2025 International Conference on Electrical, Computer and Communication Engineering (ECCE), 13-15 February 2025, CUET, Chattogram-4349, Bangladesh

2025 International Conference on Electrical, Computer and Communication Engineering (ECCE) | 979-8-3503-5750-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ECCE64574.2025.11013841

Advancing Low-Resource NLP: Contextual Question Answering for Bengali Language Using Llama

Koshik Debanath[1], Sagor Aich[2], Azmain Yakin Srizon[3]
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh
Email: [1]koshik.debanath@gmail.com, [2]sagor.aichh@gmail.com, [3]azmainsrizon@gmail.com

***Abstract*—Natural language processing (NLP) has witnessed significant advancements in recent years, particularly in improving question-answering (QA) systems for well-resourced languages such as English. However, the development of such systems for low-resource languages, including Bengali, remains insufficiently explored. This study proposes an approach to developing a Bengali QA system utilizing the Llama-3.2-3B-Instruct model, leveraging transfer learning techniques on a synthetic dataset derived from the SQuAD 2.0 benchmark. The experiments achieved an F1 score of 42.77%, marking a 4.02% improvement over the previous best performance of multilingual BERT (mBERT) variants. These results establish a benchmark against human responses and underscore the potential of transfer learning in advancing QA capabilities for Bengali and similar low-resource languages.**

***Keywords*—Natural Language Processing, Question Answering, Large Language Models, Llama Model, Fine-Tuning, Bengali Dataset.**

I. Introduction

Question Answering (QA) involves the development of systems designed to automatically respond to human queries in natural language. Text-based QA tasks can be considered information retrieval challenges, aiming to identify relevant documents, extract potential answers, and rank them by relevance. These tasks often focus on reading comprehension, with the objective of locating the exact answer, commonly referred to as a "span," within a given passage. While QA tasks span various modalities—including visual contexts (e.g., images), open-domain queries, and multimodal inputs integrating images, videos, audio, and text alongside common-sense reasoning—this study focuses specifically on text-based reading comprehension.

Despite significant progress in developing QA systems for high-resource languages such as English, advancements for Bengali—a language spoken by over 300 million people—remain limited due to the lack of comprehensive datasets and pre-trained models tailored for Bengali [1]. The absence of large-scale QA datasets and the limited availability of skilled annotators have hindered the creation of high-quality reading comprehension datasets for Bengali [2].

Previous works, such as [3], have employed the multilingual BERT model for zero-shot transfer learning and fine-tuning using synthetic training datasets for Bengali reading comprehension tasks. Additionally, other BERT model variants, including RoBERTa [4] and DistilBERT [5], have been explored in both zero-shot and fine-tuned settings. To train and evaluate their models, a substantial portion of the SQuAD 2.0 dataset [6] was translated from English to Bengali, employing fuzzy matching techniques to maintain answer quality. Furthermore, a human-annotated Bengali reading comprehension dataset was developed using popular Bangla Wikipedia articles to evaluate these models.

In a related study, [7] introduced PAL-BERT, a first-order pruning model built upon the ALBERT model [8], tailored to the characteristics of QA systems and language models.

Building on these developments, this paper employs the Llama-3.2-3B model to develop a Bengali QA system using a synthetic dataset derived from the SQuAD 2.0 benchmark [6]. By leveraging transfer learning techniques, this state-of-the-art transformer model is adapted for Bengali reading comprehension tasks. Transfer learning facilitates the use of pre-trained models trained on extensive linguistic corpora and enables their adaptation for specific tasks without requiring vast labeled data [9].

The contributions of this work include establishing a benchmark for Bengali QA systems using the Llama-3.2-3B model and evaluating its performance against human responses. This research aims to enhance accessibility and usability for diverse user groups, including students and individuals with learning difficulties [10]. The proposed system saves time and ensures accessibility, benefiting groups such as children and adults seeking precise answers from literature, which may otherwise require extensive review. By demonstrating the feasibility of the Llama-3.2-3B model for Bengali reading comprehension, this study aims to pave the way for further advancements in NLP applications for low-resource languages.

II. Literature Review

Bengali, as one of the most widely spoken languages globally, represents a notable gap in the current NLP landscape. The introduction of models such as Llama-3.2-3B-Instruct has revitalized efforts to address low-resource languages. These models, when fine-tuned on synthetic datasets derived from resources like SQuAD 2.0, exhibit

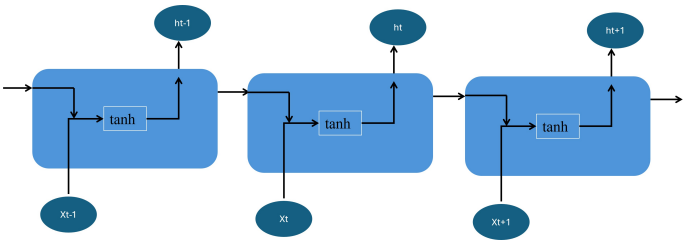

Figure 1. Architecture of a Recurrent Neural Network (RNN)

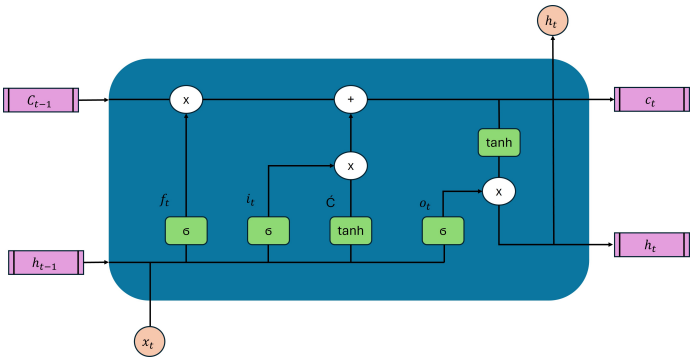

Figure 2. Architecture of a Long Short-Term Memory (LSTM) Network

significant improvements over multilingual baselines. By applying transfer learning to such datasets, it has been observed that the Llama-3.2-3B-Instruct model can outperform BERT variants.

A. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) were among the initial neural architectures developed for processing sequential data, particularly in natural language processing (NLP) tasks [11]. RNNs utilize feedback loops to maintain a hidden state that encapsulates information from previous inputs, making them well-suited for applications such as language modeling and sequence prediction [12]. However, RNNs face inherent challenges, such as vanishing and exploding gradients, which limit their capacity to learn long-term dependencies effectively [13]. Despite these limitations, RNNs laid the foundation for subsequent advancements in deep learning models.

The architecture of a basic Recurrent Neural Network (RNN) is depicted in Figure 1. In this model, each hidden state h_t is computed based on the previous hidden state h_{t-1} and the current input x_t. The mathematical expression for updating the hidden state h_t is as follows:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b)$$

Here, W_h and W_x denote the weight matrices, while b represents the bias term. Although RNNs are conceptually straightforward, they struggle to retain information over long sequences due to challenges related to gradient propagation. This limitation has been effectively addressed by Long Short-Term Memory networks (LSTMs), which are explicitly designed to overcome these difficulties.

B. Long Short-Term Memory

Long Short-Term Memory (LSTM) networks were introduced to address the limitations of traditional RNNs [14]. LSTMs incorporate memory cells and gating mechanisms that enable them to retain information over extended periods, effectively mitigating the vanishing gradient problem. This architecture has been widely adopted in various NLP applications, including machine translation and sentiment analysis [15]. LSTMs have demonstrated superior performance compared to standard RNNs in capturing contextual relationships within text data [16], making them a preferred choice for sequential modeling tasks. In [17], a comprehensive tutorial is provided on the fundamental concepts of RNNs and LSTMs.

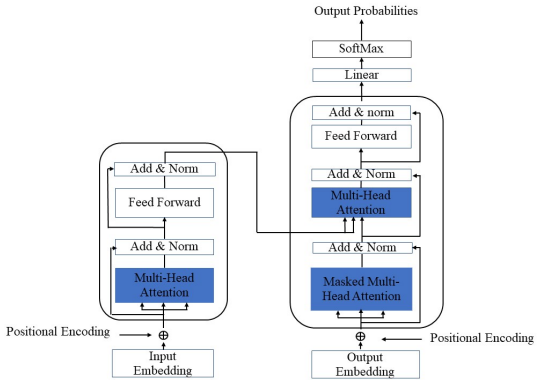

Figure 3. Architecture of the Transformer Model

1) Forget Gate: The forget gate (f_t) determines which information from the previous cell state should be discarded. This is mathematically represented as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2) Input Gate: The input gate (i_t) decides which new information should be stored in the cell state. Its formulation is:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

3) Cell State Update: The cell state update ($\tilde{C}_t$) generates new candidate values to be added to the cell state, defined as:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

4) Output Gate: The output gate (o_t) controls the output at each time step, computed as:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

5) Cell State Update and Final Output: The final cell state (C_t) is updated using the forget and input gates:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

The final hidden state (output) of the LSTM (h_t) is calculated using the output gate and the updated cell state:

$$h_t = o_t * \tanh(C_t)$$

C. Transformer

The Transformer architecture revolutionized NLP by introducing self-attention mechanisms that assess the importance of words in a sequence without relying on recurrence [18]. This innovation allows for parallel processing of sequences, improving efficiency and performance. Transformers form the foundation of state-of-the-art models like BERT and GPT, enabling robust QA systems capable of handling complex queries across languages.

Transformers have two main components: an encoder, which generates contextual representations from the input, and a decoder, which uses these representations to produce the output.

The self-attention mechanism, a core feature, calculates attention scores to weigh word relationships in the sequence. Multi-head attention enhances this by enabling the model to learn diverse aspects of the input, while positional encodings help retain word order information.

Each encoder and decoder layer also includes feed-forward networks for learning complex patterns and uses layer normalization and residual connections to stabilize training and ensure efficient gradient flow.

As illustrated in Figure 3, the architecture's layered design exemplifies its groundbreaking approach to NLP applications.

D. Large Language Models

Large Language Models (LLMs) have revolutionized natural language processing (NLP), enabling machines to comprehend and generate human-like text. Meta AI's Llama series exemplifies state-of-the-art LLMs, prioritizing efficiency and leveraging transformer architectures with self-attention mechanisms to process input data and capture long-range dependencies effectively.

LLMs are applied in domains like chatbots and educational tools, showcasing adaptability to various languages and contexts, especially in low-resource settings [19]. Models such as Llama-13B surpass GPT-3 (175B) on many benchmarks, while Llama-65B rivals top models like Chinchilla-70B and PaLM-540B [4].

Llama 2, ranging from 7 to 70 billion parameters, includes fine-tuned versions like Llama 2-Chat, optimized for conversational tasks. These models outperform most open-source alternatives across benchmarks and are viable competitors to closed-source models [5].

III. Proposed Methodology

This section details the process of building and fine-tuning the Bengali question answering (QA) system using the Llama-3.2-3B-Instruct model. The methodology comprises three main components: Data Preprocessing, Model Architecture, and Training Process.

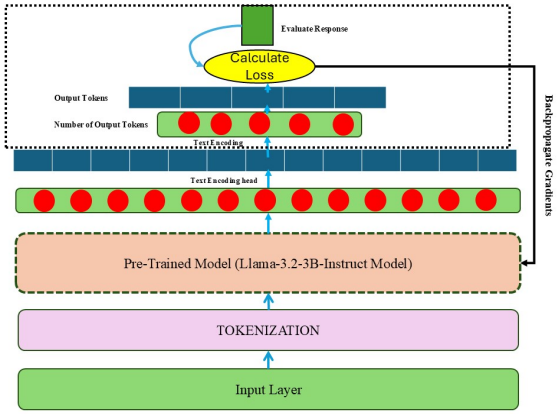

Figure 4. Architecture of the Proposed Bengali Question Answering System

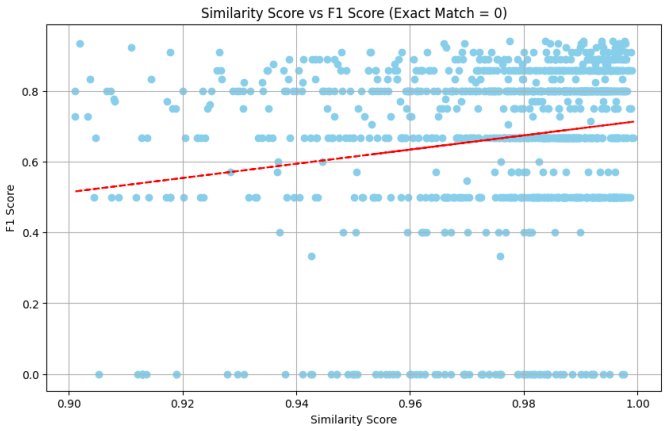

Figure 5. Relationship between Similarity Score and F1 Score (Exact Match = 0), with a trend line indicating a positive correlation.

A. Data Preprocessing

The dataset used for this study combines a synthetic Bengali QA dataset derived from the SQuAD 2.0 benchmark with a human-annotated QA dataset sourced from Bengali Wikipedia. The preprocessing pipeline began with tokenization using the tokenizer provided with the model, followed by text normalization to eliminate special characters, punctuation, and redundant spaces. The context, questions, and answers were reformatted into a conversational structure compatible with the model's input requirements. The Hugging Face tokenizer was employed to prepare the dataset for the transformer model.

The dataset was divided into training and validation subsets, ensuring that question-answer pairs were carefully aligned to preserve contextual integrity. Additional preprocessing steps included the removal of stop words and the application of lemmatization to the context and question fields to ensure consistent word forms.

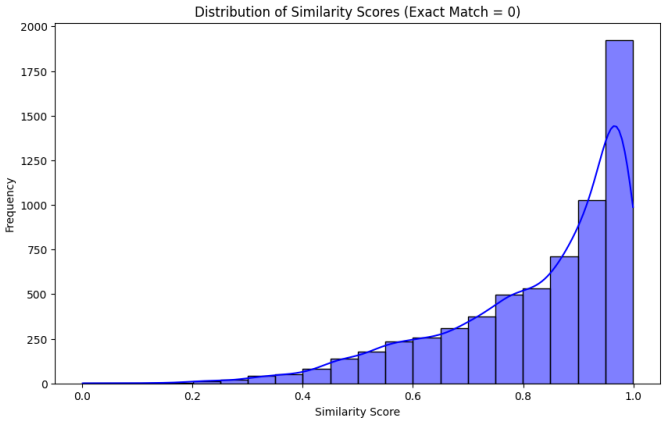

Figure 6. Distribution of Similarity Scores for Queries with No Exact Match (exact_match = 0).

B. Model Architecture

The proposed model adopts a transformer-based sequence-to-sequence architecture. The Llama-3.2-3B-Instruct model, which leverages a transformer encoder-decoder mechanism, was used. RoPE scaling was implemented to handle long sequences, while the LoRA (Low-Rank Adaptation) method was applied to fine-tune specific layers using low-rank updates, reducing computational overhead.

The architecture comprises several critical components. The tokenization and embedding layer converts input sequences into embeddings that the transformer model can process. The transformer encoder-decoder employs multiple layers of self-attention and feed-forward networks to encode the input context and generate a response. The prediction layer produces the final answer to the question based on the context using learned representations.

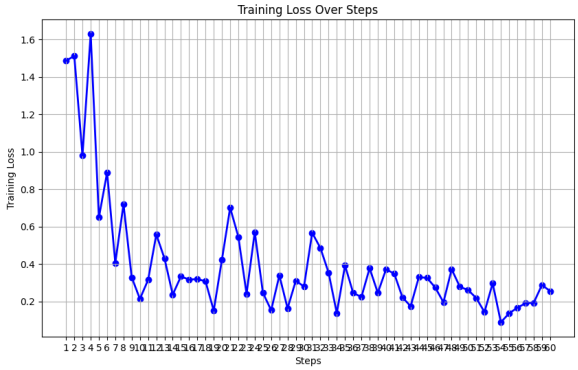

Figure 7. Training Loss Progression Across Steps

C. Training Process

The training process involved fine-tuning the Llama-3.2-3B-Instruct model with the preprocessed Bengali QA dataset using the LoRA fine-tuning technique. Memory optimization techniques, such as gradient checkpointing and 4-bit quantization, were employed to enhance computational efficiency. The model was trained for 60 steps using a learning rate of 2×10^{-4}, a batch size of 2, and a weight decay of 0.01.

The training began with loading the pre-trained Llama-3.2-3B-Instruct model and applying the LoRA technique to achieve parameter-efficient training. The model was fine-tuned with the Bengali QA dataset using the predefined hyperparameters. The AdamW optimizer, along with a linear learning rate scheduler incorporating warm-up steps, was utilized to enhance training stability. Loss values were logged during the training process, and the model's performance was periodically evaluated on the validation set to ensure alignment with the desired objectives.

D. Architecture Overview

This proposed architecture combines the efficiency of LoRA with the capabilities of the Llama-3.2-3B-Instruct model, creating a resource-efficient Bengali question-answering system with minimal computational overhead.

1) Input Layer (Tokenization): The input layer processes the user-provided context and question using a tokenizer designed for the pre-trained model. It segments the input into tokens and maps them to embeddings, enabling the model to interpret and process the data efficiently.

2) Pre-trained Language Model (Llama-3.2-3B-Instruct): The Llama-3.2-3B-Instruct model forms the core of the system. It uses transformer-based architecture with 4-bit quantization for reduced memory usage and RoPE scaling for handling longer input sequences, ensuring efficient and accurate question-answering.

3) LoRA Layers (Low-Rank Adaptation): LoRA efficiently fine-tunes the model by introducing trainable low-rank matrices into key projection layers (*q_proj*, *k_proj*, *v_proj*). Configurations include a rank (r) of 16, alpha value of 32, and no dropout, balancing resource efficiency and performance.

4) Training Loop (Supervised Fine-Tuning - SFT): Supervised fine-tuning adapts the model for question-answering tasks. The SFTTrainer uses the AdamW 8-bit optimizer, a batch size of 2, and 60 training steps to update the model's weights effectively, ensuring accurate and context-aware responses.

5) Output (Chat-based Template): The system generates outputs in a structured conversation format, comprising the system-provided context, the user's question, and the assistant's generated answer. This ensures clarity and usability for end-users.

E. Evaluation Metrics

The model's performance was evaluated using four metrics:

Table I
QUERIES WITH ZERO EXACT MATCH BUT HIGH SIMILARITY AND SAS SCORES, DEMONSTRATING PRESERVED CONTEXTUAL MEANING.

Query	Ground Truth	Prediction	Sim. Score	SAS Score
কোন শহর কুপার নদীর সরাসরি পূর্ব দিকে জমি দখল করে?	মাউন্ট প্লিজেন্ট জ	মাউন্ট প্লিজেন্ট জমিটি	0.923928	0.908574
আমেরিকা বিপ্লবের সময় কোন জেনারেল চার্লস টাউন আক্রমণ করেছিলেন?	জেনারেল স্যার হেনরি ক্লিনটন স	স্যার হেনরি ক্লিনটন স	0.973075	0.941400
আমেরিকান বিপ্লবের বৃহত্তম আমেরিকান পরাজয় কি ছিল?	চার্লস টাউনকে অবরোধ দ	চার্লস টাউনকে অবরোধ দেওয়া যুদ্ধের	0.971253	0.923876
কোপার নদীর সাথে কোন নদী মিশে গিয়ে চার্লস্টন হারবার গঠন করে?	অ্যাশলে এবং ক	অ্যাশলে এ	0.943337	0.849364
চার্লস্টনের প্রথম অবস্থানটি কোথায় ছিল?	আলবেমারল পয়েন্ট ন	অ্যাসলে নদীর পশ্চিম তীরে আলবেমারল পয়েন্ট ন	0.876391	0.860536

Table II
COMPARISON OF ACTUAL AND PREDICTED RESPONSES (CORRECTLY PREDICTED ANSWERS)

Context	Question	Actual Response	Predicted Response
১৮২০ সালের মধ্যে, চার্লস্টনের জনসংখ্যা ২৩,০০০ হয়ে দাঁড়িয়েছিল, এটি তার কালো এবং বেশিরভাগ দাস সংখ্যাগরিষ্ঠতা বজায় রেখেছিল। ১৮২২ সালের মে মাসে ডেনমার্ক ভেসি নামে একটি মুক্ত কৃষ্ণাঙ্গ দাস বিদ্রোহ প্রকাশিত হলে, সাদারা তীব্র ভয়ের সাথে প্রতিক্রিয়া দেখায়, কারণ তারা হাইতিয়ান বিপ্লবের সময়ে সাদাদের বিরুদ্ধে দাসদের সহিংস প্রতিরোধ সম্পর্কে ভালই অবগত ছিল। এরপরেই, ভেসিকে বিচার করা হয়েছিল এবং মৃত্যুদণ্ড দেওয়া হয়েছিল, জুলাইয়ের প্রথমদিকে পাঁচজন দাসকে ফাঁসি দেওয়া হয়েছিল। আরও ২৮ জন দাসকে পরে ফাঁসি দেওয়া হয়েছিল। পরবর্তীতে, রাজ্য আইনসভায় ম্যানুয়েশন দাসকে মুক্ত করা এবং বিনামূল্যে কৃষ্ণাঙ্গ ও দাসদের ক্রিয়াকলাপ নিয়ন্ত্রণের জন্য পৃথক আইনসভার অনুমোদনের জন্য আইন পাস হয়।	কোন বিপ্লব শ্বেতীদের দাসের প্রতিশোধের ভয়ে ভীত করে তুলেছিল?	হাইতিয়ান বিপ্লবের	হাইতিয়ান বিপ্লবের
চার্লসটন আমেরিকা যুক্তরাষ্ট্রের দক্ষিণ ক্যারোলাইনা রাজ্যের প্রাচীনতম এবং দ্বিতীয় বৃহত্তম শহর, চার্লসটন কাউন্টির কাউন্টি আসন এবং চার্লস্টন নর্থ চার্লস্টন সামারভিলে মেট্রোপলিটন স্ট্যাটিস্টিকাল এরিয়ার প্রধান শহর। শহরটি দক্ষিণ ক্যারোলিনার উপকূলরেখার ভৌগলিক মিডপয়েন্টের ঠিক দক্ষিণে অবস্থিত এবং অ্যাশলে এবং কুপার নদীর সংগম দ্বারা গঠিত আটলান্টিক মহাসাগরের একটি খাঁটি চার্লস্টন হারবারে অবস্থিত, অথবা স্থানীয়ভাবে প্রকাশিত হয়েছে, "যেখানে কুপার এবং অ্যাশলে নদীগুলি একত্র হয়ে আটলান্টিক মহাসাগর গঠনে আসে।	চার্লস্টন হারবার কোন মহাসাগরের খাঁড়ি?	আটলান্টিক মহাসাগরের	আটলান্টিক মহাসাগরের

Table III
COMPARISON OF ACTUAL AND PREDICTED RESPONSES (INCORRECTLY PREDICTED ANSWERS)

Context	Question	Actual Response	Predicted Response
এই শহরে রঙের মুক্ত জনগণের একটি বৃহত শ্রেণি ছিল। ১৮৬০ সালের মধ্যে, ৩,৭৮৫ রঙের মুক্ত মানুষ চার্লসটনে ছিলেন, শহরের কৃষ্ণাঙ্গ জনসংখ্যার প্রায় ১৮% এবং মোট জনসংখ্যার ৮%। গোলামের চেয়ে বর্ণের মুক্ত মানুষ মিশ্র জাতিগত পটভূমির বেশি হওয়ার সম্ভাবনা বেশি ছিল। অনেকে শিক্ষিত, দক্ষ কারুশিল্পের অনুশীলন করেছিলেন, আবার কেউ কেউ দাসসহ পর্যাপ্ত সম্পত্তির মালিকও ছিলেন। ১৭৯০ সালে তারা পারস্পরিক সহায়তার জন্য ব্রাউন ফ্যালোশিপ সোসাইটি প্রতিষ্ঠা করেছিলেন, প্রাথমিকভাবে দাফন সমিতি হিসাবে। এটি ১৯৪৫ অবধি অব্যাহত ছিল।	ব্রাউন ফেলোশিপ সোসাইটি কোন সালে প্রতিষ্ঠিত হয়েছিল?	১৮৬০ স	১৭৯০ স
পশ্চিমে গভর্নর উইলিয়াম সায়লের নেতৃত্বে বারমুডা দক্ষিণ ক্যারোলিনার পূর্বদিকে অবস্থিত যদিও এটি দক্ষিণ ক্যারোলিনার পূর্ব দিকে অবস্থিত যদিও এটি ১,০৩০ কিমি বা ৪০৪০ মাইল দূরে অবস্থিত দ্বারা এই সম্প্রদায়টি প্রতিষ্ঠিত হয়েছিল। বর্তমান শহর কেন্দ্র থেকে কয়েক মাইল উত্তর-পশ্চিমে অ্যাশলে নদীর তীর। এটি শিগগিরই আর্ল অফ শ্যাফটসবারির দ্বারা ভবিষ্যদ্বাণী করা হয়েছিল, লর্ডস প্রোপ্রেইটারদের একজন, "গ্রেট বন্দর শহরে" পরিণত হবে, এই শহরটি দ্রুতই পূর্ণ হয়েছিল। ১৬৮০ সালে, বন্দোবস্তটি অ্যাশলে নদীর পূর্বদিকে অ্যাশলে এবং কুপার নদীর মধ্যে উপদ্বীপে স্থানান্তরিত করা হয়েছিল। এই অবস্থানটি কেবল আরও ডিফেন্সেবল ছিল না, তবে এটি একটি সূক্ষ্ম প্রাকৃতিক বন্দরে অ্যাক্সেসেরও প্রস্তাব করেছিল।	বন্দোবস্তটি পূর্বে কোন নদীতে স্থানান্তরিত হয়েছিল?	অ্যাশলে নদীর	কুপার নদীর

1) Exact Match (EM): Exact Match (EM) measures the percentage of predictions that exactly match the ground-truth answer. It is calculated as the total number of exact matches divided by the total number of predictions, where $F(x_i) = 1$ if the predicted answer matches the correct answer, and 0 otherwise.

2) F1 Score: The F1 score, a less strict metric than EM, is the harmonic mean of precision and recall. It is computed as $F1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$, balancing the trade-off between these two metrics.

3) Cosine Similarity: Cosine similarity quantifies the similarity between two vectors by calculating the cosine of the angle between them. It is given by $\frac{\vec{A}\cdot\vec{B}}{\|\vec{A}\|\|\vec{B}\|}$, where $\vec{A}$ and $\vec{B}$ are the vectors, their dot product represents the numerator, and their magnitudes form the denominator. A value closer to 1 indicates higher similarity.

4) Semantic Answer Similarity (SAS) Score: The SAS score measures the semantic similarity between the predicted and reference answers. It is defined as $SAS(A,R) = \frac{\text{Sim}(A,R)}{1+\text{Dist}(A,R)}$, where A and R are the predicted and reference answers, $\text{Sim}(A,R)$ is a similarity function (e.g., cosine similarity), and $\text{Dist}(A,R)$ is a distance metric. Higher scores indicate better semantic alignment.

IV. EXPERIMENTAL ANALYSIS

The performance of the Bengali QA system was evaluated on a validation set comprising 7,488 question-answer pairs. Exact Match (EM) and F1 Score metrics were used to assess the model. The results showed an F1 score of 42.77% and an exact match score of 14.48%, highlighting reasonable performance despite the challenges of a low-resource language.

Figure 5 depicts the relationship between similarity scores and F1 scores for cases without exact matches

(`exact_match = 0`). The red dashed trend line suggests a slight positive correlation between these metrics.

Figure 6 shows a histogram of similarity scores for instances without exact matches. The x-axis represents similarity scores, while the y-axis indicates their frequency. The overlaid KDE (Kernel Density Estimate) curve highlights a concentration of similarity scores near 1, indicating that despite the absence of exact matches, most predictions closely align with the ground truth, demonstrating strong semantic consistency.

Table I showcases examples of queries where the exact match score is zero, yet the similarity and SAS scores remain high. This indicates that, despite not achieving exact string matches, the contextual meaning of the predictions aligns closely with the ground truth. For instance, minor formatting differences, such as numerals with or without suffixes, do not alter the semantic equivalence. These observations underscore the robustness of similarity and SAS scores in capturing contextual information.

Table IV
F1 Score Comparison of previous studies and Llama Models

Model	F1 Score (%)
BERT [3]	36.38
DistilBERT [3]	38.75
RoBERTa [3]	19.16
Llama 3.2-3B	**42.77**

Table IV compares the F1 scores of various models, highlighting the significant improvement achieved by our implementation. The F1 score of Llama 3.2-3B surpasses BERT and DistilBERT while being comparable to RoBERTa. Notably, Llama 3.2-3B is a lightweight model with fewer parameters, requiring only 3.4GB of GPU memory.

Figure 7 illustrates the training loss, reflecting the model's learning progression during training. Tables II and III provide examples of predictions made by the trained Bengali QA system. Table II lists correctly predicted answers, while Table III highlights incorrectly predicted ones.

V. Conclusion

This paper introduced a Bengali question answering system using the Llama-3.2-3B-Instruct model, leveraging transfer learning and LoRA techniques to address the low-resource nature of the Bengali language. By fine-tuning the model on a synthetic dataset derived from SQuAD 2.0, we achieved an F1 score of 42.77% and an Exact Match (EM) score of 14.22%, establishing a foundational benchmark for Bengali QA systems. The results demonstrate the effectiveness of large language models like Llama in improving QA performance for low-resource languages. Future directions include improving dataset quality, exploring alternative model architectures, and incorporating advanced techniques such as RLHF (Reinforcement Learning with Human Feedback). This approach lays a framework for developing NLP systems for underrepresented languages and can be adapted to other low-resource languages beyond Bengali.

References

[1] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] S. Ilić, E. Marrese-Taylor, J. A. Balazs, and Y. Matsuo, "Deep contextualized word representations for detecting sarcasm and irony," *arXiv preprint arXiv:1809.09795*, 2018.

[3] T. Tahsin Mayeesha, A. Md Sarwar, and R. M. Rahman, "Deep learning based question answering system in bengali," *Journal of Information and Telecommunication*, vol. 5, no. 2, pp. 145–178, 2021.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: a robustly optimized bert pretraining approach. corr 2019," *arXiv preprint arXiv:1907.11692*, 1907.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arxiv 2019," *arXiv preprint arXiv:1910.01108*, 2019.

[6] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *arXiv preprint arXiv:1806.03822*, 2018.

[7] W. Zheng, S. Lu, Z. Cai, R. Wang, L. Wang, and L. Yin, "Pal-bert: an improved question answering model," *Computer Modeling in Engineering & Sciences*, pp. 1–10, 2023.

[8] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," 2020.

[9] A. Conneau and G. Lample, "Cross-lingual language model pretraining," *Advances in neural information processing systems*, vol. 32, 2019.

[10] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

[11] L. Hirschman, M. Light, E. Breck, and J. Burger, "Deep read: a reading comprehension system. acl in: Proceedings of the 37th annual meeting of the association for computational linguistics (1999)."

[12] M. Richardson, C. J. Burges, and E. Renshaw, "Mctest: A challenge dataset for the open-domain machine comprehension of text," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 193–203, 2013.

[13] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *Advances in neural information processing systems*, vol. 28, 2015.

[14] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[15] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, Ieee, 2013.

[16] P. Rajpurkar, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[17] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need. advances in neural information processing systems," *Advances in neural information processing systems*, vol. 30, no. 2017, 2017.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.