Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 1: 2025
ISSN : **1906-9685**

PREDICTIVE TEXT GENERATION WITH LONG SHORT-TERM MEMORY BASED LANGUAGE MODELS

Dr D..Ragavamsi, Assistant Professor, Seshadri Rao Gudlavalleru Engineering college (An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada), Seshadri Rao Knowledge Village, Gudlavalleru-521356, Andhra Pradesh, India
SD. Sayeema Tabassum,S.Pavan Kumar,T.Bebi Unnathi,Y.Jahnavi, IV- B. Tech CSE Department of Computer Science and Engineering, Seshadri Rao Gudlavalleru Engineering college (An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada), Seshadri Rao Knowledge Village, Gudlavalleru-521356, Andhra Pradesh, India

***Abstract*— This project focuses on building a predictive model for next-word generation using deep learning Spreprocessing a text dataset to remove noise, punctuation, and irrelevant symbols, ensuring the text is ready for tokenization. A tokenizer is employed to convert the text into sequences of tokens, mapping each word to a unique index, and generating input sequences for model training. The data is then prepared by padding sequences to ensure uniform length and splitting them into input-output pairs for training. The core of the model is an LSTM-based neural network, designed to handle sequential data effectively by capturing temporal dependencies in text. The model predicts the next word in a sequence by training on these tokenized sequences. The architecture includes embedding layers for word representations, LSTM layers to process sequences, and dense layers for classification of the next word. The trained model is then evaluated for accuracy and saved for future use. To make the model accessible and interactive, a user interface is created, allowing users to input a sentence and receive a prediction for the next word. This interface leverages the trained model and tokenizer to generate predictions in real time, providing a practical tool for exploring language generation and aiding in various applications like writing assistance and text prediction.**

***Index Terms*— LSTM, Next Word Prediction, Text Preprocessing, Tokenizer, Embedding Layer, Sequential Model, Noise Removal, Padded Sequences, Model Training, Validation Split, Gradio Interface, Prediction Function, Natural Language Processing (NLP), Ker TensorFlow, Saved Model, User Interaction, Softmax Activation, Adam Optimizer, Model Checkpoint**.

I. INTRODUCTION

The paper introduces the development of a next-word prediction system tailored to the Bodhi language using Long Short-Term Memory (LSTM) networks. With the Bodhi language being rare and linguistically underrepresented, the study addresses the challenge of creating predictive language models for it, particularly given the absence of adequate datasets and tools like Google Translate [1]. Leveraging machine learning and natural language processing (NLP), the system aims to predict subsequent words based on user input, thus
preserving and supporting the usage of this endangered language.
This paper explores the creation of a next-word prediction model utilizing LSTM networks, focusing on predicting words in the Indonesian language. The study involves collecting a dataset from various Indonesian provinces and training the model with 200 epochs. It highlights the advantages of deep learning in enhancing language modeling accuracy [2]. The model is trained and tested using TensorFlow and Keras, achieving promising results in prediction accuracy, showcasing its practical applications in NLP tasks.

https://doi.org/10.36893/JNAO.2025.V16.I01.024

The introduction focuses on the use of deep learning, specifically LSTM networks, to forecast stock prices. Acknowledging the complexity and unpredictability of stock markets, the paper highlights the importance of accurate prediction tools for informed financial decisions [3]. The study emphasizes the effectiveness of LSTMs in handling long-term dependencies and their ability to outperform traditional statistical methods, leveraging NASDAQ data for training and validation.

The paper addresses the pressing challenge of detecting fake news in today's digital age using LSTM networks. It emphasizes the societal impact of misinformation and the importance of reliable computational tools to classify news as fake or real [4]. The model incorporates NLP techniques such as word embedding and TF-IDF for text preprocessing and achieves significant accuracy. This research demonstrates the potential of machine learning in mitigating the spread of fake news and enhancing information authenticity.

This study investigates the role of LSTM-based models in language modeling, emphasizing their capacity to capture long-term dependencies in textual data. It underscores the relevance of such models in various NLP applications, such as text generation, translation, and sentiment analysis [5].The research demonstrates how LSTMs surpass traditional RNNs in predictive performance by addressing issues like vanishing gradients, thereby offering a robust solution for sequential data processing.

LITERATURE SURVEY

Aditya Kumar Singh et al. [6] explored the use of LSTM networks for stock price prediction, leveraging historical stock data to anticipate future prices. The study utilized Yahoo Finance data and highlighted the efficacy of LSTMs in handling sequence prediction problems, enabling the retention of past information for improved forecasting. The research emphasizes the importance of accurate stock prediction for informed decision-making in financial markets and demonstrates the use of regression models combined with LSTM to enhance prediction accuracy. **Dr. Nirmala C. R. et al.** [7] investigated the use of LSTM-based recurrent neural networks (RNNs) to predict crude oil prices. Recognizing the economic and political significance of oil price volatility, the study focused on applying time-series analysis to historical data for accurate forecasting. LSTM's ability to capture long-term dependencies and dynamic trends made it a suitable choice for this application, aiding governments and industries in economic policymaking and risk management. **Khushi Udaysingh Chouhan et al.** [8] presented a decentralized approach to next-word prediction using federated learning. This study implemented algorithms like FedAvg and FedProx to train models on distributed mobile devices while maintaining user privacy. By addressing issues like latency and contextual awareness, the research demonstrated the advantages of federated learning in enhancing mobile keyword prediction without compromising data security. **Podakanti Satyajith Chary et al.** [9] compared the effectiveness of Markov models and LSTM networks for text generation. The study analyzed the strengths of Markov models in capturing immediate dependencies and LSTM networks' ability to learn long-term dependencies. Through rigorous experimentation, the research provided insights into how these models generate contextually relevant and coherent text across various styles and genres. **Aswini Thota and P. Srinivasa Reddy et al.** [10] explored the use of LSTM networks for language modeling, emphasizing their ability to address long-sequence dependencies. The study demonstrated significant improvements in perplexity and word error rates compared to traditional recurrent neural networks, showcasing the applicability of LSTMs in various NLP tasks like speech recognition and spelling correction. **K. Chakradhar et al.** [11] examined the use of deep learning techniques, including N-gram modeling, CNNs, and RNNs, for next-word prediction. The research emphasized the application of these models in improving typing efficiency and reducing keystrokes. By employing LSTM networks, the study highlighted improvements in predictive accuracy and practical applications in spell correction and sentence structuring. **Afika Rianti et al.** [12] developed a next-word prediction model using LSTM, trained on Indonesian text data collected via web scraping. The research utilized TensorFlow and Keras libraries and achieved a prediction accuracy of 75%. The study demonstrated the applicability of LSTMs in language modeling and highlighted their advantages in retaining long-term dependencies for accurate word prediction.

PRELIMINARIES

A.Data Preparation:

Thedataset is imported from a Kaggle source andpreprocessed to remove noise such as special characters, numbers, and non-ASCII symbols. This ensures the data is clean and ready for tokenization.

B.Tokenization:

A tokenizer is used to map each unique word in the dataset to a corresponding integer index. This mapping helps transform the textual data into numerical format suitable for model training.

Tokenized sequences are generated by creating n-grams from the text, which serve as the input-output pairs for training.

C.Sequence Padding:

To ensure uniform sequence length, shorter sequences are padded with zeros using Keras' pad_sequences. This step aligns all input sequences to a fixed maximum length.

D.Model Design:

The model employs an embedding layer to convert integer-encoded words into dense vector representations, facilitating the capture of semantic relationships between words.Two LSTM layers are stacked to learn sequential patterns and capture long-term dependencies within the data.Fully connected dense layers are used for classification, with the final layer employing a softmax activation function to output probabilities for each word in the vocabulary.

E.Training and Evaluation:

The model is compiled using the categorical crossentropy loss function and Adam optimizer, with accuracy as the evaluation metric.The training process involves splitting the data into training and validation sets, and the model is trained for 100 epochs with a batch size of 64.A checkpoint mechanism saves the model with the best validation loss during training.

F.Model Deployment:

After training, the model is saved for deployment purposes.A user-friendly interface is created using Gradio, allowing real-time next-word predictions based on user-provided text input.

INFERENCE

During inference, the user input is tokenized, padded, and fed into the trained LSTM model to predict the next word.The model outputs a probability distribution over the vocabulary, and the word with the highest probability is selected as the predicted next word.

DATASET EXPLANATION

A. Dataset Characteristics

a)Source: The dataset is loaded from a file containing textual data (1661-0.txt).

b)Nature of Data: It consists of natural language text, likely comprising sentences or paragraphs, intended for next-word prediction tasks.

c)Structure: The text is unstructured and contains various characters, numbers, and possibly multilingual data.

*B.Labels:*The target word for each sequence is encoded as a one hot vector using the to_categorical function. The total number of categories corresponds to the size of the vocabulary.

C.Dataset Role:

a)Training: The preprocessed and tokenized data is split into training and validation subsets for model training.

b)Evaluation: The trained model is evaluated on the same dataset to measure performance.

D.Prediction: Once trained, the model uses the dataset vocabulary to predict the next word based on user input.

E.Relevance:

The dataset is crucial for:Training the LSTM-based model to capture language patterns and dependencies.

Building a vocabulary for text tokenization and predictions.Validating the model's ability to predict the next word based on historical text data.

METHODOLOGY

A.Data Acquisition and Preprocessing:

a)DataCollection: A text dataset(16610.txt) was sourced from Kaggle.

b)Noise Removal: Irrelevant characters,
numbers, special symbols, and non-ASCII text were filtered out using regular expressions.

c)Text Cleaning: Multiple spaces were
removed, and the text was split into structured sequences. Additional steps were applied to normalize and organize the data.

d)Tokenization: A tokenizer was employed to convert words into numeric indices, creating a mapping between words and unique integer values. This tokenizer was later serialized into JSON for reuse.

e)Sequence Generation: Input-output sequences were prepared by breaking down the cleaned text into smaller sequences. For each sequence, the final word became the target (output), and the preceding words were used as inputs.

*f)Padding***:** All sequences were padded to a uniform length to ensure compatibility with the neural network model.

B. Model Architecture Design

*a) Embedding Layer***:** Used to transform word indices into dense vector representations of a fixed size (100 dimensions).

b) LSTM Layers: Two stacked LSTM layers were implemented to capture sequential dependencies, with 500 and 300 units, respectively.

*c) Dense Layers***:** A dense layer with 256 units and ReLU activation followed by a final output layer with a softmax activation function to classify the next word.

d) Output Layer: Configured for multi-class classification, where the number of classes corresponds to the vocabulary size.

C. Model Compilation and Training

a) Loss Function: Categorical Crossentropy, suitable for multi-class classification.

b) Optimizer: Adam optimizer, chosen for its efficiency in training deep learning models.

c) Metrics: Accuracy was used to evaluate model performance during training.

d) Training Configuration: The model was trained for 100 epochs with a batch size of 64, using 20% of the data as the validation set.

*e) Checkpointing***:** A ModelCheckpoint callback was implemented to save the model with the best validation loss.

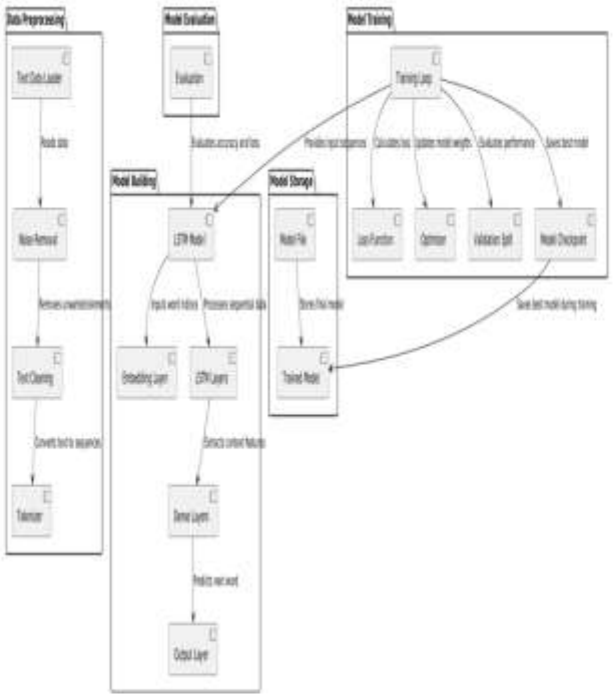


Fig 1**:** Component diagram

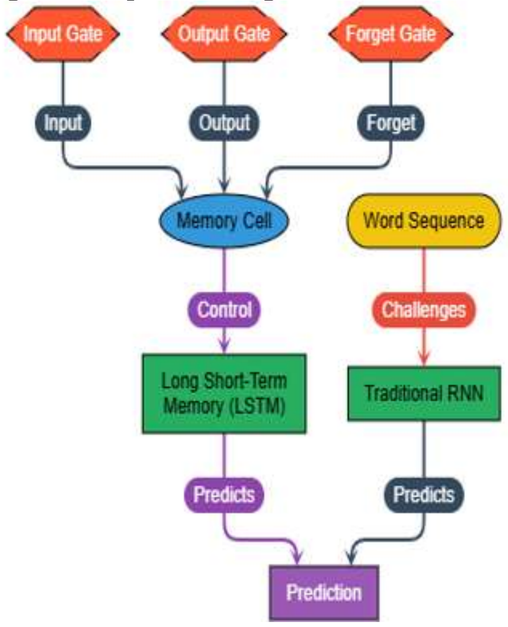


Fig 2**:** Architecture Diagram

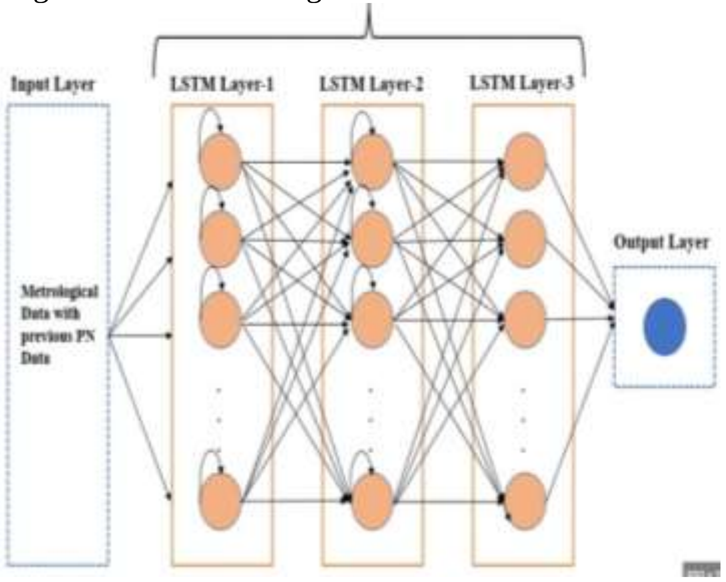


Fig 3**:** LSTM layers

D. Evaluation and Saving the Model:The model's performance was evaluated on the training data to compute accuracy and loss.The trained model was saved in .h5 format for deployment.

E. Deployment and User Interface

a)Gradio Interface: A Gradio-based web interface was created to allow real-time user interaction.

*b) Prediction Function****:*** The function takes a user-input sentence, tokenizes and pads it, and uses the trained LSTM model to predict the next word.

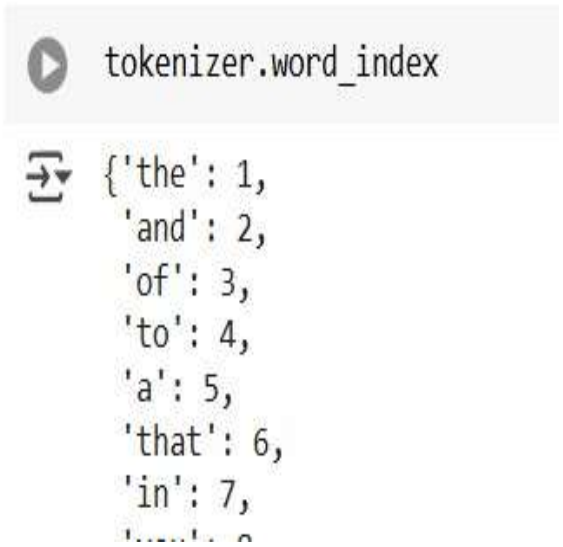
c) Privacy Consideration: The prediction process avoids sharing sensitive data by working on local sequences provided by users

Fig1 explains the component diagram of the LSTM model. Fig2 explains the architecture of LSTM for the precdiction of next word. Fig3 explains the layers of LSTM.

EXPERIMENTAL RESULTS

Fig4 shows the tokenization of words and allocation of index to them. Fig5 shows the sequences of indexes. Fig6 represents the LSTM model. Fig7 represents the epochs of the model. Fig8 and Fig9 shows the sample results of LSTM model.

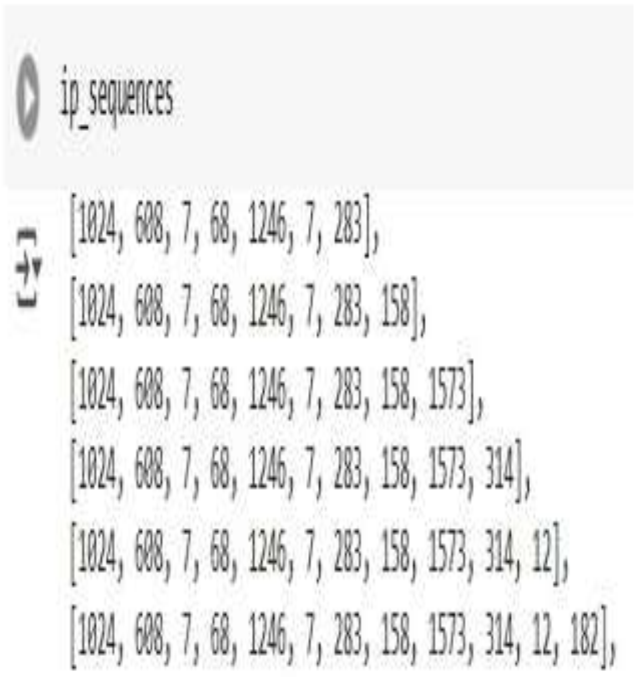
RESULTS



```
tokenizer.word_index
```

```
{'the': 1,
 'and': 2,
 'of': 3,
 'to': 4,
 'a': 5,
 'that': 6,
 'in': 7,
```

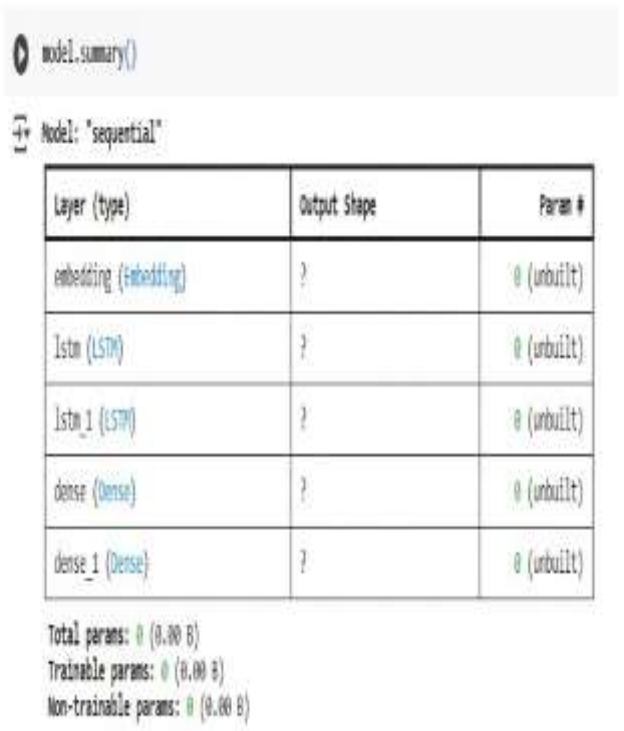
Fig 4**:** Tokenizer



```
ip_sequences

[1824, 608, 7, 68, 1246, 7, 283],
[1824, 608, 7, 68, 1246, 7, 283, 158],
[1824, 608, 7, 68, 1246, 7, 283, 158, 1573],
[1824, 608, 7, 68, 1246, 7, 283, 158, 1573, 314],
[1824, 608, 7, 68, 1246, 7, 283, 158, 1573, 314, 12],
[1824, 608, 7, 68, 1246, 7, 283, 158, 1573, 314, 12, 182],
```

Fig 5:Sequences



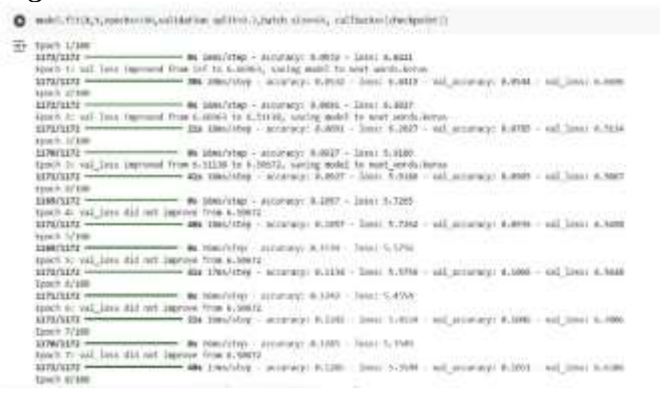
```
model.summary()

Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
lstm (LSTM)	?	0 (unbuilt)
lstm_1 (LSTM)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)
dense_1 (Dense)	?	0 (unbuilt)

```
Total params: 0 (0.00 B)
Trainable params: 0 (0.00 B)
Non-trainable params: 0 (0.00 B)
```

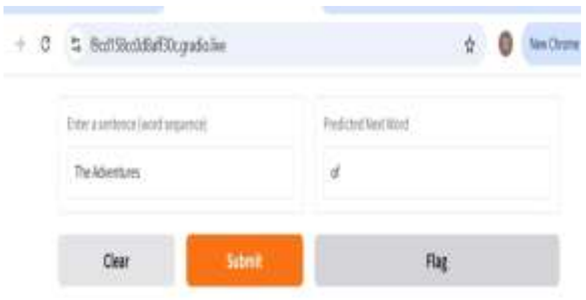
Fig 6:LSTM Model



```
model.fit(X,y,epochs=100,validation_split=0.1,batch_size=64, callbacks=[checkpoint])

Epoch 1/100
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.0519 - loss: 6.8221
Epoch 1: val_loss improved from inf to 6.60963, saving model to next_words.keras
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 28s 20ms/step - accuracy: 0.0532 - loss: 6.8419 - val_accuracy: 0.0544 - val_loss: 6.6096
Epoch 2/100
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.0691 - loss: 6.3027
Epoch 2: val_loss improved from 6.60963 to 6.51130, saving model to next_words.keras
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 22s 18ms/step - accuracy: 0.0691 - loss: 6.2027 - val_accuracy: 0.0785 - val_loss: 6.5114
Epoch 3/100
1170/1172 ━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.0927 - loss: 5.9100
Epoch 3: val_loss improved from 6.51130 to 6.50572, saving model to next_words.keras
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 42s 19ms/step - accuracy: 0.0927 - loss: 5.9168 - val_accuracy: 0.0909 - val_loss: 6.5057
Epoch 4/100
1169/1172 ━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.1057 - loss: 5.7265
Epoch 4: val_loss did not improve from 6.50572
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 40s 19ms/step - accuracy: 0.1057 - loss: 5.7262 - val_accuracy: 0.0990 - val_loss: 6.5088
Epoch 5/100
1168/1172 ━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.1134 - loss: 5.5756
Epoch 5: val_loss did not improve from 6.50572
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 41s 17ms/step - accuracy: 0.1136 - loss: 5.5756 - val_accuracy: 0.1000 - val_loss: 6.5648
Epoch 6/100
1171/1172 ━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.1242 - loss: 5.4554
Epoch 6: val_loss did not improve from 6.50572
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 22s 19ms/step - accuracy: 0.1241 - loss: 5.4554 - val_accuracy: 0.1006 - val_loss: 6.6006
Epoch 7/100
1170/1172 ━━━━━━━━━━━━━━━━━━━ 0s 16ms/step - accuracy: 0.1285 - loss: 5.3549
Epoch 7: val_loss did not improve from 6.50572
1172/1172 ━━━━━━━━━━━━━━━━━━━━ 40s 17ms/step - accuracy: 0.1286 - loss: 5.3549 - val_accuracy: 0.1051 - val_loss: 6.6100
Epoch 8/100
```

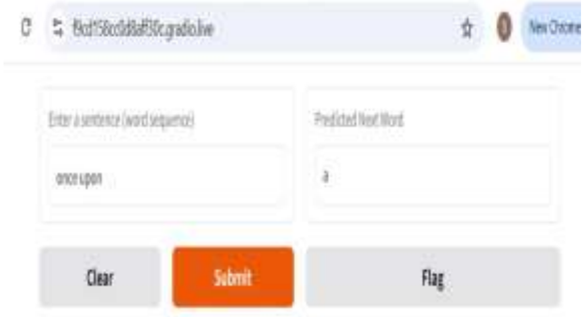
Fig 7:Epochs



Enter a sentence (word sequence)
The Adventures

Predicted Next Word
of

Fig 8:Sample Result(1)



Enter a sentence (word sequence)
once upon

Predicted Next Word
a

Fig 9:Sample Result(2)

CONCLUSION

The implementation of the next-word prediction system using LSTM demonstrates the effectiveness of deep learning in handling sequential data for language modeling tasks. By leveraging the power of Long Short-Term Memory (LSTM) networks, the model successfully captures both short-term and long-term dependencies in text, enabling accurate word predictions. The data preprocessing pipeline ensures clean and structured input for the model, while the tokenizer and embedding layers provide an efficient representation of the textual data.

The integration of a user-friendly Gradio interface allows for real-time interaction, making the system accessible and practical for various applications such as writing assistance, language learning, and text auto-completion. Additionally, the use of
advanced techniques like padding, categorical encoding, and model checkpointing ensures robust training and deployment processes.

Overall, the project highlights the potential of LSTM-based architectures in natural language processing, offering a scalable solution for enhancing user productivity and interaction with text-based systems. Future improvements could include training on larger and multilingual datasets, implementing transformer-based models for enhanced accuracy, and optimizing the system for deployment on edge devices.

REFERENCES

[1] Kumar A., Mishra P. K., Namgail T., Kumar S. "Next Word Prediction in Bodhi Language Using LSTM-based Approach." *International Journal of Computer Applications Technology and Research*, Vol. 12, Issue 05, 2023, pp. 21-27. DOI: 10.7753/IJCATR1205.1005.
[2] Singh S., Patil V. "Stock Prediction Overview and a Simple LSTM-Based Prediction Model." *International Research Journal of Engineering and Technology (IRJET)*, Vol. 07, Issue 04, 2020, pp. 5935-5939.
[3] Yesugade T., Kokate S., Patil S., Varma R., Pawar S. "Fake News Detection Using LSTM." *International Research Journal of Engineering and Technology (IRJET)*, Vol. 08, Issue 04, 2021, pp. 2500-2504.
[4] Rianti A., Widodo S., Ayuningtyas A. D., Hermawan F. B. "Next Word Prediction Using LSTM." *Journal of Information Technology and Its Utilization*, Vol. 5, Issue 1, 2022, pp. 10-15. DOI: 10.56873/jitu.5.1.4748.
[5] Chary P. S. "Text Generation: Using Markov Model & LSTM Networks to Generate Realistic Text." *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Vol. 11, Issue XII, 2023, pp. 1323-1328. DOI: 10.22214/ijraset.2023.57601.
[6] Chouhan K. U., Jha N. P. K., Jha R. S., Kamaluddin S. I., Giri N. "Mobile Keyword Prediction Using Federated Learning." *International Journal for Research in Applied Science & Engineering Technology*

(IJRASET), Vol. 11, Issue IV, 2023, pp. 3144-3149. DOI: 10.22214/ijraset.2023.50826.
[7] Thota A., Reddy P. S. "Word-Based Prediction Using LSTM Neural Networks for Language Modeling." *International Journal of Creative Research Thoughts (IJCRT)*, Vol. 9, Issue 8, 2021, pp. e216-e220.
[8] Chakradhar K., Kiran K. S., Shanmukh K., Kumar K. S., Sagar K. D. "Next Word Prediction Using Deep Learning." *International Journal of Research Publication and Reviews*, Vol. 3, No. 11, 2022, pp. 1182-1189.
[9] Rianti A., Widodo S., Ayuningtyas A. D., Hermawan F. B. "Next Word Prediction Using LSTM." *Journal of Information Technology and Its Utilization*, Vol. 5, Issue 1, 2022, pp. 10-15. DOI: 10.56873/jitu.5.1.4748.
[10] Kumar A., Mishra P. K., Namgail T., Kumar S. "Next Word Prediction in Bodhi Language Using LSTM-based Approach." *International Journal of Computer Applications Technology and Research*, Vol. 12, Issue 05, 2023, pp. 21-27. DOI: 10.7753/IJCATR1205.1005
[11] Singh A. K., Gupta A., Rabbani F., Yadav A., Singh A. P. "Stock Prediction Using LSTM Technique." *International Journal for Research in Applied Science & Engineering Technology (*IJRASET*)*, Vol. 12, Issue V, May 2024, pp. 1352-1355. DOI: 10.22214/ijraset.2024.61815.
[12] Dr. Nirmala C. R., Kumar A., Ajay K., Rangaswamy V. R., Shivkumar S. "Forecasting Price of the Crude Oil Using LSTM Based on RNN." *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Vol. 10, Issue VII, July 2022, pp. 196-203. DOI: 10.22214/ijraset.2022.45094