Enhancing Text Composition: A Transformer-based Approach to Sentence Auto-completion

Sri Vatsanka, Hema Radhika Reddy, Sachin Kumar S
Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, India

***Abstract*—The goal of this project report is to improve the accuracy and efficiency of text writing by investigating the creation and application of a transformer-based method for sentence auto-completion. The auto-completion system was built and assessed using both conventional n-gram models and cutting-edge deep learning architectures, such as Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM). The LSTM and BiLSTM models were trained on a vast corpus of text data to predict the words that would come after in a phrase, with the n-gram model acting as a baseline. According to the data, the LSTM and BiLSTM models perform noticeably better than the n-gram model; the BiLSTM model has an accuracy of 82.9%.The results highlight how cutting-edge deep learning methods can boost applications in natural language processing.**

***Index Terms*—Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM),n-gram model.**

I. Introduction

By anticipating the next word or phrase based on the context, sentence auto-completion plays a critical role in improving communication efficiency. This reduces errors, speeds up typing, and even inspires creativity. In many applications, such as writing aids, search engines, and communication tools, this technology is essential since it makes it easier and more efficient for users to express their ideas. Sentence auto-completion is a challenging problem in Natural Language Processing (NLP) that necessitates correct contextual knowledge and generation of human language.

Conventional approaches to sentence auto-completion frequently depended on statistical language models and n-grams [1]. Although helpful, these methods had trouble collecting context and long-term dependencies that went beyond a few sentences before. Consequently, they often generated completions that were inconsistent and irrelevant when applied to lengthier phrases. The emergence of neural network-based techniques, especially those utilising Long Short-Term Memory (LSTM) networks, represented a major advancement by more efficiently taking sequential data into account. However, because LSTMs are sequential processors by design, they were still limited in their ability to interpret long-range dependencies. By removing these restrictions, the emergence of transformer models transformed natural language processing.

Transformers can analyse complete sentences at once thanks to their self-attention processes, which improves their ability to capture complex relationships and contextual information [2]. This capacity to take into account the context of the complete sentence results in more relevant and accurate auto-completions, which significantly improves user experience [3]. Sentence auto-completion is just one of the NLP tasks where the transformer architecture, demonstrated by models like GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers), has set new standards. Despite these advancements, current text composition tools often lack the sophistication to suggest contextually-aware continuations, hindering writing fluency and user productivity [4]. This project aims to bridge this gap by leveraging transformer models to develop an advanced sentence auto-completion system. By harnessing the power of transformers, we seek to provide more accurate, relevant, and contextually appropriate suggestions, thereby enhancing text composition and communication efficiency [5]. The problem statement of this project revolves around enhancing text prediction capabilities through the development and implementation of robust models utilizing both traditional n-gram techniques and advanced LSTM architectures [6]. The primary objective is to address the challenges associated with predicting the next word in a sequence of text by exploring various preprocessing, cleaning, and model building approaches. Specifically, the project aims to achieve the following objectives:

- Develop a comprehensive preprocessing pipeline to clean and prepare text data for model training.
- Implemented n-gram models to capture local word dependencies and calculate conditional probabilities.
- Construct LSTM architectures to capture long-term dependencies and mitigate the vanishing gradient problem.
- Explored the integration of Bidirectional LSTM (BiLSTM) to enhance context capture in sequential data.

II. Related Works

The literature on next word prediction encompasses various models and datasets, highlighting the evolution from traditional methods to advanced deep learning techniques. Rianti et al. (2022) and Soam et al. (2022) demonstrate the effectiveness of LSTM and BiLSTM models over n-gram models, with notable accuracy improvements. Cruz-Benito et al. (2021) and Aguiar and Fadavi illustrate the application of deep learning in code auto-completion, achieving significant accuracy with models like GPT-2 and BERT. Studies like Tiwari et al. (2022) and Hoque et al. (2023) emphasize the versatility of deep learning across different languages, achieving high prediction accuracies. Overall, the trend shows a clear preference for

deep learning models, particularly LSTM and BiLSTM, for superior performance in next word prediction tasks. [conference]IEEEtran

III. Dataset Description

The Kaggle dataset "Tweets, Blogs, News" is accessible. The "SwiftKey Dataset (4 million)" is an extensive compilation of textual information gathered from multiple web sources, such as news articles, blogs, and social media tweets. With a whopping 4 million records, this dataset provides a large and varied corpus for study on machine learning and natural language processing (NLP). Because of its wide range of use, which includes a variety of writing genres and styles, researchers can investigate sentiment analysis, linguistic patterns, and text generating jobs. This dataset is a priceless tool for researching language trends and evolution since it offers insightful information on how language is used in modern society across various online platforms. Its large-scale nature also makes it possible to build and assess reliable NLP models that can efficiently handle text data from the real world.

IV. Model Description

In this section, we outline the rationale behind the selection of models used in our project for text prediction tasks, including the n-gram model, LSTM, and BiLSTM architectures.

N-GRAM MODEL: A basic statistical language model called the n-gram model uses the occurrence rates of n-grams—where n is the number of consecutive words taken into consideration—to predict the likelihood of the following word in a sequence. The text corpus was subjected to trigram, bigram, and unigram models in order to identify local word relationships.

LSTM MODEL: Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem inherent in traditional RNNs. LSTM consists of memory cells with gated units, allowing for the retention and retrieval of information over long sequences.We opted for LSTM due to its capability to capture long-range dependencies and retain relevant information over extended sequences. The architecture's gated units, including input, output, and forget gates, facilitate selective information retention, making it well-suited for learning complex patterns and sequences prevalent in text data.

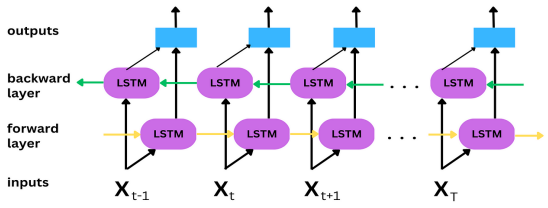

Fig. 1: Architecture of BILSTM

BILSTM MODEL:

Bidirectional Long Short-Term Memory (BiLSTM) extends the LSTM architecture by processing input sequences in both forward and backward directions through two separate LSTM layers. The outputs of both layers are concatenated to capture contextual information from past and future contexts.We chose BiLSTM for its ability to leverage context from both preceding and succeeding words, providing a more comprehensive understanding of the input sequence. By considering information from both directions, BiLSTM enhances the model's capacity to capture nuanced dependencies within the text data, ultimately leading to improved prediction accuracy, especially in tasks reliant on contextual understanding.

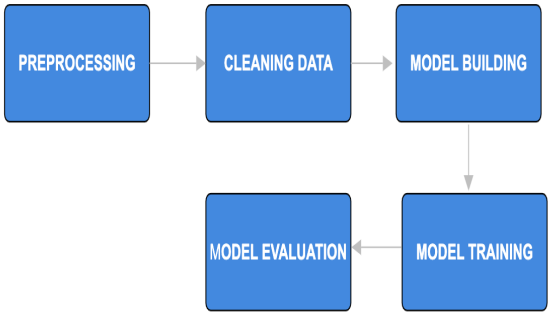

Fig. 2: Proposed Methodology

V. Proposed Methodology

We employ a structured approach comprising preprocessing, cleaning, and model building steps to develop robust text prediction models utilizing LSTM and n-gram techniques.

We preprocess the dataset by splitting it based on the newline character, removing emojis, special characters, and leading/trailing spaces, and tokenizing sentences. Empty sentences are dropped to ensure data integrity. Additionally, we create a cleaning function to construct a frequency dictionary, handle out-of-vocabulary (OOV) words, and incorporate unique tokens to enrich the dataset.

We utilize n-gram models to count the occurrences of n-grams and calculate conditional probabilities of words following given sequences. Maximum likelihood estimation with k-smoothing is applied to handle zero probabilities and suggest the most likely next word based on the previous tokens. For LSTM model construction, we delve into its key components including cell state, forget gate, input gate, and output gate, enabling the network to capture long-term dependencies and mitigate the vanishing gradient problem. We integrate Bidirectional LSTM (BiLSTM) to further enhance sequence modeling by leveraging information from both directions, ensuring better context capture even in complex sentences.

VI. Results and Discussions

The image shown in fig 3 is the accuracy of model. The y-axis of the graph is labeled "accuracy" and ranges from 0.1

Study	Study Purpose	Dataset	Methods/Techniques	Relevant Findings	Results (Accuracy)
Rianti, Afika, et al. (2022) [1]	Next word prediction using LSTM	data of 180 Indonesian destinations from nine provinces	LSTM	LSTM models can effectively predict the next word in a sequence, showing potential for applications in various text-based tasks.	75/% for LSTM
Soam, Milind, and Sanjeev Thakur (2022) [2]	Comparative study of deep learning methods	Medium articles Dataset(6508 articles)	Deep learning (various architectures)	Different deep learning architectures, including LSTM and GRU, have varying effectiveness in next word prediction, with LSTM generally performing well.	58.27% for LSTM, 66.1% for Bi-LSTM
Cruz-Benito, Juan, et al. (2021)	[3]	GitHub CodeSearchNet Challenge dataset	Deep learning models (various)	Language models like LSTM and GPT-3 can assist in code generation and auto-completion, enhancing developer productivity and accuracy.	49.48 % for AWD-LSTM word, 55.72% for AWD-LSTM unigram, 58.03% for AWD-LSTM BPE, 77.96% for AWD-LSTM char, 51.57% for AWD-QRNN word, 53.95% for AWD-QRNN unigram, 53.82% for AWD-QRNN BPE, 73.63% for AWD-QRNN char, 74.37% for GPT-2, 99.92% for BERT, 99.94% for RoBERTa
Aguiar, Rui, and Faraz Fadavi [4]	Keyword-based code auto-complete	java-small dataset	Custom project (specific techniques not detailed)	Custom methods tailored to keyword-based auto-completion can improve coding efficiency and reduce errors.	metrics used is average edit distance 31.14 for trigram, 27.34 for 4 gram, 24.27 for 5 gram, 2.2768 for neural LSTM, 4.1196 for neural Transformer
Hariharan, U. (2024) [5]	LSTM-based next keyword prediction	HC Corpora	LSTM	LSTM can predict the next keyword with reasonable accuracy, showing promise for applications in text and code prediction.	accuracy between 10% and 18% with a sample size of 1-10% of the original corpus.
Siami-Namini, Sima, Neda Tavakoli, and Akbar Siami Namin (2019) [7]	Forecasting time series with LSTM and BiLSTM	daily, weekly, and monthly time series of some stock data for the period of Jan 1985 to Aug 2018(Yahoo finance Website)	LSTM, BiLSTM	Both LSTM and BiLSTM are effective for time series forecasting, with BiLSTM often providing superior performance due to its bidirectional nature.	RMSE metrics is used avg of 39.09 for LSTM, 20.17 for Bi-LSTM
Tiwari, Aditya, Neha Sengar, and Vrinda Yadav (2022) [8]	Next word prediction using deep learning	IIT Bombay English-Hindi parallel corpus	Deep learning (specific architecture not detailed)	Deep learning models are effective for next word prediction, demonstrating significant improvements over traditional statistical methods.	91.78% for LSTM, 92.12% Bi-LSTM
Hoque, Afranul, et al. (2023) [6]	Next word prediction in Bangla using GRU-based RNN	Bangla language text dataset	GRU-based RNN, N-Gram Language Model	GRU-based RNN models combined with N-Gram techniques perform well in predicting the next word in Bangla, showcasing the adaptability of these models.	81.22% for unigram, 89.31% for bigram, 97.69% for trigram, 99.43% for 4 gram, 99.78% for 5 gram
Niharika, P., and S. John Justin Thangaraj (2023) [9]	Automatic next word generation using LSTM vs N-gram	dataset from Gutenberg website	LSTM, N-gram model	LSTM model significantly outperforms the N-gram model in next word generation tasks for text-based applications.	85% for LSTM, 65% for N-gram model

TABLE I: Summary of Studies on Next Word Prediction

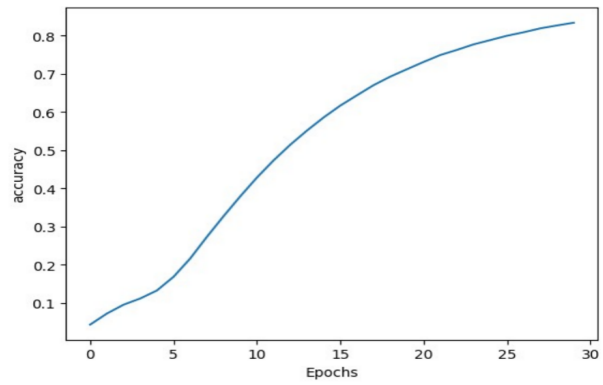

Fig. 3: Accuarcy of BILSTM

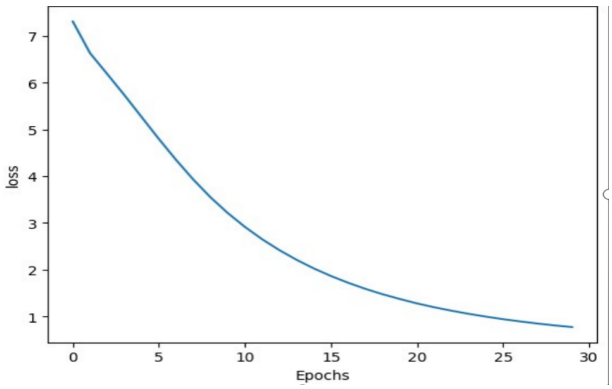

Fig. 4: Training Loss of BILSTM

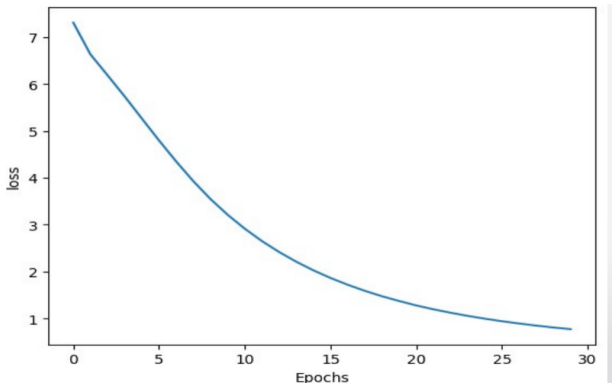

Fig. 5: Training Loss of LSTM

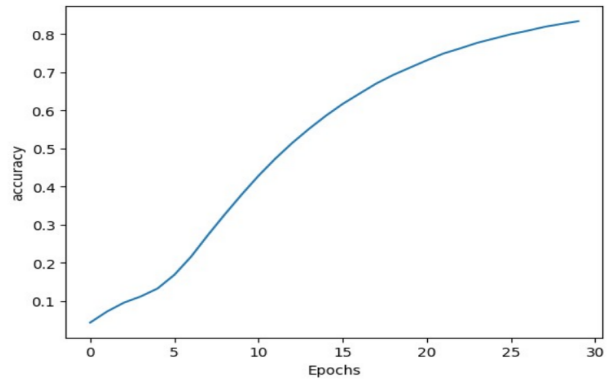

Fig. 6: Accuracy of LSTM

to 0.8. The x-axis is labeled ”Epochs” and ranges from 0 to 30. There are vertical grid lines at every 5 epochs.The line on the graph shows the accuracy of a model over time. The accuracy is measured on a scale from 0 to 1, with 1 being the most accurate. The line starts at a low accuracy and gradually increases over time. This suggests that the model is learning and improving its accuracy as it is trained on more data. The image shown in fig 4 is the training loss of graph.The y-axis is labeled ”loss” and has values ranging from 1 to 7. The x-axis is labeled ”Epochs” with values from 0 to 30. There are vertical grid lines at every 5 epochs.

Conclusion and Future Directions

In conclusion, our project successfully implements preprocessing, cleaning, and model building techniques to develop text prediction models using both traditional n-gram and advanced LSTM architectures. Through systematic preprocessing and cleaning, we ensure data integrity and enhance model performance.

Moving forward, we aim to explore additional techniques for improving model accuracy, such as fine-tuning hyperparameters, incorporating attention mechanisms in LSTM models, and experimenting with different neural network architectures. Additionally, we plan to expand the dataset to include a wider range of sources and languages to enhance model generalization.

Furthermore, we intend to explore transfer learning approaches to leverage pre-trained language models for text prediction tasks. This could involve fine-tuning models such as BERT or GPT on our dataset to leverage their contextual understanding capabilities.

Overall, our project lays the foundation for further research in text prediction and natural language processing, with potential applications in autocomplete systems, chatbots, and predictive typing tools. By continually refining and expanding our models, we aim to contribute to advancements in language understanding and generation technologies.

References

[1] A. F. Ganai and F. Khursheed, “Predicting next word using rnn and lstm cells: Stastical language modeling,” in *2019 fifth international conference on image information processing (ICIIP)*. IEEE, 2019, pp. 469–474.

[2] K. Jain and S. Kaushal, “A comparative study of machine learning and deep learning techniques for sentiment analysis,” in *2018 7th International conference on reliability, infocom technologies and optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2018, pp. 483–487.

[3] J. Cruz-Benito, S. Vishwakarma, F. Martin-Fernandez, and I. Faro, “Automated source code generation and auto-completion using deep learning: Comparing and discussing current language model-related approaches,” *AI*, vol. 2, no. 1, pp. 1–16, 2021.

[4] N. Agrawal and M. Swain, “Auto complete using graph mining: A different approach,” in *2011 Proceedings of IEEE Southeastcon*. IEEE, 2011, pp. 268–271.

[5] N. Choudhury, F. Faisal, and M. Khushi, “Towards an lstm-based predictive framework for literature-based knowledge discovery,” *arXiv preprint arXiv:1907.09395*, 2019.

[6] 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," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE, 2019, pp. 1–6.

[7] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International conference on big data (Big Data)*. IEEE, 2019, pp. 3285–3292.

[8] M. Soam and S. Thakur, "Next word prediction using deep learning: A comparative study," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2022, pp. 653–658.

[9] P. Niharika and S. J. J. Thangaraj, "Long short term memory model-based automatic next word generation for text-based applications in contrast to the n-gram model," *Journal of Survey in Fisheries Sciences*, vol. 10, no. 1S, pp. 1907–1915, 2023.