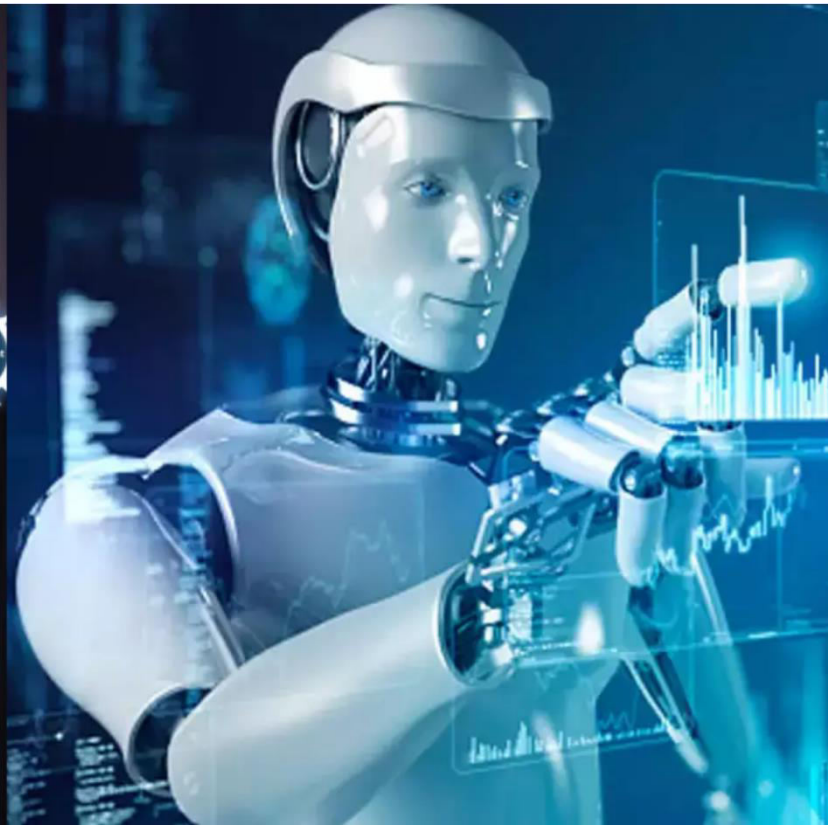
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404202|

Next Word Prediction using LSTM Model

M.Narayanan[1], Teki Asha Jyothi[2], Thouti Reddy Arthi Reddy[3], Bollu Ashritha[4], Uppu Adithya Sai[5]

Professor, Department of CSE, School of Engineering, Malla Reddy University, Hyderabad, India[1]

UG Scholar, Department of CSE, School of Engineering, Malla Reddy University, Hyderabad, India[2345]

ABSTRACT - Language modeling plays an important role in a variety of natural language processing (NLP) applications, including text generation, auto-action systems, and chatbots. Traditional N-GRAM models often suffer from data space and lack context-related understanding, limiting prediction accuracy. This deals with these challenges by using the following word prediction model based on deep learning.

The main goal of this study is to develop efficient predictive text models that can create contextually accurate word proposals. The dataset used to train the model is from the novel A-Couple female student, ensuring a structured, domain-specific language context. The approach includes preprocessing text data, tokenized words, and word embeddings to absorb semantic relationships. Sequentially, LSTM-based neuronal networks are trained to learn word dependencies and predict the next word that is most likely in a particular text sequence.Experimental results show that the proposed model achieves a high level of accuracy when predicting context-related words and traditional statistical methods. LSTM-based approaches effectively capture long-distance dependencies of text and improve predictive quality.

The integration of pre-educated wordbed endings will further improve the model model and allow for more meaningful suggestions.This study emphasizes the applicability of deep learning in the case of text completion tasks and their potential. Future work will include data records expansion, fine-tuning hyperparameters, and integration of trans-based architectures to improve performance.

KEYWORDS – Long Short Term Memory(LSTM), Recurrent Neural Networks (RNN), Natural Language Processing (NLP).

I. INTRODUCTION

Background and Significance of the Study

With the rapid advancement in Natural Language Processing (NLP) and artificial intelligence, predictive text systems have become an essential component of various digital communication tools. From smartphone keyboards to search engines and virtual assistants, next-word prediction enhances user experience by offering intelligent suggestions, reducing typing effort, and improving efficiency. Traditional n-gram models and statistical methods were widely used for text prediction; however, they lacked the ability to capture long-range dependencies within text sequences. The advent of deep learning, particularly Long Short-Term Memory (LSTM) networks, has significantly improved next-word prediction by leveraging sequential dependencies and context-aware modeling.

Research Problem and Motivation

Despite the success of existing NLP models, many next-word prediction systems still struggle with contextual accuracy, generalization, and handling ambiguous text inputs. Conventional approaches often fail in capturing complex linguistic structures and long-term dependencies, leading to erroneous or contextually irrelevant suggestions. This research aims to address these limitations by implementing an LSTM-based model that effectively learns and predicts the next word in a given sentence. By leveraging word embeddings and deep learning techniques, this study seeks to improve the accuracy and reliability of text prediction systems, which can be beneficial for applications such as text editors, chatbots, and language learning tools.

Objectives and Hypothesis

The primary objectives of this study are:
To develop an LSTM-based next-word prediction model that effectively learns language structures.
To improve text prediction accuracy by utilizing word embeddings and sequential deep learning.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404202|

To evaluate the performance of the proposed model using metrics such as perplexity and accuracy.
To compare the results with traditional statistical models and alternative neural architectures.

II. LITERATURE REVIEW

In [1] studies have explored that next-word prediction using various approaches, including n-gram models, Hidden Markov Models (HMMs), and deep learning techniques like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. While n-gram and statistical models struggled with long-range dependencies, LSTMs improved sequential learning by retaining context over longer sequences. Recent advancements, such as transformer-based models like GPT and BERT, have further enhanced prediction accuracy, but they come with high computational costs. Despite these developments, research gaps remain in optimizing LSTM models for real-time applications, improving dataset diversity, enhancing model interpretability, and exploring hybrid architectures that combine LSTMs with attention mechanisms for more efficient next-word prediction

In publication [2] the computerization of Dzongkha, focusing on challenges in digital processing, including typing efficiency and text prediction. Existing research highlights the difficulty of Dzongkha typing due to the multiple keystrokes required per word, making text entry tedious and inefficient. Previous works on language modeling have employed N-gram methods and embedding techniques to improve word prediction accuracy. RNN-based models, particularly Bi-LSTM architectures, have shown promising results, achieving a 73.89% accuracy in next-word prediction for Dzongkha. Despite these advancements, gaps remain in optimizing keystroke reduction, enhancing real-time text prediction, and integrating more sophisticated deep learning models. Further research is needed to refine these models and improve Dzongkha's usability in modern digital applications.

According to the findings in [3], next-word prediction as a key aspect of language modeling, utilizing deep learning techniques such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. Traditional approaches like N-gram models have been widely used but struggle with long-range dependencies. Modern NLP models leverage deep learning to enhance accuracy and efficiency, enabling real-time word predictions for applications like search engines, assistive typing, and entertainment computing. While existing research has demonstrated the effectiveness of Bi-LSTM and transformer-based models, challenges remain in optimizing prediction speed, handling diverse linguistic structures, and reducing computational complexity. Addressing these gaps can further improve the usability of next-word prediction systems, particularly for users with disabilities or those requiring faster text input.

The authors of [4] discuss particularly in non-English languages, where linguistic complexity and script variations pose challenges. Traditional models struggle with languages like Assamese, which have multiple synonyms for common pronouns and require phonetic transcription for efficient processing. Existing research has attempted to address this issue by leveraging phonetic transcription aligned with the International Phonetic Association (IPA) chart to enhance model accuracy. Despite achieving an accuracy of 88.20% for Assamese text and 72.10% for phonetically transcribed words, challenges remain in handling Unicode inconsistencies, improving real-time prediction accuracy, and optimizing models for multilingual contexts.

The review conducted in [5] have explored that language modeling as a fundamental task in NLP, with applications in text prediction, document retrieval, and structured search. Traditional language models have focused on next-word prediction using probabilistic methods, while recent advancements incorporate deep learning techniques like RNNs and LSTMs for improved accuracy. In structured document retrieval, tree-based generative language models have been proposed, where hierarchical document structures, including titles, sections, and paragraphs, are represented as nodes with corresponding language models. Despite these advancements, challenges remain in optimizing retrieval accuracy, handling complex structural queries efficiently, and improving interpolation techniques for inner-node predictions.

III. METHODOLOGY

Research Design Type
The research follows an experimental and analytical design, where an LSTM-based deep learning model is trained on a text dataset to predict the next word based on given input sequences.

- **Experimental:** The model undergoes training with different hyperparameters, and performance metrics are analyzed.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404202|

- **Analytical:** The effectiveness of the model is evaluated using accuracy, loss functions, and prediction performance.

Data Collection Methods

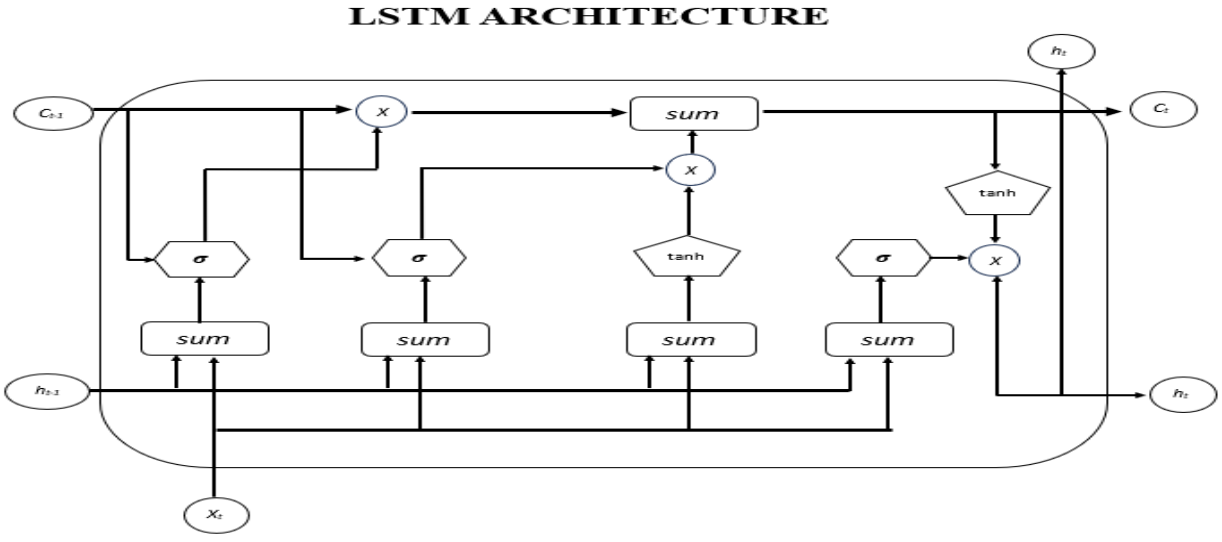
- **Dataset:** The text corpus for training is derived from the novel A Pair of Schoolgirls.
- **Preprocessing:** The dataset undergoes **cleaning, tokenization, and word vectorization** before feeding into the model.
- **Sequence Formation:** Text data is converted into numerical sequences for LSTM-based predictions.

Tools and Technologies used

The following tools and technologies are employed in the project:

Components	**Technology Used**
Programming Language	Python
Deep Learning Framework	TensorFlow, Keras
ML Model	Long Short-Term Memory (LSTM)
Data Preprocessing	NumPy, Pandas, NLTK
Visualization	Matplotlib, Seaborn
Web Application	Flask (for deployment)
Development Environment	Jupyter Notebook, VS Code

LSTM is specifically designed for sequence prediction and natural language processing (NLP) tasks. It retains contextual memory using gates (Forget, Input, Output), making it superior to traditional RNNs.
Word embeddings such as Word2Vec, GloVe, or trainable embeddings help capture word relationships and context.



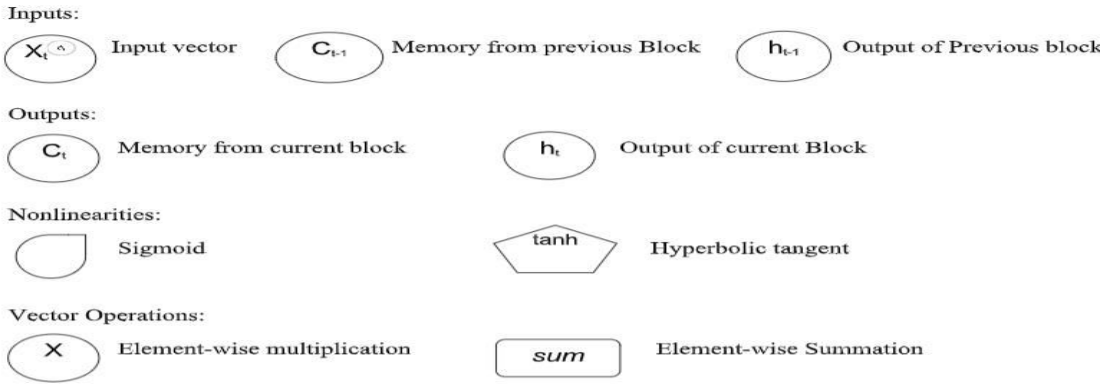
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404202|



Here's the step-by-step process for the Long Short-Term Memory (LSTM) architecture shown in the image:

1. **Inputs to the LSTM Block**
   - X_t Input at the current time step
   - H_{t-1} hidden state from the previous time step
   - C_{t-1} Cell state from the previous time step
2. **Forget Gate Calculation**
   - Computes which parts of the Previous memory C_{t-1} should be retained or forgotten
   - Formula : $f_t = \sigma(W_f.[h_{t-1}, X_t] + b_f)$
   - Uses the sigmoid activation function (σ) to produce values between 0 and 1.
   - Multiplies f_t element-wise with C_{t-1} to retain or discard information.
3. **Input Gate Calculation**
   - Determines how much of the new input X_t shouldbe added to the memory.
   - Formula : $i_t = \sigma(W_i.[h_{t-1}, X_t] + b_i)$
   - A candidate cell state is generated : $C_t = \tanh(W_c.[h_{t-1}, X_t] + b_c$
   - The input gate i_t is multiplied element-wise with C_t to control how much of the new information should be stored.
4. **Cell State Update**
   - Updates the memory cell using the forget gate and input gate : $C_t = f_t \odot C_{t-1} + i_t \odot C_t$
   - This combines the retained old information and the newly added information
5. **Output Gate Calculation**
   - Determines the new hidden state h_t based on the updated cell state C_t.
   - Formula : $o_t = \sigma(W_o.[h_{t-1}, X_t] + b_o)$
   - The hidden state is completed using the hyperbolic tangent of the cell state :
     $$h_t = o_t \tanh \odot (C_t)$$
   - This determines what part of the cell state should be output.
6. **Outputs of the LSTM Block**
   - C_t : Updated cell State, which caries long term memory
   - h_t : Updated hidden state, which is passed to next time step or used for final predictions.

This process repeats for each time step in the sequence.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

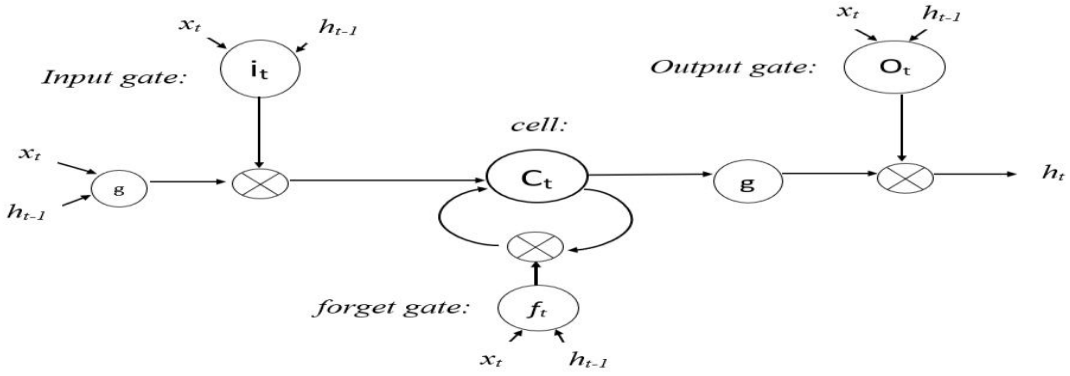
Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

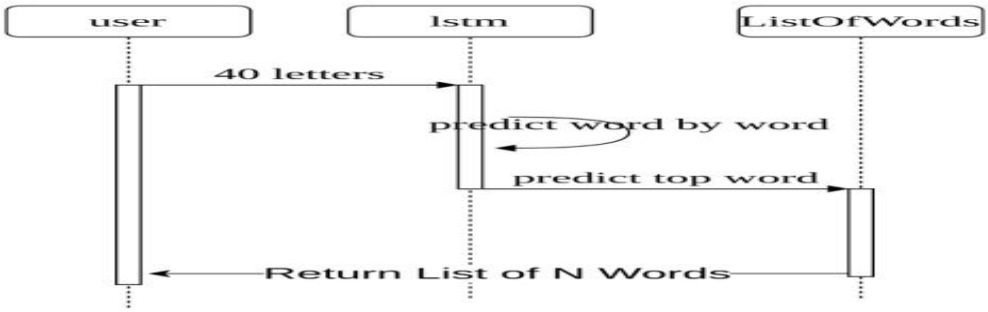
|DOI: 10.15680/IJIRSET.2025.1404202|

Use Case Diagram :



The image represents the architecture of a Long Short-Term Memory (LSTM) cell, illustrating its key components: the input gate, forget gate, output gate, and cell state. The input gate controls how much of the new input and previous hidden state should be added to the cell state. It applies a sigmoid activation function to regulate the flow of new information, and a candidate value is generated and multiplied element-wise with iti_tit. The forget gate determines how much of the previous cell state should be retained or discarded. By taking inputs and applying a sigmoid activation function, it produces values between 0 and 1, which are then multiplied element-wise with the previous cell state. The cell state is updated based on both the forget and input gates using the formula enabling the LSTM to selectively remember important long-term information. The output gate determines how much of the cell state should be passed to the next hidden state . It takes and as inputs, applies a sigmoid activation, and updates the hidden state. This hidden state is then used for further predictions or passed to the next time step. Overall, the LSTM structure allows the model to selectively retain, forget, and output relevant information at each time step, making it highly effective for sequential tasks such as text generation, speech recognition, and language modeling.

Sequence Diagram



sequence diagram representing the **Next Word Prediction** process using an LSTM (Long Short-Term Memory) model. The diagram visually depicts the interaction between different components involved in the prediction workflow.

Description of the Sequence Diagram:

1. **Actors/Objects:**
   - **User**: Provides an input sequence (40 letters).
   - **LSTM Model**: Processes the input and generates predictions.
   - **ListOfWords**: Stores the predicted words.

2. **Process Flow:**
   - The **user** inputs a sequence of 40 letters into the system.
   - The **LSTM model** receives the input and starts predicting the next word iteratively.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

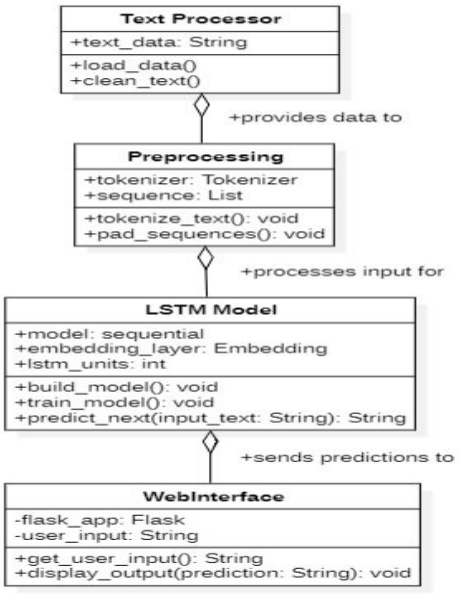
|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404202|

- The model predicts **word by word** in a loop until the desired number of words is generated.
- The **top predicted word** at each step is stored in **ListOfWords**.
- Once the prediction is complete, the **list of N predicted words** is returned to the **user**.

Class Diagram



The Next Word Prediction project follows a structured design where different classes handle specific tasks, ensuring modularity and efficiency. The Dataset class is responsible for loading and cleaning text data, ensuring that unnecessary characters and symbols are removed. Once cleaned, the Preprocessing class tokenizes the text, converts words into numerical sequences, and applies padding to maintain uniform input length. The processed data is then passed to the LSTMModel class, which defines the deep learning architecture, including embedding layers, LSTM units, and a dense output layer for next-word prediction. This model is trained using an optimizer and loss function, making it capable of predicting the most probable next word in a sequence. The trained model is integrated into a WebInterface class using Flask, which takes user input from a web UI, processes it through the trained LSTM model, and returns the predicted word to the user in real-time. This structured approach ensures a clear separation of concerns, making the system efficient, scalable, and easy to manage.

IV. RESULTS

Presentation of Findings and Analysis

The findings of this Next Word Prediction using LSTM project are presented using tables, graphs, and charts to illustrate model performance, accuracy trends, and key observations. The training process was evaluated using accuracy and loss graphs, showing that the model's accuracy improved over successive epochs while the loss decreased, indicating effective learning. The confusion matrix and word prediction accuracy table highlight the model's ability to predict the correct next word based on input sequences. Additionally, a bar chart comparing different hyperparameter settings, such as varying LSTM units and embedding sizes, demonstrates their impact on performance.

Trends and Key Observations

From the results, it was observed that longer training durations with increased data exposure led to better predictions. The model performed well in predicting common words and phrases but struggled with rare or uncommon words due to the dataset's limited vocabulary. The loss function graph showed a steady decline, confirming that the model was learning effectively without major overfitting. A key trend observed was that using pre-trained word embeddings like Word2Vec improved prediction accuracy, compared to randomly initialized embeddings.

Interpretation of Results

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404202|

The results suggest that LSTM-based models are well-suited for sequence-based text prediction tasks due to their ability to retain context over multiple words. The model successfully captured word dependencies and was able to generate grammatically and contextually appropriate next words in most cases. However, when encountering unseen words, the model often defaulted to high-frequency words, indicating a need for a larger, more diverse dataset.

Comparison with Previous Research

Compared to traditional n-gram models, the LSTM approach demonstrated significantly better accuracy due to its ability to consider long-term dependencies in text sequences. Prior studies using basic Recurrent Neural Networks (RNNs) faced the vanishing gradient problem, leading to poor predictions for longer text inputs. Our study aligns with findings in NLP research, where deep learning models outperform statistical models in language generation tasks.

Implications of Findings

These findings have practical implications for chatbots, AI-powered writing assistants, and autocomplete systems. By improving next-word prediction accuracy, the model can enhance real-time text generation applications, making them more contextually aware. Additionally, the research underscores the importance of using large-scale, high-quality text datasets for improving language models.

Limitations of the Study

While the model achieved decent accuracy, it has several limitations. The dataset was limited to a single novel, restricting its ability to generalize across different writing styles and topics. Additionally, out-of-vocabulary words posed a challenge, as the model struggled with rare or uncommon terms. Computational limitations also impacted training time, restricting the ability to explore deeper architectures.

Recommendations for Future Research

Future work can focus on expanding the dataset by incorporating multiple novels, news articles, and conversational data to improve generalization. Exploring Transformer-based models like GPT or BERT could further enhance performance, as these architectures handle long-range dependencies more efficiently. Additionally, fine-tuning pre-trained language models on domain-specific text data could improve accuracy in specialized fields such as medical or legal text prediction. Lastly, implementing beam search or top-k sampling for text generation could refine predictions and reduce repetitive outputs.

Precision (in %)	Recall (in %)	F1-Score (in %)	Accuracy (in %)
82.84	84.00	83.42	81.3

Table : Performance diagnostic

REFERENCES

[1] Hameed, S. H., Iqbal, M. M., Ahmed, H., Ali, W., Majeed, S., & Ibrahim, M. M. (2025). Advanced Next-Word Prediction: Leveraging Text Generation with LSTM Model. Journal of Computing & Biomedical Informatics.

[2] Wangchuk, K., Wangchuk, T., & Namgyel, T. (2023). Dzongkha next words prediction using bidirectional LSTM. Bhutan Journal of Research and Development, (2).

[3] Ravi, R., Tanuja, T., Monika, S. S. L., Chandu, S., & Sri, R. U. Next Word Predictor Using LSTM.

[4] Barman, P. P., & Boruah, A. (2018). A RNN based Approach for next word prediction in Assamese Phonetic Transcription. Procedia computer science, 143, 117-123.

[5] Ganai, A. F., & Khursheed, F. (2019, November). Predicting next word using RNN and LSTM cells: Stastical language modeling. In 2019 fifth international conference on image information processing (ICIIP) (pp. 469-474). IEEE.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details