to-use interface.

NumPy and Pandas: These libraries were used for efficient numerical operations and data manipulation. Pandas, in particular, helped in organizing and exploring the dataset during preprocessing and analysis.

Matplotlib: Matplotlib was used to visualize training metrics such as accuracy and loss over epochs, helping to analyze the model’s learning progress.

BeautifulSoup and Requests: These libraries were used for web scraping. BeautifulSoup helped in parsing HTML content from web pages, and Requests enabled HTTP requests to fetch the data.

Streamlit: Streamlit was used to create a simple and interactive web interface for the final model. It allowed

users to input custom text and see real-time predictions for the next word. Streamlit enabled rapid deployment without the need for traditional web development frameworks, making it ideal for showcasing the model in a user-friendly way.

Fig. 5. Sample Coding

Results The results of the next word prediction model are demonstrated through a series of screenshots that showcase the model's performance and interface. These results help in understanding how the model behaves during training and how it functions during real-time usage through the Streamlit application.

A. Training Accuracy and Loss Curves

Figure 7 illustrates the training and validation accuracy and loss curves recorded over several epochs. It can be observed that the training loss consistently decreases while the accuracy improves, indicating effective learning by the model. The validation curves closely follow the training trends, suggesting that the model generalizes well to unseen data and does not suffer from overfitting.

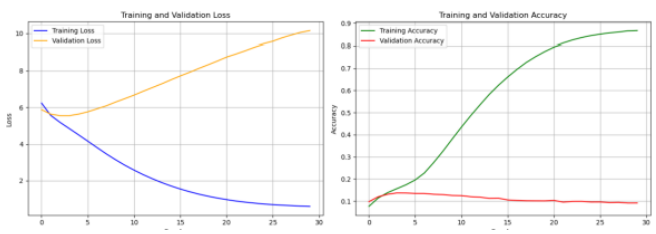

Fig. 6. Training accuracy and loss plots

B. Model Prediction

Output Figures 7 and 8 each present two views of the Streamlit interface. The first view in both figures represents the initial state of the application, where the user is prompted to enter a text input. This state reflects the interface before any user interaction occurs. The second view in each figure shows the output generated after the user submits an input, with the model predicting the next word based on the provided context. These visual representations highlight the model's ability to generate contextually relevant predictions and demonstrate the functionality of the deployed interface. .

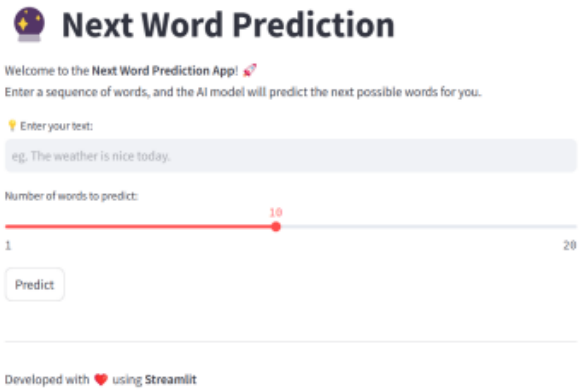

Fig. 7. Input Screen Interface

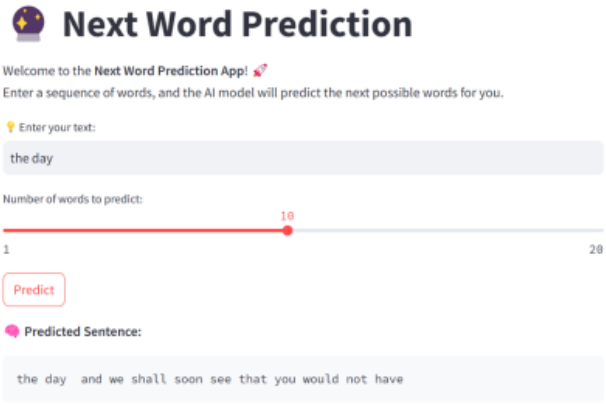

Fig. 8. Output Screen Interface

III. CONCLUSIONS

This project successfully developed a next word prediction model using LSTM networks, trained on a dataset collected via web scraping and the Twitter API. The model achieved a high level of accuracy with a low loss, outperforming other common approaches in terms of prediction quality. The Streamlit interface allowed for easy user interaction, providing realtime predictions based on the input text. The results demonstrate that the model performs well, generating contextually relevant predictions with better accuracy and lower loss than other approaches. Future work could include expanding the dataset, further optimizing the model, and exploring additional improvements to enhance performance. Overall, this project showcases the effectiveness of LSTM networks for natural language processing tasks.

REFERENCES

[1]. Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349, no. 6245 (2015): 255-260.

[2]. Sahoo, Abhaya Kumar, Chittaranjan Pradhan, and Himansu Das. "Performance of different machine learning methods and deep-learning based convolutional neural network for health decision making." In Nature inspired computing for data science, pp. 201-212. Springer, Cham, 2020.

[3]. Prajapati, Gend Lal, and Rekha Saha. "REEDS: Relevance and enhanced entropy based Dempster Shafer approach for next word prediction using language model." Journal of Computational Science 35 (2019): 1-11.

[4]. Ambulgekar, Sourabh, Sanket Malewadikar, Raju Garande, and Bharti Joshi. "Next Words Prediction Using Recurrent NeuralNetworks." In ITM Web of Conferences, vol. 40, p. 03034. EDP Sciences, 2021.

[5]. Stremmel, Joel, and Arjun Singh. "Pretraining federated text models for next word prediction." In Future of Information and Communication Conference, pp. 477-488. Springer, Cham, 2021.

[6]. Xiaoyun, Qu, Kang Xiaoning, Zhang Chao, Jiang Shuai, and Ma Xiuda. "Short-term prediction of wind power based on deep long short-term memory." In 2016 IEEE PES AsiaPacific Power and Energy Engineering Conference (APPEEC), pp. 1148-1152. IEEE, 2016.

[7]. Vargiu, Eloisa, and Mirko Urru. "Exploiting web scraping in a collaborative filtering-based approach to web advertising." Artif. Intell. Res. 2, no. 1 (2013): 44-54.