*International Journal Of Engineering Research And Development*
*e- ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com*
*Volume 20, Issue 4 (April. 2024), PP. 140-144*

# Predicting Subsequent Words Using LSTM

[1] K Bhavani [2] S Rakshitha[3]S Vishal
*B.Tech Students, Dept.of CSE-DS,Sphoorthy Engineering College*
*[4] Swathi Sugur, Asst.Professor, CSE-DS, Sphoorthy Engineering College*

***Abstract***
*Next word prediction plays a crucial role in enhancing natural language processing applications, enabling more fluent and contextually relevant text generation. This project explores the implementation of Long Short-Term Memory (LSTM) networks for next word prediction. By leveraging the capabilities of LSTM, a type of recurrent neural network (RNN). This research aims to develop an efficient system for predicting the next word in a given sequence of text. Using the methodology, LSTM model architecture, training process, performance evaluation metrics, and analysis, showcasing the effectiveness and accuracy of LSTM.*
***Keywords :*** *LSTM,,RNN,Efficiency, Lingusitic,Evaluation Metrics*

---

Date of Submission: 12-04-2024 Date of Acceptance: 25-04-2024

---

## I. Introduction

Wepresent a language model-based framework designed to enhance rapid communication, particularly benefiting individuals with slower writing speeds. Our framework utilizes word prediction tools to anticipate the next word based on a given set of preceding words. This technique streamlines the process of composing text messages or emails by predicting the most probable word to follow a series of initial text fragments.

The primary objective of our framework is to expedite instant communication by offering pertinent word suggestions to users. By incorporating sophisticated algorithms, our system can accurately predict the subsequent word, thereby reducing the time and effort required for typing. This predictive capability not only improves the efficiency of communication but also aids individuals who may struggle with traditional typing methods due to physical limitations or other constraints.

Furthermore, our framework is adaptable and customizable, allowing users to modify various parameters to suit their preferences and writing styles. This flexibility ensures that the word prediction tool can effectively cater to a diverse range of users, accommodating different linguistic patterns and communication contexts.

In essence, our framework represents a significant advancement in facilitating rapid electronic communication. By harnessing the power of predictive technology, we aim to empower individuals with enhanced writing efficiency and accessibilityby fostering more seamless and inclusive communication experiences.

## II. Literature Survey

Next Word Prediction in Telugu using RNN Mechanism[6],exchanging textual content by entering information and sending it to others has become one of the most popular ways of information exchange these days.

Predicting the next word in a sequence can reduce the count of number of letters typed made by the user[10]. Next word prediction is an application of NLP, also known as language modelling.

Many applications use these systems such as, autocorrect which is primarily used in emails/messages. It is also used by MS Word, Google search that predicts the next word based on search history. We studied , the NLP applications,LSTM and a deep learning technique to carry out the next word prediction process.

## III. Methodology

In existing system we observed the pitfallthat,traditional methods face challenges in adapting to evolving linguistic patterns and changes in language usage over time. To overcome this infeasible problem LSTMs are capable of capturing long-range dependencies, adapting to evolving linguistic patterns, and modeling complex language structures, making them well-suited for improving the performance of next word prediction tasks.

**System Architecture**

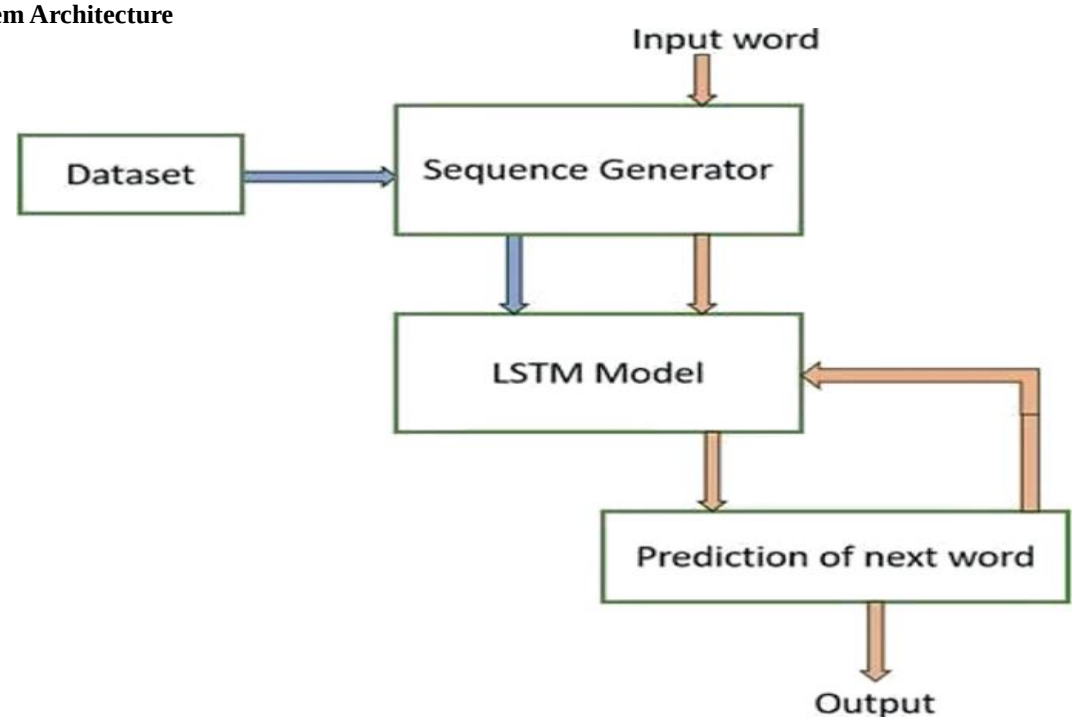


**Figure 1. System Archictecture**

**Data- Flow Diagram**

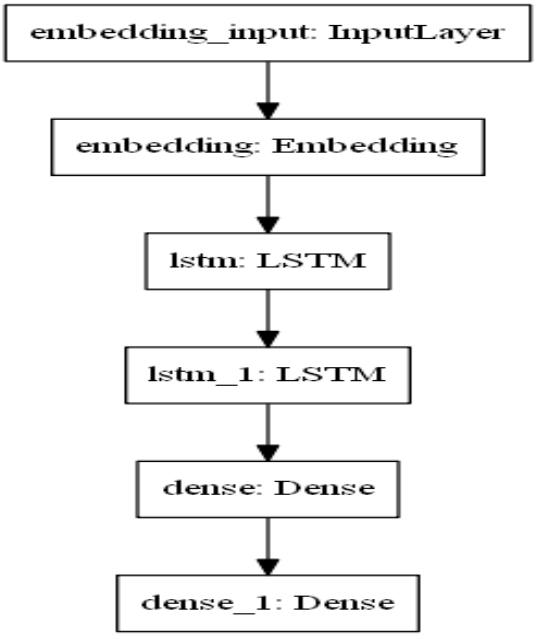


**Figure 2 . Data- Flow Diagram**

System Analysis for Predicting Subsequent Words Using LSTM:

**1. Data Collection and Preprocessing:**

- The system begins by collecting a large corpus of text data from various sources, such as books, articles, or websites.
- The text data is preprocessed to clean and standardize it, which may involve tasks like tokenization, lowercasing, removing punctuation, and splitting the text into sequences of fixed length.

**2. Feature Extraction and Encoding:**

- Each word in the corpus is represented numerically using techniques like one-hot encoding or word embeddings.
- Sequences of words are then generated, with each sequence consisting of a fixed number of input words (context) and the next word (target) to predict.

**3. Model Architecture:**

- The LSTM model architecture is defined, typically comprising one or more LSTM layers followed by one or more dense (fully connected) layers.
- The LSTM layers are responsible for capturing the long-range dependencies in the input sequences, while the dense layers help in mapping the LSTM outputs to the vocabulary space for word prediction.
- Optionally, additional layers such as dropout or batch normalization may be added to prevent overfitting and improve model generalization.

**4. Training:**

- The model is trained using the prepared dataset, where input sequences serve as input features, and the corresponding next word serves as the target label.
- During training, the model adjusts its parameters (weights and biases) iteratively to minimize a chosen loss function, such as categorical cross-entropy, by backpropagating gradients through the network.

**5. Validation and Evaluation:**

- After training, the model's performance is evaluated on a separate validation dataset to assess its generalization ability and detect overfitting.
- Evaluation metrics such as accuracy, perplexity, or BLEU score may be used to quantify the model's predictive performance.

**6. Deployment and Inference:**

- Once the model is trained and validated, it can be deployed for real-time inference.
- Given a sequence of input words, the model predicts the subsequent word(s) by feeding the input sequence through the trained LSTM network and selecting the word(s) with the highest probability according to the model's output distribution.
- The predicted word(s) can then be used to generate text, provide suggestions in text editors, or assist users in various natural language processing tasks.

**7. Monitoring and Maintenance:**

- The deployed model may require periodic monitoring to ensure its continued performance and reliability.
- Maintenance tasks may include retraining the model with new data to adapt to evolving language patterns, fine-tuning hyperparameters to improve performance, and updating dependencies to address security vulnerabilities or compatibility issues.

## IV. Result

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 56, 100)           28300

 lstm (LSTM)                 (None, 150)               150600

 dense (Dense)               (None, 283)               42733

=================================================================
Total params: 221633 (865.75 KB)
Trainable params: 221633 (865.75 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

**Figure 3 . Model Summary**

```
Epoch 1/100
27/27 [==============================] - 6s 114ms/step - loss: 5.4540 - accuracy: 0.0684
Epoch 2/100
27/27 [==============================] - 2s 92ms/step - loss: 5.0829 - accuracy: 0.0776
Epoch 3/100
27/27 [==============================] - 3s 95ms/step - loss: 5.0248 - accuracy: 0.0776
Epoch 4/100
27/27 [==============================] - 3s 94ms/step - loss: 4.9894 - accuracy: 0.0776
Epoch 5/100
27/27 [==============================] - 4s 145ms/step - loss: 4.9427 - accuracy: 0.0776
Epoch 6/100
27/27 [==============================] - 3s 93ms/step - loss: 4.8529 - accuracy: 0.0788
Epoch 7/100
27/27 [==============================] - 3s 95ms/step - loss: 4.7061 - accuracy: 0.1112
Epoch 8/100
27/27 [==============================] - 3s 96ms/step - loss: 4.5085 - accuracy: 0.1448
```

**Figure 4. Training Data**

```
1/1 [==============================] - 0s 26ms/step
what is the fee of
1/1 [==============================] - 0s 37ms/step
what is the fee of the
1/1 [==============================] - 0s 27ms/step
what is the fee of the course
1/1 [==============================] - 0s 27ms/step
what is the fee of the course fee
1/1 [==============================] - 0s 26ms/step
what is the fee of the course fee for
1/1 [==============================] - 0s 27ms/step
what is the fee of the course fee for data
1/1 [==============================] - 0s 26ms/step
what is the fee of the course fee for data science
1/1 [==============================] - 0s 37ms/step
what is the fee of the course fee for data science mentorship
1/1 [==============================] - 0s 25ms/step
what is the fee of the course fee for data science mentorship program
1/1 [==============================] - 0s 31ms/step
what is the fee of the course fee for data science mentorship program dsmp
```

**Figure 5. Model Prediction**

## V. Conclusion

Predicting subsequent words using LSTM involves a systematic process of data collection, preprocessing, model design, training, evaluation, deployment, and maintenance. By leveraging the sequential nature of text data and the memory capabilities of LSTM networks, such systems can effectively generate contextually relevant word predictions, enhancing various text-based applications and user experiences.The success of our project underscores the potential for leveraging deep learning techniques to address the inherent limitations of existing next word prediction systems. Looking ahead, the integration of LSTM networks in language modeling holds promise not only for predictive text applications but also for broader applications in natural language understanding. As technology continues to advance, our project stands as a testament to the continuous evolution of language processing systems, offering a glimpse into the future of more intelligent and contextually aware interactions with digital content.

## References

[1]. R. Sharma, N. Goel, N. Aggarwal, P. Kaur and C. Prakash, "Next Word Prediction in Hindi Using Deep Learning Techniques", 2019 International Conference on Data Science and Engineering (ICDSE), pp. 55-60, 2019.

[2]. O. F. Rakib, S. Akter, M. A. Khan, A. K. Das and K. M. Habibullah, "Bangla Word Prediction and Sentence Completion Using GRU: An Extended Version of RNN on N-gram Language Model", 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019.

[3]. Wessel Stoop and Antal van den Bosch, "Using idiolects and sociolects to improve word prediction", Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, April 2014.

[4]. Hozan K. Hamarashid, Soran A. Saeed and Tarik A. Rashid, "Next word prediction based on the N-gram model for Kurdish Sorani and Kurmanji", Neural Computing and Applications NCAA-D-19–02773R1, July 2020.
[5]. Aurnhammer Christopha and Stefan L. Frankb, "Evaluating information-theoretic measures of word prediction in naturalistic sentence reading", Neuropsychologia, vol. 134, pp. 107198, 2019, ISSN 0028-3932.
[6]. Partha Pratim Barman and Abhijit Boruah, "An RNN-based Approach for next word prediction in Assamese Phonetic Transcription", 8th International Conference on Advances in Computing & Communications (ICACC-2018), 2019.
[7]. Eisape Tiwalayo, Zaslavsky Noga and Levy Roger, "Cloze Distillation: Improving Neural Language Models with Human Next-Word Prediction", Association for Computational Linguistics, 2020.
[8]. K Smagulova and AP James, "A survey on LSTM memristive neural network architectures and applications", The European Physical Journal, 2019.