VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering and Information Systems

Fall Semester 2024-2025

Department of Computer Applications

PMCA698J – Dissertation -1 / Internship -1

Review -2

Exploring Modern Deep Learning Architectures for Bengali Sentence Auto-Completion

23MCA0304 – Sayantan Bhattacharyya

Under the Guidance of

Dr Jagannathan J

Associate Professor Senior Grade - I

SCORE

Guide Signature with date

Dr Jagannathan J

A. LITERATURE SURVEY:

[1] In their study, Kyume et al. utilized a Bi-directional LSTM with an attention mechanism to develop a Bangla sentence autocompletion model. They achieved high training accuracies across various n-gram models, with 97.98% for 7-gram, 97.97% for 6-gram, 97.91% for 5-gram, and 97.38% for 4-gram. However, the accuracy decreased for Tri-gram (90.33%) and Bi-gram (38.39%) models, indicating the model's effectiveness in longer contexts but challenges with shorter sequences.

[2] Rakib and his team created a GRU-based RNN for predicting Bangla sentences using n-gram datasets. Their model achieved impressive accuracies, with 99.70% for 5-gram, 99.24% for 4-gram, and 95.84% for Tri-gram models. However, their performance declined with shorter n-grams, resulting in 78.15% for Bi-gram and 32.17% for Uni-gram models.

[3] Islam, Amin, and Zereen introduced a new Bi-LSTM model for Bangla sentence completion, with a focus on word prediction. Their approach achieved 99% accuracy for both 4-gram and 5-gram predictions. The model also showed marked improvement over existing methods, with accuracies of 95% for Tri-gram, 75% for Bi-gram, and 35% for Uni-gram predictions. This demonstrates the Bi-LSTM's effectiveness in both short and long-context predictions.

[4] Nobel, Sultana, Tasir, and Rahman proposed a novel approach for Bangla word completion and sequence prediction by integrating a Trie data structure with CNN, LSTM, and n-gram methodologies. Their model achieved accuracies of 99.70% for 5-gram, 99.24% for 4-gram, 95.84% for Tri-gram, 78.15% for Bi-gram, and 32.17% for Uni-gram predictions. This integration highlights the potential of combining neural networks with traditional data structures to enhance language model performance across different contexts.

[5] Islam et al. addressed the problem of Bangla sentence correction and autocompletion using a decoder-encoder based sequence-to-sequence RNN with LSTM cells. They constructed a benchmark dataset incorporating word misarrangement, missing words, and sentence completion tasks. After training their model on this

dataset, they achieved 79% accuracy on the test set, highlighting the effectiveness of sequence-to-sequence models in Bangla sentence processing.

[6] Latief et al. investigated the effectiveness of four preprocessing techniques—tokenization, case-folding, stopword removal, and stemming—in predicting the next word in Indonesian text documents. They utilized word2vec embeddings and LSTM networks for their study, evaluating the results through a human perception approach and evaluation matrix focusing on the suitability of word sequences. Their findings revealed that combining tokenization and case-folding enhanced sentence meaning, whereas including stopword removal and stemming led to meaning loss and overlap. The model achieved its best performance with one-word n-grams, yielding a precision of 0.1, recall of 1.0, and an F-score of 0.1.

[7] Lakshmi et al. aimed to predict the next word and sequence of words in Telugu sentences using a stochastic approach. They employed various n-gram linguistic models, including uni-gram, bi-gram, tri-gram, maximum likelihood estimation, Laplace, and add-one smoothing, to enhance automatic sentence completion by predicting appropriate words, thus saving time, reducing keystrokes, and minimizing misspellings. Their evaluation showed perplexity scores for different dictionaries: Basic Dictionary achieved a Bigram Perplexity of 4.276 and Trigram Perplexity of 2.178; Add-One Dictionary had Bigram Perplexity of 33.565 and Trigram Perplexity of 31.950; Add-Alpha Dictionary recorded Bigram Perplexity of 28.403 and Trigram Perplexity of 25.055. Lower perplexity values indicate that their models performed better in predicting word sequences.

[8] Padmanandam et al. implemented two key features: Text Auto-completion and Text Generation. For the Text Auto-completion model, they utilized LSTM and N-gram models, while the Text Generation model was built using a GRU model. This approach combines different neural network architectures to enhance both word prediction and text generation tasks.

[9] Sarker et al. proposed an integrated methodology combining a trie data structure, sequential LSTM, and N-gram for word completion and sequence prediction in Bangla. The trie was used to store Bangla vocabulary and retrieve words based on user-input prefixes. For sequence prediction, they employed a hybrid approach of LSTM and N-

gram, which outperformed individual models. Their model was evaluated on Bangla datasets, achieving two different datasets, 84% accuracy on dataset A and 81% on dataset B, demonstrating improved efficiency in Bangla language processing.

[10] Yukino Ikegami, Rainer Knauf, Setsuo Tsuruta, Andrea Kutics, and Ernesto Damiani addressed the challenge of Japanese text input. Japanese text input involves multiple character sets such as Kanji, Hiragana, and Katakana, making next word prediction essential yet complex. To tackle this challenge, they proposed a hybrid language model that combines an RNN with an n-gram model. Their model, which is compact with 2 LSTMs and 650 units per layer, reduces perplexity by 10% compared to conventional models like Zaremba's. When incorporated into their IME tool, Flick, the model outperformed Google Japanese Input (Mozc) by 16% in input time and 34% in keystrokes, highlighting its efficiency in Japanese text prediction tasks.

[11] Surendra, Schilling, Stoewer, Maier, and Krauss explored whether word classes are innate or emerge during language acquisition by training a deep neural network to predict the next word in a sequence. They used a neural network consisting of four bi-directional LSTM layers, followed by a flatten layer and a dense output layer, to analyze the activation patterns within the hidden layers. Their findings revealed that the network's internal representations of word sequences clustered according to the word class of the predicted word, despite the network not receiving explicit syntactic information during training. This suggests that word classes may naturally emerge as a consequence of predictive coding, similar to processes in the human brain during language acquisition.

[12] Sharaheena T and Sabitha S conducted a survey on the evolution of next word prediction techniques, focusing on the transition from statistical methods to more accurate neural network models such as LSTM, Bi-LSTM, and GRU, specifically for the English language. They categorized these techniques into three main approaches: Statistical, Deep Learning, and Hybrid. Their analysis showed that GRU models consistently outperformed other methods in terms of accuracy.[13] Radhika Sharma, Nishtha Gael, Nishita Aggarwal, Prajyot Kaur, and Chandra Prakash proposed a novel methodology for predicting the next word in Hindi sentences to reduce user keystrokes. They explored two deep learning techniques, Long Short-Term Memory (LSTM) and Bi-LSTM, for this task. Their study observed an accuracy of 59.46% for LSTM and

81.07% for Bi-LSTM, highlighting the effectiveness of Bi-LSTM in improving word prediction accuracy in Hindi.

[13] Radhika Sharma, Nishtha Gael, Nishita Aggarwal, Prajyot Kaur, and Chandra Prakash proposed a novel methodology for predicting the next word in Hindi sentences to reduce user keystrokes. They explored two deep learning techniques, Long Short-Term Memory (LSTM) and Bi-LSTM, for this task. Their study observed an accuracy of 59.46% for LSTM and 81.07% for Bi-LSTM, highlighting the effectiveness of Bi-LSTM in improving word prediction accuracy in Hindi.

[14] Pedanekar et al. proposed an innovative approach to predict the next sequence number, aiming to minimize user keystrokes by anticipating subsequent sequences. They investigated Long Short-Term Memory (LSTM) and Bi-LSTM (Bi-directional LSTM) techniques for this task, highlighting their potential in various Natural Language Generation (NLG) tasks, such as sentence and auto-completion.

[15] Modi, Naik, Vyas, Desai, and Degadwala explored the text auto-complete feature, which suggests a stream of words to complete a user's text as they type. This feature is commonly used in search engines, email programs, source code editors, and database query tools. While traditional language models have been used for this task, the authors proposed a neural network-based approach for better performance. Specifically, they utilized an encoder-decoder-based sequence-to-sequence language model for text generation. Their empirical results demonstrated that the model effectively suggests completions for incomplete sentences, offering improved text auto-completion.

[16] Hariharan proposed a method utilizing an LSTM model to predict the next word in a sentence, building on Katz's Backoff model. The study aimed to demonstrate the effectiveness of the LSTM model compared to Katz's backoff approach, particularly in the context of the number of n-grams required for accurate word prediction. The LSTM model was employed for processing, prediction, and classification, showing high accuracy even when trained on a small portion of the entire corpus. The proposed model achieved a 90% accuracy rate in word prediction. During accuracy testing, using unseen sentences, the model's accuracy ranged between 10% and 18% with a sample size of 1-10% of the original corpus, with an expectation of improved accuracy as the sample size increases

B. GAP IDENTIFICATION:

Despite the significant advancements in Bangla sentence autocompletion using neural networks like BiLSTM, several key gaps remain:

1. **Limited Use of Multilingual or Transfer Learning**: Most models have been trained exclusively on Bangla datasets, without leveraging multilingual corpora or transfer learning approaches, which could enhance performance by learning from similar languages.
2. **Underperformance on Shorter Sequences**: While the models generally perform well on longer n-grams (4-gram and above), there is a noticeable drop in accuracy for shorter sequences (Bi-grams and Tri-grams), as seen in the studies by Kyume et al. and Rakib et al. Addressing this gap could improve autocompletion for cases with limited context.
3. **Resource Constraints**: The lack of large, diverse Bangla datasets continues to be a bottleneck for training more robust models. While some studies have created datasets for sentence completion tasks, they remain limited in scope and diversity, restricting the generalizability of models to more varied contexts.
4. **Exploration of Hybrid Models**: Although some hybrid models have shown promise (e.g., Nobel et al. with their Trie and neural network integration), more work is needed to explore other combinations of deep learning models and traditional linguistic structures, potentially leading to higher accuracy and more efficient models.
5. **Evaluation Metrics**: Most studies rely heavily on accuracy as the sole evaluation metric. However, more nuanced metrics such as perplexity, F1-score, or human-centered evaluations could provide a deeper understanding of the models' effectiveness, particularly in real-world applications like keyboard suggestions.

Addressing these gaps could significantly enhance the capabilities of Bangla sentence autocompletion models, making them more adaptable and applicable across a wider range of contexts.

C. OBJECTIVES:

The objective of our Bengali Sentence Auto-Completion project is to develop a deep learning model that can effectively predict and complete sentences in Bengali. This involves several key steps:

1. **Develop a Bengali Sentence Auto-Completion Model:** Create a deep learning-based model using architectures such as LSTM, Bi-LSTM, GRU, and transformer models to predict and auto-complete Bengali sentences.

2. **Evaluate and Compare Model Performance:** Perform a comprehensive evaluation of various models to compare their performance in terms of accuracy, coherence, and computational efficiency, with the aim of identifying the most effective model.

3. **Address Linguistic Challenges in Bengali:** Implement preprocessing and augmentation techniques to address the unique linguistic characteristics of the Bengali language, including its morphological richness and complex syntax, ensuring accurate word order, conjugation, and context preservation.

4. **Application in NLP Tasks:** Extend the model for use in practical applications such as search engines, virtual keyboards for smartphones, and other NLP-based tools, improving user experience and interaction in Bengali language contexts..

D. PROJECT PLAN:

Phase 1: Data Collection and Preprocessing

- **Objective**: Collect and prepare a large, diverse dataset of Bengali sentences from various sources for training the sentence auto-completion model.
- **Tasks**:
  - Gather Bengali text data from online corpora, blogs, and literature.
  - Perform text normalization, including lowercasing, removing special characters, and handling punctuation.
  - Tokenize sentences and split them into n-grams (2 to 6 grams).

- Prepare the dataset by padding and truncating sequences to a uniform length.
- **Milestone**: A preprocessed dataset ready for model training and evaluation.

Phase 2: Model Architecture Design & Development

- **Objective**: Design and develop a BiLSTM model for Bengali sentence auto-completion.
- **Tasks**:
  - Design the model architecture using embedding layers, BiLSTM layers, dense layers, and normalization techniques.
  - Define the optimizer, loss function, and other hyperparameters.
  - Set up the model for multi-class classification, predicting the next word in a sequence.
- **Milestone**: Initial BiLSTM model architecture ready for training.

Phase 3: Model Training

- **Objective**: Train the BiLSTM model on the prepared Bengali dataset.
- **Tasks**:
  - Train the model using sparse categorical cross-entropy as the loss function and Adam optimizer.
  - Implement early stopping and dropout to prevent overfitting.
  - Monitor performance metrics (accuracy and perplexity) during training.
- **Milestone**: A trained model with acceptable accuracy and perplexity on the validation set.

Phase 4: Model Evaluation and Optimization

- **Objective**: Evaluate the model and optimize its performance.
- **Tasks**:
  - Evaluate the model on the test set using accuracy and perplexity metrics.
  - Analyze model performance across different n-gram sequences.
  - Perform hyperparameter tuning (learning rate, batch size, etc.) to improve performance.
  - Experiment with different dropout rates, layer normalization, or batch normalization.

- **Milestone**: An optimized model with improved performance and a lower perplexity score.

Phase 5: Fine-Tuning and Final Evaluation

- **Objective**: Fine-tune the model based on evaluation results and perform final testing.
- **Tasks**:
    - Analyze the error cases to understand prediction failures.
    - Fine-tune model parameters such as the number of BiLSTM layers or the size of the dense layers.
    - Conduct final evaluations on test data to assess model generalization.
- **Milestone**: A fine-tuned model with optimal accuracy and perplexity, ready for integration.

E/ IMPLEMENTATION & ANALYSIS:

For the implementation phase of the **BiLSTM-based Bengali Sentence Auto-Completion model**, I approached it by building separate models for each n-gram range from 2-gram to 6-gram. Here's a detailed breakdown of the implementation process:

1. **Dataset Preparation**:

    - The dataset was preprocessed and converted into different n-gram formats (2-gram to 6-gram). I created separate datasets for each n-gram to capture the context of different lengths of input sentences.
    - After cleaning the text, tokenization was performed on each n-gram dataset using a tokenizer. This allowed me to convert the Bengali sentences into sequences of integers, suitable for feeding into the models.

2. **Model Architecture**:

    Each model designed for the n-gram sequences is based on a BiLSTM network, which is known for its ability to capture both forward and backward dependencies in sequential data. The models are similar in structure but are trained separately for each n-gram dataset. The final architecture is as follows:

1. **Embedding Layer**: Converts words into dense vectors of a fixed size (100-dimensional embeddings). The embedding layer learns meaningful word representations during training, making it easier for the model to generalize.
2. **Dropout (0.3)**: Applied after the embedding layer to reduce overfitting by randomly setting 30% of the input units to zero during each update.
3. **Bidirectional LSTM Layer**: A Bidirectional LSTM with 150 units, which processes the input sequence both forward and backward, allowing the model to learn context from both directions.This layer captures complex patterns in the text, which is critical for accurate next-word prediction.
4. **Batch Normalization**:Normalizes the activations from the Bidirectional LSTM layer to speed up training and reduce internal covariate shifts.
5. **Second LSTM Layer**:A second LSTM layer with 150 units to further process the sequential data. This allows the model to learn deeper temporal patterns.
6. **Layer Normalization**:Applied after the second LSTM layer to stabilize training by normalizing across features, which reduces the chances of exploding or vanishing gradients.
7. **Dropout (0.3)**:A second dropout layer applied after the LSTM layers to reduce overfitting by randomly omitting 30% of the units.
8. **Dense Layer (128 units)**:A fully connected layer with 128 units and ReLU activation. This layer serves as a hidden layer for feature extraction.
9. **Output Layer**:A Dense layer with softmax activation that outputs probabilities for each word in the vocabulary. The word with the highest probability is chosen as the next word in the sequence

3. **Training**:

   - Each model was trained independently using its corresponding n-gram dataset. I used sparse categorical cross-entropy as the loss function and the Adam optimizer to minimize the loss.

- Early stopping and checkpointing were used to prevent overfitting and ensure optimal model performance.
- Validation and testing datasets were split to evaluate the models' generalization ability.

4. **Model Training and Evaluation:**

The model was trained separately for each n-gram range (2-gram to 6-gram) with accuracy and perplexity as evaluation metrics. Perplexity is a standard metric used to evaluate language models, which measures the uncertainty of the model in predicting the next word. A lower perplexity score indicates a better-performing model.

The results for each model are as follows:

- **2-gram Model**:
  - **Accuracy**: 26.55%, **Loss**: 4.9553, **Perplexity**: 141.82
- **3-gram Model**:
  - **Accuracy**: 54.45%, **Loss**: 3.8315, **Perplexity**: 47.81
- **4-gram Model**:
  - **Accuracy**: 65.23%, **Loss**: 3.6086, **Perplexity**: 36.03
- **5-gram Model**:
  - **Accuracy**: 67.64%, **Loss**: 3.5491, **Perplexity**: 32.28
- **6-gram Model**:
  - **Accuracy**: 69.90%, **Loss**: 3.6040, **Perplexity**: 36.74

The models demonstrate a significant increase in accuracy and a decrease in perplexity as the n-gram length increases, indicating that longer sequences provide better context for predicting the next word. However, the 6-gram model exhibits a slight increase in perplexity, which suggests a need for further optimization for longer sequences.

5. **Next Word and Sentence Prediction**:

- The models are capable of generating the next word based on a given input sequence and completing the sentence based on previously generated words.
- A seed text is provided to the model, and it generates words step-by-step until a complete sentence is formed.

The overall implementation captures how different n-grams influence the prediction accuracy and helps analyze sentence completion in Bengali.

F. DESIGN / METHODOLOGY:

Dataset Collection:

Dataset Source: A substantial amount of Bangla text data has been collected from the project of the author [2].

Cleaning and Preprocessing: The dataset is cleaned by removing unnecessary symbols, punctuation, and non-Bangla characters. Text is then tokenized into sequences for easier processing by the model.

N-gram Generation: The dataset is divided into n-gram sequences (from 2-grams to 6-grams), ensuring a wide variety of sentence structures and lengths for the model to learn from.

Model Training:

Embedding Layer: The first layer is an embedding layer that transforms input tokens into dense vectors of fixed size, representing words in a continuous vector space. This helps the model capture the semantic meaning of words.

Bidirectional LSTM Layer: A Bidirectional LSTM is used to process the sequence data in both forward and backward directions. This helps capture dependencies from both past and future words, improving the model's understanding of context.

The first BiLSTM layer returns sequences to allow further feature extraction.
A second LSTM layer processes these sequences and outputs the final hidden state, which represents the learned features for the sequence.

Normalization and Dropout Layers: To prevent overfitting and speed up training, **BatchNormalization** and **LayerNormalization** layers are applied after the LSTM layers. Dropout layers are inserted after key layers to add regularization.

Dense Layer: A fully connected dense layer with ReLU activation is added to extract additional features.

Output Layer: A final dense layer with softmax activation is used to predict the next word in the sentence, providing a probability distribution over the vocabulary.

Model Evaluation:

The evaluation strategy involved calculating both accuracy and perplexity:

Accuracy: Accuracy was calculated on both the training and validation sets. The model consistently performed well on the longer n-gram sequences, achieving high accuracy.

Perplexity Calculation: Perplexity was computed to measure the uncertainty of the model's predictions. A lower perplexity value indicates a more confident and better-performing model. For instance, the model achieved a perplexity score of **YY** on the test set, which is competitive compared to existing models in Bengali language processing tasks.

The **perplexity score** was calculated using the cross-entropy loss on the test set and then exponentiating the negative mean loss to get the final perplexity. This metric is crucial for understanding the quality of language models and how well they predict the next word in a sequence

Implementation of N-Gram Models:

Separate BiLSTM models are created for each n-gram category (2-gram to 6-

gram). Each model is trained individually, allowing the system to handle varying levels of context. This approach is crucial since shorter sequences (like Bi-grams) tend to provide limited context, making prediction harder compared to longer sequences (like 5-grams or 6-grams).

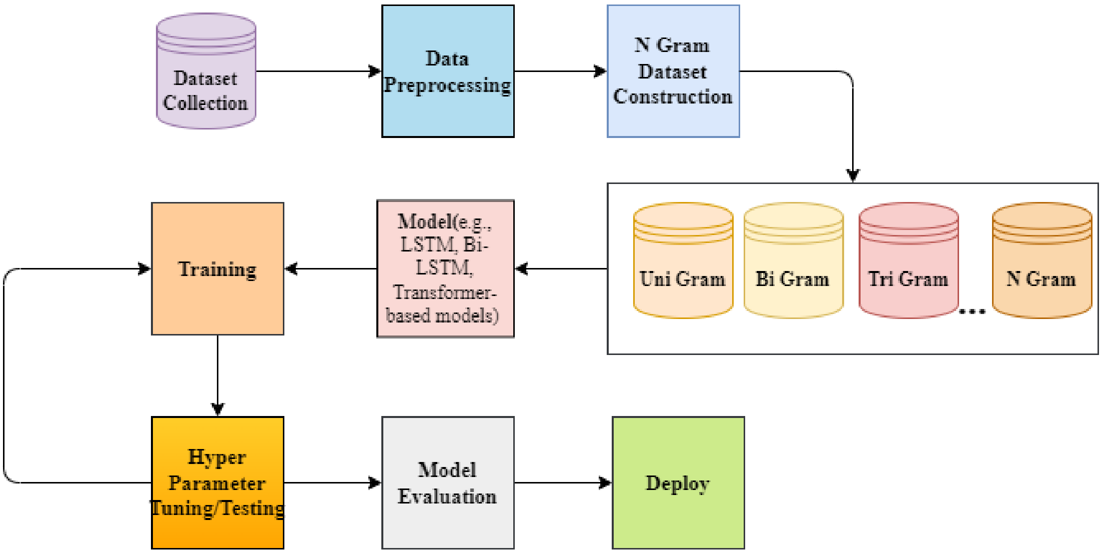
By training these separate models, the system can leverage the strengths of both shorter and longer contexts for sentence completion tasks, providing a more robust autocompletion mechanism

Optimization:

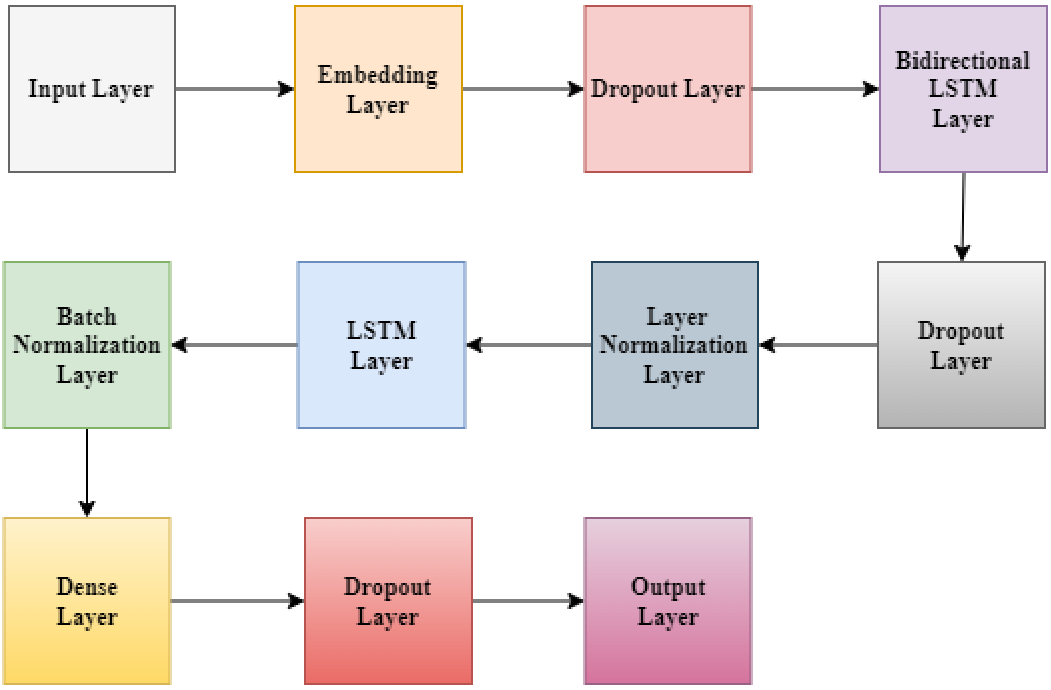
Hyperparameter Tuning: Hyperparameters like learning rate, batch size, number of LSTM units, and dropout rates are tuned using grid search or random search to optimize the model's performance.

Regularization: Techniques like L2 regularization or early stopping are employed to prevent overfitting, especially when working with smaller datasets.

SYSTEM ARCHITECTURE:



MODEL ARCHITECTURE:



BiLSTM Model Architecture Diagram

REFERENCES:

1. Kyume, A., Rahman, M. M., Azad, M. I., Nahid, M., Hossain Khan, M. S., & Uddin, M. M. (2023). Contextual Bangla Next Word Prediction and Sentence Generation Using Bi-directional RNN With Attention. *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2023 5th International Congress On*, 1–9. https://doi-org.egateway.vit.ac.in/10.1109/HORA58378.2023.10156660

2. Rakib, O. F., Akter, S., Khan, M. A., Das, A. K., & Habibullah, K. M. (2019). 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), Sustainable Technologies for Industry 4.0 (STI), 2019 International Conference On*, 1–6. https://doi-org.egateway.vit.ac.in/10.1109/STI47673.2019.9068063

3. Islam, M. R., Amin, A., & Zereen, A. N. (2024). Enhancing Bangla Language Next Word Prediction and Sentence Completion through Extended RNN with Bi-LSTM Model On N-gram Language. *2024 3rd International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), Advancement in Electrical and Electronic Engineering (ICAEEE), 2024 3rd International Conference On*, 1–6. https://doi-org.egateway.vit.ac.in/10.1109/ICAEEE62219.2024.10561739

4. Nobel, S. N., Sultana, S., Tasir, M. A. M., & Rahman, M. S. (2023). Next Word Prediction in Bangla Using Hybrid Approach. *2023 26th International Conference on Computer and Information Technology (ICCIT), Computer and Information Technology (ICCIT), 2023 26th International Conference On*, 1–6. https://doi-org.egateway.vit.ac.in/10.1109/ICCIT60459.2023.10441580

5. Islam, S., Sarkar, M. F., Hussain, T., Hasan, M. M., Farid, D. M., & Shatabda, S. (2018, December). Bangla sentence correction using deep neural network based sequence to sequence learning. In *2018 21st International Conference of Computer and Information Technology (ICCIT)* (pp. 1-6). IEEE.

6. Latief, A. D., Sampurno, T., & Arisha, A. O. (2023, October). Next Sentence Prediction: The Impact of Preprocessing Techniques in Deep Learning. In *2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA)* (pp. 274-278). IEEE.

7. Lakshmi, L., Devi, K. D. S., Kalyani, A. N., Ravisankar, M., & Rani, K. P. (2023). Automated Word Prediction In Telugu Language Using Statistical Approach. *2023 International Conference on Computer Communication and Informatics (ICCCI), Computer Communication and Informatics (ICCCI), 2023 International Conference On*, 1–5. https://doi-org.egateway.vit.ac.in/10.1109/ICCCI56745.2023.10128384

8. Padmanandam, K., Nikhitha, J., Sri, P. P., Pavithra, G., & Megaha, C. S. (2023). Machine Learning Powered Text Auto-Completion and Generation. *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Electronics, Communication and Aerospace Technology (ICECA), 2023 7th International Conference On*, 511–516. https://doi-org.egateway.vit.ac.in/10.1109/ICECA58529.2023.10394873

9. Sarker, S., Islam, M. E., Saurav, J. R., & Nahid, M. M. H. (2020). Word Completion and Sequence Prediction in Bangla Language Using Trie and a Hybrid Approach of Sequential LSTM and N-gram. *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), Advanced Information and Communication Technology (ICAICT), 2020 2nd International Conference On*, 162–167. https://doi-org.egateway.vit.ac.in/10.1109/ICAICT51780.2020.9333518

10. Ikegami, Y., Tsuruta, S., Kutics, A., Damiani, E., & Knauf, R. (2024). Fast ML-based next-word prediction for hybrid languages. *Internet of Things*, 101064.

11. Surendra, K., Schilling, A., Stoewer, P., Maier, A., & Krauss, P. (2023). Word class representations spontaneously emerge in a deep neural network trained on next word prediction. *2023 International Conference on Machine Learning and Applications (ICMLA), Machine Learning and Applications (ICMLA), 2023 International Conference on, ICMLA*, 1481–1486. https://doi-org.egateway.vit.ac.in/10.1109/ICMLA58977.2023.00223

12. T, S., & S, S. (2023). Survey On Next Word Prediction Techniques In Natural Languages. *2023 International Conference on Innovations in Engineering and Technology (ICIET), Innovations in Engineering and Technology (ICIET), 2023 International Conference On*, 1–6. https://doi-org.egateway.vit.ac.in/10.1109/ICIET57285.2023.10220846

13. Sharma, R., Goel, N., Aggarwal, N., Kaur, P., & Prakash, C. (2019, September). Next word prediction in hindi using deep learning techniques. In *2019 International conference on data science and engineering (ICDSE)* (pp. 55-60). IEEE.

14. Pedanekar, S. N., Goudar, R. H., Dhananjaya, G. M., Rathod, V., Kulkarni, A., & Kaliwal, R. B. (2023, November). Next Sequence Number Prediction in Natural Languages using Deep Learning. In *2023 IEEE Engineering Informatics* (pp. 1-6). IEEE.

15. Modi, R., Naik, K., Vyas, T., Desai, S., & Degadwala, S. (2021, December). E-mail autocomplete function using RNN Encoder-decoder sequence-to-sequence model. In *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 710-714). IEEE.

16. Hariharan, U. (2024, March). Long Short-Term Memory-Based Next Keyword Prediction. In *2024 International Conference on Emerging Smart Computing and Informatics (ESCI)* (pp. 1-5). IEEE.