
TOURISM QUESTION ANSWER SYSTEM IN INDIAN LANGUAGE USING DOMAIN-ADAPTED FOUNDATION MODELS

A PREPRINT

Praveen Gatla

Department of Linguistics
Banaras Hindu University
Varanasi
praveengatla@bhu.ac.in

Anushka

Department of Humanistic Studies
Indian Institute of Technology (BHU)
Varanasi
anushkasinha992@gmail.com

Nikita Kanwar *

Department of Computer Science and Engineering
Indian Institute of Technology
Bhilai
rathorenik2@gmail.com

Gouri Sahoo

Department of Linguistics
Banaras Hindu University
Varanasi
gourisahoo.cs@gmail.com

Rajesh Kumar Mundotiya

Department of Computer Science and Engineering
Indian Institute of Technology
Bhilai
rajeshkm.mundotiya@gmail.com

ABSTRACT

This article presents the first comprehensive study on designing a baseline extractive question answering (QA) system for the Hindi tourism domain, with a specialized focus on the Varanasi-a cultural and spiritual hub renowned for its Bhakti-Bhaav (devotional ethos). Targeting ten tourism-centric subdomains-Ganga Aarti, Cruise, Food Court, Public Toilet, Kund, Museum, General, Ashram, Temple and Travel, the work addresses the absence of language-specific QA resources in Hindi for culturally nuanced applications. In this paper, a dataset comprising 7,715 Hindi QA pairs pertaining to Varanasi tourism was constructed and subsequently augmented with 27,455 pairs generated via Llama zero-shot prompting. We propose a framework leveraging foundation models-BERT and RoBERTa, fine-tuned using Supervised Fine-Tuning (SFT) and Low-Rank Adaptation (LoRA), to optimize parameter efficiency and task performance. Multiple variants of BERT, including pre-trained languages (e.g., Hindi-BERT), are evaluated to assess their suitability for low-resource domain-specific QA. Evaluation metrics - F1, BLEU, and ROUGE-L - highlight trade-offs between answer precision and linguistic fluency. Experiments demonstrate that LoRA-based fine-tuning achieves competitive performance (85.3% F1) while reducing trainable parameters by 98% compared to SFT, striking a balance between efficiency and accuracy. Comparative analysis across models reveals that RoBERTa with SFT outperforms BERT variants in capturing contextual nuances, particularly for culturally embedded terms (e.g., Aarti, Kund). This work establishes a foundational baseline for Hindi tourism QA systems, emphasizing the role of LORA in low-resource settings and underscoring the need for culturally contextualized NLP frameworks in the tourism domain.

Keywords SFT · LORA · Tourism Domain · Hindi Language · QA System · BERT · RoBERTa

*The work has done during her internship at IIT Bhilai

1 Introduction

Natural Language Processing (NLP) is a field of computer science that helps machines to understand, interpret, and generate human language. It plays an important role in **information extraction (IE)** by identifying keywords from the text and enhances **information retrieval (IR)** by improving search accuracy, which helps in fetching relevant data Chen et al. [2017], Lee et al. [2021]. These capabilities are essential for **Question Answering (QA)** system, which is developed to automatically answer the queries of the users based on the database or a set of documents. It tries to provide specific answers to the posed questions in a natural language. It can be tailored to various domains, such as education, healthcare, e-commerce, tourism, etc. It provides direct, accurate, and user-friendly access to information in a natural language, accessing knowledge faster and more effective by bridging the gap between questions and accurate answers.

The QA systems offer efficiency, convenience, and enhanced user experience by providing domain-specific expertise with continuous 24/7 availability. Baseball Green et al. [1961] and Lunar Woods [1973] were among the initial QA systems, developed to address specific domains. The Baseball system was designed to answer queries related to the U.S. baseball league over a one-year period, while the Lunar system focused on responding to questions concerning the geological analysis of rock samples retrieved during the Apollo moon mission. Traditional QA paradigms established foundational frameworks through rule-based approaches, incorporating template-based Fabbri et al. [2020], Sammut and Banerji [1986], Liu et al. [2018], syntax-based Straach and Truemper [1999], Harabagiu et al. [2005], Heilman and Smith [2010], and semantic-based Levy and Andrew [2006], Dhole and Manning [2020], Yao and Zhang [2010] methodologies. These early implementations laid the crucial foundation through dependency parsing and the identification of semantic relationships within textual contexts Cao et al. [2017]. QA systems have undergone substantial advancements through the development of extractive and abstractive methodologies Chen et al. [2017]. Abstractive QA systems generate free-form answers by mimicking human summarization, enabled by advances in sequence-to-sequence architectures like AG Lewis et al. [2020a], Llama3 Tran et al. [2024], Yadav et al. [2024]. The T5 model Raffel et al. [2020] unified diverse NLP tasks into a text-to-text framework, facilitating flexible answer generation, while BART Lewis et al. [2020b] combined bidirectional and autoregressive pretraining to excel in generative benchmarks. Extractive QA systems identify and retrieve answer spans from provided contexts, evolved from rule-based approaches to transformer-based architectures Farea and Emmert-Streib [2024], Pandey and Roy [2024], Sengupta et al. [2025]. The introduction of BERT by Devlin et al. (2018) Devlin et al. [2019] marked a paradigm shift, leveraging bidirectional attention to contextualize tokens dynamically, thereby achieving state-of-the-art performance on benchmarks such as SQuAD Rajpurkar et al. [2016a]. Subsequent models like RoBERTa Liu et al. [2019] and ELECTRA Clark et al. [2020] refined pre-training strategies, optimizing span prediction accuracy and computational efficiency. Authors have also explored retrieval methods for extractive QA Kruit et al. [2024]. Despite progress, extractive systems remain constrained by their inability to synthesize answers for less explored domains, such as tourism in low-resource languages.

According to the statistics released by the Uttar Pradesh Government on 30 March 2023, the number of tourists in 2022 in the Varanasi region was 71,701,816², which increased to 129,405,720 in 2023, according to data released on 29 March 2024³. This significant growth in tourism, combined with Hindi's status as the most widely spoken language in the region, highlights the necessity for language-accessible resources to cater to the needs of visitors. According to the 2011 Census of India, Hindi has 528 million speakers⁴, constituting 43.63% of the total population.

1.1 Contribution

Given the confluence of the rising tourist influx and Hindi's dominance, we have developed a dedicated Hindi QA system for Varanasi Tourism to enhance accessibility and user engagement. Here is the key contribution of the article:

- We have developed a comprehensive Hindi QA dataset for Varanasi tourism, initially comprising 7,715 manually curated question-answer pairs, which was subsequently augmented to 27,455 pairs using a Llama-based approach. This dataset spans diverse subdomains, including Temples, Ashrams, Kunds, Museums, Ganga Aarti, Cruise, Travel Agencies, Food Court, Public Toilets, and General Enquiry.
- We have conducted an extensive experimental study comparing the performance of two state-of-the-art foundation models-mBERT and RoBERTa, fine-tuned using supervised fine-tuning on this dataset.
- Since the individual subdomains are very small, they are treated as low-resource settings that necessitate careful optimization to extract maximum performance from limited data. To address this, we integrate Low-Rank

²<https://uptourism.gov.in/en/article/year-wise-tourist-statistics>

³<https://uptourism.gov.in/en/post/Year-wise-Tourist-Statistics>

⁴https://language.census.gov.in/eLanguageDivision_VirtualPath/eArchive/pdf/C-16_2011.pdf

Adaptation (LoRA) with mBERT, systematically exploring various configurations with ranks of 2, 4, 8, 16, and 32 to optimize parameter efficiency and achieve the best trade-off between performance and model complexity.

2 Related Work

Roy et al. (2022) [Roy et al., 2022] explored a generative approach models like OAAG (using BiLSTM) & Chime (using XL-Net transformer), for analysis, the BM25 algorithm for answering customer queries in e-commerce. An Amazon product review dataset was utilized, comprising 1.4 million user reviews along with corresponding product evaluations. The models are evaluated on two product categories: Home & Kitchen and Sports & Outdoors, using the ROUGE metric. Al-Laith (2025) Al-Laith [2025] explored multilingual LLMs for financial QA, demonstrating fine-tuned XLM-RoBERTa-Large's superiority (SAS: 0.96–0.98, EM: 0.76–0.81) over GPT-4o, which improved via few-shot learning (EM: 0.48–0.52). The study highlighted trade-offs between precision in fine-tuned models for extractive tasks and generative LLMs' flexibility in low-resource scenarios. These insights reinforce context-driven model selection for domain-specific NLP, balancing accuracy and adaptability. Kasai et al. (2023) Kasai et al. [2023] proposed a real-time QA framework that implements six baseline approaches, leveraging a robust pre-trained model. These include four open-book methods based on Dense Passage Retrieval (DPR) and two closed-book models—Retrieval-Augmented Generation (RAG) and a prompting-based approach utilizing GPT-3.

Ali Al-Laith (2025) Al-Laith [2025] worked on the Exploring the Effectiveness of Multilingual and Generative Large Language Models for Question Answering in Financial Texts, provided a comprehensive analysis of large language models (LLMs) for financial causality detection using the FinCausal 2025 shared task dataset. This dataset consisted of 3,999 training samples and 999 test samples from financial disclosures in English and Spanish, structured for a hybrid question-answering task. The study employed both generative and discriminative techniques, utilizing four pre-trained language models: GPT-4o for generative QA and fine-tuned versions of XLM-RoBERTa (base and large) and BERT-base-multilingual-cased for multilingual QA. The evaluation was based on two key accuracy metrics: Semantic Answer Similarity (SAS) and Exact Match (EM). The results revealed that the fine-tuned XLM-RoBERTa-Large model outperformed others, achieving SAS scores of 0.96 (English) and 0.98 (Spanish), and EM scores of 0.762 and 0.808, respectively. While GPT-4o initially underperformed in a zero-shot setting (SAS: 0.77, EM: 0.002), it showed significant improvement with few-shot prompting, reaching SAS scores of 0.94 and EM scores of 0.515 (English) and 0.487 (Spanish). The paper effectively highlighted the strengths of fine-tuned PLMs in extractive question answering while showcasing GPT-4o's adaptability in scenarios where extensive fine-tuning was not feasible. The study's detailed comparative analysis and experimental techniques provided valuable insights into financial NLP, reinforcing the importance of model selection in domain-specific tasks.

Kasai et al. (2024) Kasai et al. [2023] developed an framework and a benchmarking timeline for real-time QA system submission. For the evaluation of the system, they used 1,470 QA pairs and further, they provided 2,886 QA pairs and they included nearly 30 multiple choice questions at 3 am GMT on every Saturday and they used API search for these questions. REALTIME QA executed six baselines in real time that are based on a strong pre-trained model: four open-book and two closed-book models. Open-book QA model retrieved the documents from DPR, and for answer prediction, they used two methods: retrieval-augmented generation (RAG) and a prompting method with GPT-3. They used the BART-based RAG-sequence model for the RAG baseline, again finetuned on Natural Questions from the Transformers library. Two methods were used for closed-book QA: the finetuning method and the prompting method. In the Finetuning Method, they used the T5 model finetuned on the Natural Questions data again from the Transformers library. They applied a prompting method to GPT-3 similar to the open-book baselines. GPT-3 with retrieval achieves the best performance; EM scores were 34.6, and F1 scores were 45.3.

2.1 Indian Languages QA Systems Including Hindi

Thirumala and Ferracane (2022) Thirumala and Ferracane [2022] explored extractive question answering for Hindi and Tamil using Wikipedia articles as context, with questions prepared by native speakers. They evaluated XLM-RoBERTa, XLM-RoBERTa+fine-tuning, and RoBERTa+Hindi/Tamil fine-tuning. The models achieved word-level Jaccard scores of 0.656 (XLM-RoBERTa), 0.749 (XLM-RoBERTa+fine-tune), 0.958 (RoBERTa+Hindi fine-tune), and 0.829 (RoBERTa+Tamil fine-tune). They concluded that RoBERTa + Hindi/Tamil fine-tuning outperformed the other models.

Amin et al. (2023) Amin et al. [2023] has developed a Marathi QA system using multilingual models such as DistilBERT, mBERT, XLM-RoBERTa, Indo-Aryan XLM, RoBERTa, MuRIL, and monolingual models- MahabERT, IndicBERT, MahaRoBERTa, MahaALBERT, Marathi DistilBERT, DevBERT, DevRoBERTa, DevALBERT, DevBERT-Scratch, along with the MrSQuAD dataset containing 47,065 training and 5,832 testing QA pairs. Among multilingual models, MuRIL achieved the highest performance with an EM score of 0.64, BERT score of 0.93, and F1 score of 0.74. Among

monolingual models, MahaBERT and DevBERT performed best, each achieving an EM score of 0.63, BERT score of 0.93 (MahaBERT)/0.92 (DevBERT), and F1 score of 0.73. The study compared multilingual and monolingual approaches to assess their effectiveness on the Marathi dataset.

Sabane et al. (2023) Sabane et al. [2023] developed a QA dataset for Hindi and Marathi by translating the English SQuAD 2.0 dataset Rajpurkar et al. [2016b] using IndicTrans Ramesh et al. [2022] from AI4Bharat. The dataset, derived from Wikipedia articles, comprises 28,000 samples, with 21,000 for training, 4,200 for validation, and 4,200 for testing. Transformer-based models, including mBERT, XLM-RoBERTa, and DistilBERT, were used for experimentation.

For Marathi, MahaBERT, a fine-tuned version of multilingual BERT-base cased, achieved the best results, with an EM score of 42.97%, Rouge-2 of 0.38, Rouge-L of 0.62, BLEU (Unigram) of 57.42%, and BLEU (Bigram) of 39.70%. For Hindi, HindiBERT, trained on publicly available Hindi monolingual datasets, performed best with an EM score of 47.84%, Rouge-2 of 0.36, Rouge-L of 0.66, BLEU (unigram) of 61.47%, and BLEU (Bigram) of 38.51%. The study systematically evaluates monolingual and multilingual large models to assess their effectiveness on Hindi and Marathi QA tasks. Singh et al. (2025) Singh et al. [2024] developed the INDIC QA BENCHMARK to evaluate LLMs for Indic languages. Their dataset, comprising context-question-answer triples, spans multiple domains, including geography, Indian culture, and news. They assessed Extractive and Generative QA approaches using Bloom, Gemma, Llama-3, and OpenHathi, testing across eleven Indic languages, such as Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Pali, Tamil and Telugu, to analyze LLM performance in multilingual settings.

Vats et al. (2025) Vats et al. [2025] explored State Space Models for structured QA in Hindi and Marathi, addressing challenges like linguistic diversity, complex grammar, and data scarcity. Using a dataset of 28,000 samples (21,000 Hindi, 18,500 Marathi for training, and 7,000 per language for testing), they evaluated models including Mamba, Mamba-2, Falcon Mamba, Jamba, Zamba, Samba, and Hymba. Mamba-2 achieved the best performance in contextual understanding, long-sequence modeling, and token-level alignment, with fine-tuning improving span localization and semantic understanding. Hindi outperformed Marathi due to a larger dataset and better tokenization, while Marathi's complex morphology and syntax posed challenges. Models like Falcon Mamba and Hymba struggled with Indic linguistic complexity. The study highlighted the need for language-specific pre-training and proposed future improvements in dataset diversity and model adaptation for a unified multilingual QA system. Table 1 summarizes the work done on QA systems for distinct domains.

Table 1: Comparison of existing QA datasets

Language	Domain	Source	Corpus Size	Model	Limitations	Reference
Chinese	Tourism	Horse beehive travel and Baidu travel.	32786 The NER dataset contains training data (21859), verification data (5463) and testing data (5464)	Pipelined approach (CRF, BERT, BiLSTM+CRF, BERT+CRF, Transformer+CRF, HMM, BERT+BiLSTM+CRF)	Template based approach, Not adaptive for unseen domain	Sui [2021]
Chinese	Tourism	eTrip, Tuinu, and CNCN	3,635 the dataset	BERT used as a NER	Restricted Data Source Diversity, Fluency Challenge in Answer Text	Li et al. [2022]
Chinese	Tourism	Cultour	51K dataset	LoRA	Inconsistent model performance	Wei et al. [2024]
Korean	Tourism	Chatbot system consisting of the NER server, DST server, Neo4j graph DB's tourism knowledge base, and QA server	More than 1,000 contexts and 10,000 questions	BigBird	Inconsistent model performance	Kang et al. [2024]
English	Tourism	Reddit, specifically from travel domain subreddits	1,000 posts	Mistral QLoRA, LLaMa QLoRA, Mistral RAFT, LLaMa RAFT, Mistral RAFT RLHF, GPT-4	Limited time period dataset (2021 only), noisy output from the model	Meyer et al. [2024]
English and Spanish	Financial	FinCausal 2025 shared task dataset	3,999 training samples and 999 test samples	GPT-4o, XLM-RoBERTa, BERT-base	Limited finance dataset, sub-domains unexplored	Al-Laith [2025]
Hindi and Tamil	General domain including Wikipedia articles	Kaggle competition chai - Hindi and Tamil	1,104 entries (740 entries in Hindi and 364 entries in Tamil)	XLM-RoBERTa, XLM-RoBERTa+finetune, RoBERTa+Hindi finetune/Tamil finetune	Annotation Inconsistencies and Noisy Labels, Incorrect Predictions due to Language Transfer Issues	Thirumala and Ferencane [2022]
Marathi	General domain including Wikipedia articles	MrSQuAD by D Am-invari, Sagar Kulni	17,337 contexts, 46,960 questions, and 30,162 answers	Multilingual models (DistilBERT, mBERT, XLM-RoBERTa, Indo-Aryan XLM, RoBERTa, MuRIL) and monolingual models (MahaBERT, IndicBERT, MahaRoBERTa, MahaAIBERT, Marathi DistilBERT, DevBERT, DevRoBERTa, DevAIBERT, DevBERT-Scratch)	translation inaccuracies, script validation issues, non-updated machine translation dataset, not manually annotated	Amin et al. [2023]
Hindi and Marathi	Wikipedia articles	SQuAD 2.0 English dataset	28,000 in Hindi and Marathi	MBert, XLM-RoBERTa, DistilBERT, HindBERT and HindRoBERT	Translated Dataset Instead of Native Text	Sabane et al. [2023]
Hindi and Marathi	History, Science, Literature	Mamba, Mamba-2, Falcon Mamba, Jamba, Zamba, Samba, and Hymba. Mamba-2	28000 Hindi dataset	Mamba, Mamba-2, Falcon Mamba, Jamba, Zamba, Samba, and Hymba. Mamba-2	Scarcity of High-Quality, Large-Scale QA Datasets for Indic Languages, Limited Applicability to Underrepresented Indic Languages, Weak Performance on Multi-Sentence or Ambiguous Questions	Vats et al. [2025]

2.2 QA systems for Tourism Domain

Sui (2021) Sui [2021] developed a tourism QA system using a knowledge graph. The system converts natural language questions into Cypher queries, enabling efficient retrieval of tourism-related information. Data was sourced from popular travel platforms such as Horse Beehive Travel and Baidu Travel, ensuring a rich knowledge base for query execution. Similarly, Li et al. (2022) Li et al. [2022] also proposed a knowledge-based tourism QA system specifically for Zhejiang, China. The researchers collected data from Baidu keywords and surveys to identify key scenic spot attributes. Kang et al. (2024) Kang et al. [2024] developed a Machine Reading Comprehension (MRC)-based tourism QA system using BERT series pre-trained models. The system, implemented as a smart tourism chatbot, processes input sentences up to 512 tokens in standard Transformer models, while the BigBird model handles 4096 tokens using block sparse attention for efficiency. The tourism QA dataset contains 1,000+ contexts and 10,000+ questions. The KoBigBird model achieved EM 96.85 and F1 98.84, demonstrating high accuracy in tourism-related QA tasks.

Kirtl et al. (2024) Kirtl et al. [2024] focused on developing an AI chatbot for tourism and viniculture in Türkiye using OpenAI's GPT-3.5-turbo, fine-tuned with QA and plaintext formats. The resulting model, WineBot, demonstrated superior performance over base ChatGPT models by reducing misleading or incomplete responses, highlighting the efficacy of domain-specific fine-tuning. Wei et al. (2024) Wei et al. [2024] introduced Cultour, a Chinese SFT dataset for culture and tourism, combining 9,004 QA pairs, 1,792 travelogues, and 2,027 diverse QAs. Using LoRA-based fine-tuning, their TourLLM-7B model outperformed benchmarks like ChatGPT and Qwen1.5 in automated metrics (e.g., BLEU-1: 2.77, Meteor: 18.54) and human evaluations (CRA scores), though Qwen1.5-7B excelled in ROUGE-1 (27.05) and readability (2.69). Meyer et al. (2024) Meyer et al. [2024] compared QLoRA and RAFT fine-tuning techniques on LLaMa 2 7B and Mistral 7B for travel chatbots. Using a refined dataset of 10,500 Reddit-sourced QA entries, they found Mistral with RAFT outperformed LLaMa, though post-processing was critical. Evaluation combined human feedback, NLP metrics, Ragas, and GPT-4, emphasizing human input as vital for accuracy.

This highlights that the tourism domain remains underexplored in the Indian context in Indian languages, with limited datasets available to support the development of QA systems in this field.

3 Dataset Preparation

In this study, we present a comprehensive Hindi QA dataset tailored to Varanasi tourism. The dataset comprises 7,755 manually created QA pairs, which were subsequently augmented using LLMs to address a wide range of tourist queries.

3.1 Data Collection and Augmentation Process

A detailed questionnaire was first developed to capture all necessary information related to Varanasi tourism. Data were collected from both secondary sources (as mentioned in Table 14) and primary sources (official temple websites⁵⁶, pamphlets, visiting cards, etc) during the project tenure (February to August 2024). Primary data were gathered during 10 field visits⁸ to key sites including temples, travel agencies, museums, ashrams, and kunds. The primary data served to validate and refine the information obtained from secondary sources, with rigorous proofreading ensuring accuracy. After the initial manual creation, the dataset was augmented by using the Llama model to generate additional similar Hindi questions. These Llama-generated questions were cross-checked at the syntactic and semantic levels by the annotators, validated using the agreement between the annotators, measured by Cohen's Kappa score, which was 0.9. The example of semantically repeated pairs, removed by the annotators shown in Appendix ??.

3.2 Sub-domain Covered

The Hindi QA dataset addresses multiple sub-domains pertinent to Varanasi tourism, ensuring that visitors receive well-rounded and reliable information. The key domains include:

- **Temples:** Details on timings, rituals, special events, and facilities at major temples such as Shri Kashi Vishwanath Mandir, Sankat Mochan Mandir, and Durga Mandir.
- **Museums:** Information regarding operating hours, entry fees, and facilities available at museums like Bharat Kala Bhavan, Ramnagar Fort Museum, and Man Singh Observatory.

⁵<https://www.shrikashivishwanath.org/>

⁶<https://kashiannapurnaannakshetratrust.org/>

⁷<https://sankatmochanmandirvaranasi.com/>

⁸Researchers physically visited majority of the tourist sites to get the authentic and reliable information from the tourist places in Varanasi.

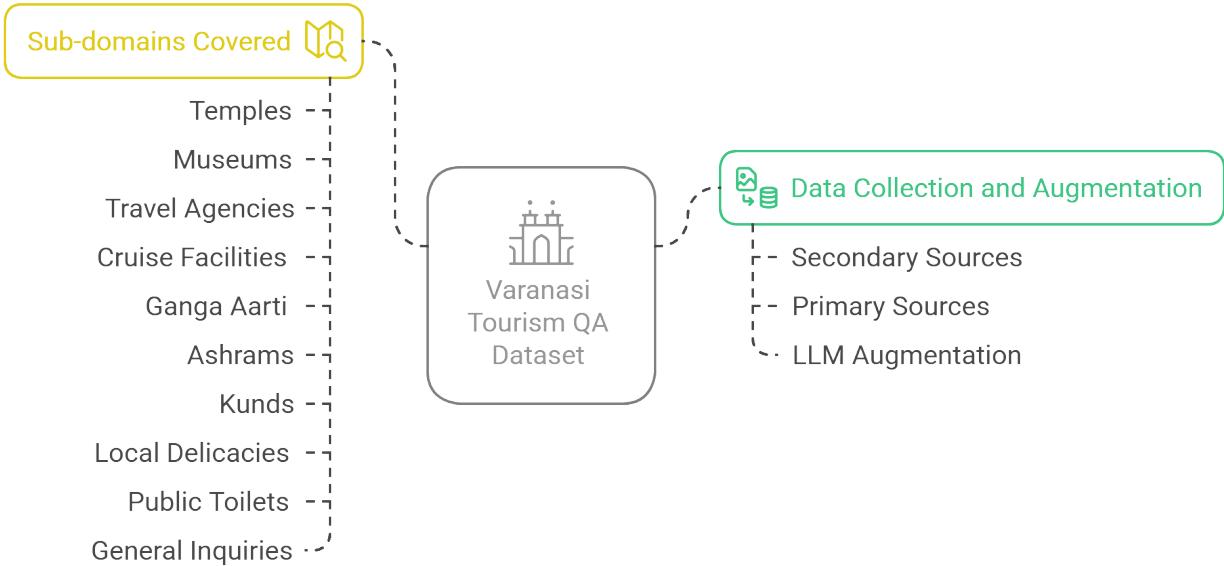


Figure 1: Varanasi Tourism QA Dataset: Structure

- **Travel Agencies:** Insights into guided tours, transportation options, travel guides, and customized itineraries offered by local agencies.
- **Cruise Facilities:** Data on Ganges river cruises, including routes, timings, ticketing booking, and onboard amenities.
- **Ganga Aarti:** Information about the Ganga Aarti at Dashashwamedh and Assi Ghat, covering timings, viewing spots, and event durations.
- **Ashrams:** Coverage of oldest spiritual ashrams (e.g., Andhra Ashram, Aditya Ashram, Jangamvadi) offering accommodations and meditation programs.
- **Kunds:** Coverage of historical kunds (e.g., Durga Kund, Lolark Kund) and spiritual ashrams (e.g., Andhra Ashram, Aditya Ashram, Jangamvadi) offering accommodations and meditation programs.
- **Local Delicacies:** A showcase of authentic Banarasi dishes such as *malaiyo* (light/whipped cream delicacy), *lavang latika* (clove-flavored sweet pastry), *kachori sabzi* (fried stuffed bread with vegetable curry), *tamatar chat* (spiced tomato-based street snack).
- **Public Toilets:** Details about well-maintained public toilet facilities (e.g., the Sulabh complex) in key tourist areas.
- **General Inquiries:** A broad category addressing common questions about weather, best travel seasons, local transportation, accommodation, selfie spots, and boating facilities.

To curate this dataset, both purposive and convenience sampling techniques were employed. The outline of data collection, domains, and augmentation is represented in Figure 1.

3.2.1 Purposive Sampling

We selectively targeted prominent and high-demand locations such as:

- *Temples:* Including Shri Kashi Vishwanath Mandir, Annapurna Mandir, Sankat Mochan Mandir, Vishalakshi Mandir, etc.
- *Ashrams:* Such as Annapurna Ashram, Kumarswami Ashram, and Jangamwadi Mutt.
- *Travel Agencies:* Including Shreenath Ji Tours and Travels, Endeman Tour, and Ganga Travels.
- *Kunds:* Like Lolark Kund, Durga Kund, Krim Kund, and Manikarnika Kund.
- *Major Activities:* Such as Ganga Aarti and cruise services.

हाँ, श्री काशी विश्वनाथ मंदिर वाराणसी के गोदौलिया में स्थित है। गोदौलिया वाराणसी का प्रमुख धार्मिक क्षेत्र है। मंदिर तक पहुँचने के लिए गोदौलिया से पैदल मार्ग या रिक्शा का उपयोग किया जाता है, जहाँ भक्तों को आध्यात्मिक शांति का अनुभव होता है। गोदौलिया का यह स्थान प्राचीनता और आधुनिकता का अनूठा संगम है, जो वाराणसी के ऐतिहासिक और धार्मिक महत्व को दर्शाता है।

1. श्री काशी विश्वनाथ मंदिर कहाँ स्थित है?
 - श्री काशी विश्वनाथ मंदिर वाराणसी के गोदौलिया में स्थित है।

2. श्री काशी विश्वनाथ मंदिर वाराणसी में किस क्षेत्र में स्थित है?
 - श्री काशी विश्वनाथ मंदिर वाराणसी के गोदौलिया में स्थित है।

3. गोदौलिया चौक से श्री काशी विश्वनाथ मंदिर तक पहुँचने के लिए कैसे जा सकता है?
 - गोदौलिया चौक से श्री काशी विश्वनाथ मंदिर तक पहुँचने के लिए पैदल मार्ग मिल सकता है या रिक्शा सुविधा भी उपलब्ध है।

Figure 2: Tourism Domain Example

3.2.2 Convenience Sampling

Due to constraints such as high location density, limited access to some informants, and time restrictions, convenience sampling was applied to include accessible locations like *Mani Mandir, Rudra Kund, and Udupi 2 Mumbai Food Court*. This ensured that the dataset remained diverse yet practical.

3.3 Sub-domain-wise Statistics

The statistics of the data set for various subdomains related to Varanasi tourism are presented, with a focus on the Manually Created Hindi Question Answer Dataset (MCHQAD) and the Llama Generated Hindi Question Answer Dataset (LGHQAD), which was augmented by zero-shot prompting Scius-Bertrand et al. [2025]. Among the covered domains, *Temples* constitute the largest category, comprising 2,686 MCHQAD and 9,496 LGHQAD. The *Kunds* subdomain includes 470 MCHQAD and 2,398 LGHQAD, while the *Ashram* category consists of 1,555 MCHQAD and 5,284 LGHQAD. The *Museum* domain includes 484 MCHQAD but does not contain any LGHQAD. The *Travel* The agency domain represents the second largest category in the dataset, with 2,413 MCHQAD and 6,828 LGHQAD. This highlights the significant focus on tourism-related services and infrastructure.

Additionally, several smaller yet essential subdomains have been included to enhance the comprehensiveness of the dataset. The *Ganga Aarti* category comprises 15 MCHQAD and 46 LGHQAD, reflecting the importance of this spiritual event for visitors. *Cruise Services*, which cater to river tourism experiences, contain 19 MCHQAD and 60 LGHQAD. The *Food Courts* sub-domain, which provides insights into local dining options, includes 11 MCHQAD and 15 LGHQAD. In addition, the public infrastructure is also addressed within the data set. The *Public Toilets* domain includes 9 MCHQAD and 13 LGHQAD, ensuring that accessibility-related queries are covered. The *General Enquiries* category, with 53 MCHQAD and 81 LGHQAD, serves as a broad repository for common tourist questions and essential information. An example QA pair from the dataset is shown in Figure 2. The English translation of the example is given below:

“Yes, Shri Kashi Vishwanath Temple is located in Godowlia, Varanasi. Godowlia is the main religious area of Varanasi. Devotees can walk on foot or use a rickshaw from Godowlia to reach the temple, where they can experience spiritual peace. Godowlia is a unique place with the confluence of antiquity and modernity, which reflects the historical and religious importance of Varanasi.

Que. - 1. Where is Shri Kashi Vishwanath Temple located?

Ans. - Shri Kashi Vishwanath Temple is located in Godaulia, Varanasi.

Que. - 2. In which area the Shri Kashi Vishwanath Temple is located in Varanasi?

Ans. - Shri Kashi Vishwanath Temple is located in Godowlia, Varanasi.

Que. - 3. How to reach Shri Kashi Vishwanath Temple from Godaulia Chowk?

Ans. - To reach Shri Kashi Vishwanath Temple from Godowlia Chowk, one can go by foot, or a rickshaw facility is also available.”

These diverse categories collectively contribute to the building of a well-rounded QA system for Varanasi tourism, ensuring coverage of both the main and minor aspects of the tourist experience. The total numbers of MCHQAD and LGHQAD are 7715 and 27455, respectively, as shown in Table 2. A detailed questionnaire was first developed to capture all necessary information related to Varanasi tourism. Data were collected from both secondary sources (as mentioned in Table 14) and primary sources (official temple websites, pamphlets, visiting cards, etc). Primary data were gathered during 10 field visits to key sites, including temples, travel agencies, museums, ashrams, and kunds. The primary data served to validate and refine the information obtained from secondary sources, with rigorous proofreading ensuring accuracy. A total of six annotators were involved in the creation of the dataset. Two annotators created the initial manual dataset, and four were involved in the data augmentation process. Initially, two Hindi language experts validated the manually created dataset, which was revised and proofread multiple times during its development. After the manual creation, the dataset was augmented using the LLaMA model to generate additional similar Hindi questions, which were then cross-checked and manually validated.

Table 2: Sub-domain-wise Distribution of QA Dataset for Varanasi Tourism

Sub-domain	# of MCHQAD	# of LGHQAD
Temples	2686	11691
Kunds	470	2398
Ashrams	1555	5284
Museums	484	1039
Travel Agencies	2413	6828
Ganga Aarti	15	46
Cruise	19	60
Food Court	11	15
Public Toilet	9	13
General Enquiries	53	81
Total	7715	27455

4 Problem Formulation

Extractive QA is a fundamental task in NLP that requires models to extract relevant information from a given context to answer a question. Formally, a QA system can be defined as a function:

$$f : (Q, C) \rightarrow A \quad (1)$$

where Q represents the question, C denotes the context or passage containing the answer, and A is the extracted answer span. In this article, we relied on Transformer-based modern approaches, such as BERT (Bidirectional Encoder Representations from Transformers), RoBERTa (Robustly Optimized BERT Pretraining Approach), along with associated fine-tuning techniques like LORA (Low-Rank Adaptation) to achieve high accuracy while maintaining efficiency. The overview of the complete architecture is shown in Figure 3.

5 Question Answering with BERT

BERT Devlin et al. [2019] is a transformer-based model for contextualized word representations using bidirectional self-attention. Given an input sequence X of tokens $[[CLS], x_1, \dots, x_n, [SEP]]$, BERT applies multiple layers of a transformer encoder Enc_θ , producing contextualized embeddings:

$$H = \text{Enc}_\theta(X) = \{h_1, h_2, \dots, h_n\}, \quad h_i \in \mathbb{R}^d \quad (2)$$

BERT pretraining involves Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), where MLM randomly masks tokens $X_m \subset X$ and predicts them using adjacent words:

$$L_{\text{MLM}} = - \sum_{x_i \in X_m} \log P_\theta(x_i | X_{\text{masked}}) \quad (3)$$

Similarly, NSP tells the consecutiveness of two sentences (S_1, S_2) by using:

$$L_{\text{NSP}} = -y \log P_\theta(y | S_1, S_2) - (1 - y) \log(1 - P_\theta(y | S_1, S_2)) \quad (4)$$

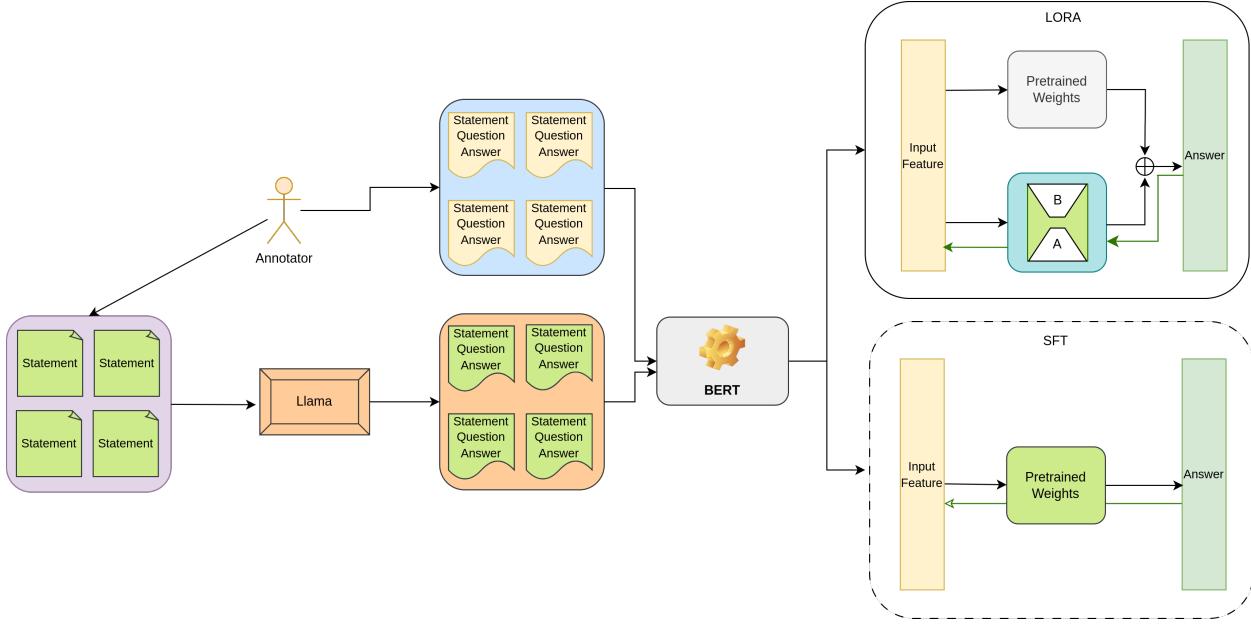


Figure 3: Overview of the complete methodology of the model. The green arrows in the SFT and LoRA modules indicate the back-propagation steps during fine-tuning. At any given time, either SFT or LoRA is activated for modeling the question-answering task.

Since both MLM and NSP are performed during pretraining, the objective is to minimize the total loss:

$$L_{BERT} = L_{MLM} + L_{NSP} \quad (5)$$

5.1 Question Answering

As specified in Section 4 for QA input and output, the input sequence to BERT consists of the given question Q and context C , represented as:

$$X = [\text{CLS}]; Q; [\text{SEP}]; C; [\text{SEP}] \quad (6)$$

where $[\text{CLS}]$ is a special classification token and $[\text{SEP}]$ marks boundaries between the question and the context. The model applies multiple layers of a Transformer-based encoder Enc_θ (parameterized by θ) to generate contextualized token embeddings using equation 2

6 Question Answering With RoBERTa

RoBERTa Liu et al. [2019] builds upon BERT but introduces modifications to the pretraining and fine-tuning mechanisms, leading to superior performance on QA tasks. Let X be a sequence of tokens designed using equation 6 and let $M(X)$ denote a random masked version of X . In BERT, static masking is applied once, whereas RoBERTa uses dynamic masking, where each token x_i is masked with a probability p_m in each batch:

$$X_{\text{masked}} = M(X), \quad M(X) = \{x_i \text{ with probability } p_m\} \quad (7)$$

The MLM loss is computed as:

$$L_{MLM} = - \sum_{x_i \in M(X)} \log P_\phi(x_i | X_{\text{masked}}) \quad (8)$$

where P_ϕ is the probability assigned by RoBERTa's encoder Enc_ϕ (as shown in equation 2 parameterized by ϕ).

7 Supervised Fine-Tuning with Task-Specific Layers

Supervised fine-tuning the pretrained BERT or RoBERTa involves modifying their architecture by adding task-specific layers and optimizing the model on a downstream task loss function. In the context of extractive QA, this requires training the model to predict the start and end indices of the answer span within a given context.

For span prediction, both models employ two linear classifiers to determine the start and end positions of the answer:

$$P(s) = \text{softmax}(HW_s), \quad P(e) = \text{softmax}(HW_e) \quad (9)$$

where $W_s, W_e \in \mathbb{R}^d$ are learned weight vectors for predicting start and end positions, and softmax ensures probabilities sum to 1 across possible token positions. H is the pretrained weights of the models. The model is fine-tuned by minimizing the negative log-likelihood loss for the correct start and end indices (s^*, e^*):

$$L_{\text{QA}} = -\log P(s = s^*) - \log P(e = e^*) \quad (10)$$

where s^* and e^* are the starting and end positions of the ground truth, while s and e are respective positions obtained from prediction. The model maximizes the probability of selecting the correct answer span.

7.1 Fine-Tuning with LoRA

Fine-tuning BERT or RoBERTa on large QA datasets requires updating all parameters θ , leading to high computational costs. The standard parameter updation expression is represented by:

$$\theta_{\text{new}} = \theta - \eta \nabla_{\theta} L_{\text{QA}} \quad (11)$$

where η is the learning rate. This full fine-tuning is expensive. To address this, LoRA (Low-Rank Adaptation) Shen et al. freezes the original model parameters and injects low-rank updates into weight matrices. Instead of modifying a full weight matrix $W \in \mathbb{R}^{d \times k}$, LoRA decomposes updates as:

$$\Delta W = BA^{\top} \quad (12)$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{k \times r}$ are low-rank matrices and $r \ll \min(d, k)$ ensures a small number of trainable parameters. The adapted representation for layer l becomes:

$$H^l = (W^l + \Delta W^l)H^{l-1} \quad (13)$$

To maintain stability, the loss function for LoRA fine-tuning regularizes low-rank updates:

$$L_{\text{LoRA}} = L_{\text{QA}} + \lambda \|BA^{\top}\|_F^2 \quad (14)$$

where $\|\cdot\|_F$ is the Frobenius norm, and λ is a hyperparameter controlling regularization. The algorithm 1 describes the SFT and LORA-based fine-tuning.

8 Experimental Settings

For preprocessing, the input Hindi QA data was tokenized with a maximum sequence length of 384 tokens and a stride of 128 tokens to effectively capture overlapping contexts; additionally, both overflowing tokens and offset mappings were returned, and all sequences were padded to the maximum length to ensure uniformity. We conduct an extensive evaluation of two transformer-based language models, multilingual foundation Models-BERT and RoBERTa, employing SFT on a multi-domain dataset comprising diverse sub-domains such as temples, museums, travel agencies, cruise facilities, and other culturally significant areas. We have used `bert-base-multilingual-cased` Devlin et al. [2019], `ai4bharat/indic-bert` Kakwani et al. [2020], `13cube-pune/hindi-bert-v2` Joshi [2022] and `13cube-pune/hindi-roberta` Joshi [2022], referred as **mBERT**, **IndicBERT**, **HindiBERT** and **Hindi-RoBERTa**, respectively. The dataset has been split into 80 and 20 ratios for training and testing, respectively. Both models were fine-tuned under a consistent hyperparameter regime, using a learning rate of 3e-5, a batch size of 48, and training for upto 3 epochs with early stopping based on validation loss, while optimization was

Algorithm 1: Extractive QA Fine-Tuning with BERT/RoBERTa using SFT and LoRA

Input: Pretrained encoder Enc_θ (BERT/RoBERTa), QA dataset $\mathcal{D} = \{(Q_i, C_i, A_i)\}_{i=1}^N$, learning rate η , batch size B , adapter rank r , regularization weight λ

Output: Fine-tuned weights θ_{SFT} and/or LORA adapter θ_{LORA}

▷ Preprocessing

for $i \leftarrow 1$ **to** N **do**

- $X_i \leftarrow [[\text{CLS}]; Q_i; [\text{SEP}]; C_i; [\text{SEP}]]$ // Tokenize input
- $(s_i^*, e_i^*) \leftarrow \text{align}(X_i, A_i)$ // Map answer to token indices

end

▷ SFT

foreach $\text{batch } \mathcal{B} \subset \mathcal{D}$ **with** $|\mathcal{B}| = B$ **do**

- $L_{\text{batch}} \leftarrow 0$
- foreach** $(Q, C, s^*, e^*) \in \mathcal{B}$ **do**

 - $X \leftarrow [[\text{CLS}]; Q; [\text{SEP}]; C; [\text{SEP}]]$
 - $H \leftarrow \text{Enc}_\theta(X)$ // Contextual embedding
 - $P_s \leftarrow \text{softmax}(HW_s); P_e \leftarrow \text{softmax}(HW_e)$ // Start/end probs
 - $L_{\text{QA}} \leftarrow -\log P_s[s^*] - \log P_e[e^*]$ // QA loss
 - $L_{\text{batch}} \leftarrow L_{\text{batch}} + L_{\text{QA}}$

- end**
- $\theta \leftarrow \theta - \eta \nabla_\theta L_{\text{batch}}$ // Gradient update

end

return θ_{SFT}

▷ LORA

Freeze θ ; initialize $A^l \in \mathbb{R}^{k \times r}, B^l \in \mathbb{R}^{d \times r}$ // Adapter params

foreach $\text{batch } \mathcal{B} \subset \mathcal{D}$ **do**

- $L_{\text{batch}} \leftarrow 0$
- foreach** $(Q, C, s^*, e^*) \in \mathcal{B}$ **do**

 - $X \leftarrow [[\text{CLS}]; Q; [\text{SEP}]; C; [\text{SEP}]]$
 - $W_{\text{eff}}^l \leftarrow W^l + B^l(A^l)^\top$ // LoRA update
 - $H \leftarrow \text{Enc}_\theta(X; W_{\text{eff}}^l)$ // Adapted encoding
 - Compute P_s, P_e, L_{QA} as in SFT
 - $L_{\text{batch}} \leftarrow L_{\text{batch}} + L_{\text{QA}}$

- end**
- $L_{\text{reg}} \leftarrow \sum_l \|B^l(A^l)^\top\|_F^2$ // Frobenius regularization
- $L \leftarrow L_{\text{batch}} + \lambda L_{\text{reg}}$
- $(A^l, B^l) \leftarrow (A^l, B^l) - \eta \nabla_{A^l, B^l} L$ // Update adapters

end

return $\theta_{\text{LORA}} \leftarrow \{A^l, B^l\}$

carried out using the AdamW optimizer. To further promote regularization, a weight decay of 0.01 was also employed. We targeted the model’s query and value modules for adaptation, applying a dropout rate of 0.1 to the LoRA layers and opting for no bias adaptation, thereby ensuring parameter efficiency and robust model adjustment. The LoRA exclusively used with mBERT by introducing trainable low-rank matrices. We have explored LoRA configurations with rank values of 2, 4, 8, 16, and 32.

For data augmentation, the Llama model has explored with zero-shot prompting Scius-Bertrand et al. [2025] where the prompt was Statement: Generate questions in Hindi along with their answers from *Context*.

9 Results and Discussion

Table 3, for the Aarti domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 8.67, 2.361, and 20 respectively. The BERT model had an F1 Score of 52.664, a BLEU score of & 28.578, and RougeL score of 20. In the case of the LORA-bert(r2) model, the F1 Score was 13.979, BLEU score was 8.203, and RougeL score was 0. The LORA-bert(r4) model had an F1 Score of 16.512, BLEU score of 7.477, and RougeL score was found to be 0. The LORA-bert(r8) model had an F1 Score of 15.569, BLEU score of 6.409, and RougeL score was found to be 0. The LORA-bert(r16) model had an F1 Score of 15.569, BLEU score of 6.409, and RougeL score was 0. The LORA-bert(r32) model had an F1 Score of 15.569, BLEU score of 6.409, and RougeL score was 0. For the HindiBERT

Table 3: Ganga Aarti Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	8.67	2.361	20
BERT	52.664	28.578	20
LORA-bert(r2)	13.979	18.203	0
LORA-bert(r4)	16.512	7.477	0
LORA-bert(r8)	15.569	6.409	0
LORA-bert(r16)	15.569	6.409	0
LORA-bert(r32)	15.569	6.409	0
HindiBERT	57.774	29.748	20
Hindi-RoBERTa	63.064	45.539	20

model, F1 Score was found to be 57.774, BLEU score 29.748, and RougeL score was 20. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model, were found to be 63.064, 45.539 and 20, respectively.

Table 4: Cruise Domain Model Score

Model	F1 Score	BLEU	RougeL
IndicBERT	6.691	0.875	33.333
BERT	73.477	50.658	34.999
LORA-bert(r2)	12.267	3.032	0
LORA-bert(r4)	12.05	3.829	0
LORA-bert(r8)	12.267	3.032	0
LORA-bert(r16)	12.267	3.032	0
LORA-bert(r32)	12.267	3.032	0
HindiBERT	30.02	15.925	45
Hindi-RoBERTa	41.843	20.305	35

Table 4, for the Cruise domain, the F1, BLEU, and RougeL scores of the IndicBERT model were found to be 6.691, 0.875 and 33.333 respectively. The BERT model had an F1 Score of 73.477, BLEU score was 50.658, and RougeL score 34.999. Whereas in case of the LORA-bert(r2) model, the F1 Score was 12.267, BLEU score was 3.032, and RougeL score was 0. The LORA-bert(r4) model had an F1 Score of 12.05, BLEU score was 3.829, and RougeL score was 0. The LORA-bert(r8) model had an F1 Score of 12.267, BLEU score was 3.032, and RougeL score was 0. The LORA-bert(r16) model had an F1 Score of 12.267, BLEU score was 3.032, and RougeL score was 0. The LORA-bert(r32) model had an F1 Score of 13.605, BLEU score was 9.7, and RougeL score was 0. HindiBERT had an F1 Score of 30.02, BLEU score was 15.925, and RougeL score was 45. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 41.843, 20.305, and 35, respectively.

Table 5, for the Food Court domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 1.666, 0.992, and 0 respectively. The BERT model had the F1 Score of 53.333, BLEU score was 34.042, and RougeL score was 0. Whereas in case of the LORA-bert(r2) model, the F1 Score was 2.469, BLEU score was 0.511, and RougeL score was 0. The LORA-bert(r4) model had an F1 Score of 2.469, BLEU score was 0.511, and RougeL score was 0. The LORA-bert(r8) model had an F1 Score of 2.469, BLEU score was 0.511, and RougeL score was 0. The LORA-bert(r16) model had an F1 Score of 2.469, BLEU score was 0.511, and RougeL score was 0. The LORA-bert(r32) model had an F1 Score of 2.469, BLEU score was 0.511, and RougeL score was 0. HindiBERT had an

Table 5: Food Court Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	1.666	0.992	0
BERT	53.333	34.042	0
LORA-bert(r2)	2.469	0.511	0
LORA-bert(r4)	2.469	0.511	0
LORA-bert(r8)	2.469	0.511	0
LORA-bert(r16)	2.469	0.511	0
LORA-bert(r32)	2.469	0.511	0
HindiBERT	52.197	32.837	0
Hindi-RoBERTa	69.334	46.16	0

F1 Score of 52.197, BLEU score was 32.837, and RougeL was score 0. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 69.334, 46.16, and 0, respectively.

Table 6: Toilet Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	0	0	0
BERT	73.469	30.786	66.666
LORA-bert(r2)	33.939	31.252	0
LORA-bert(r4)	33.939	31.252	0
LORA-bert(r8)	33.939	31.252	0
LORA-bert(r16)	33.939	31.252	0
LORA-bert(r32)	33.939	31.252	0
HindiBERT	64.285	58.456	66.666
Hindi-RoBERTa	66.666	64.118	66.666

Table 6, for the Toilet domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 0, 0, and 0 respectively. The BERT model had an F1 Score of 73.469, BLEU score was 30.786, and RougeL score was 66.666. Whereas in case of the LORA-bert(r2) model, the F1 Score was 33.939, BLEU score was 31.252, and RougeL score was 0. The LORA-bert(r4) model had an F1 Score of 33.939, BLEU score was 31.252, and RougeL score was 0. The LORA-bert(r8) model had an F1 Score of 33.939, BLEU score was 31.252, and RougeL score was 0. The LORA-bert(r16) model had an F1 Score of 33.939, BLEU score was 31.252, and RougeL score was 0. The LORA-bert(r32) model had an F1 Score of 33.939, BLEU score was 31.252, and RougeL score was 0. HindiBERT had an F1 Score of 64.285, BLEU score was 58.456, and RougeL score was 66.666. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 66.666, 64.118, and 66.666, respectively.

Table 7, for the Kund domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 5.357, 0.352 and 14.374 respectively. The BERT model had an F1 Score of 72.14, BLEU score was 64.559, and RougeL score was 14.791. Whereas in case of the LORA-bert(r2) model, the F1 Score was 6.312, BLEU score was 1.99, and RougeL score was 0.208. The LORA-bert(r4) model had an F1 Score of 11.258, BLEU score was 10.105, and RougeL score was 0.208. The LORA-bert(r8) model had an F1 Score of 7.063, BLEU score was 3.868, and RougeL score was 0. The LORA-bert(r16) model had an F1 Score of 7.506, BLEU score was 4.231, and RougeL score was 0.416. The LORA-bert(r32) model had an F1 Score of 9.919, BLEU score was 5.314, and RougeL score was 0.416. HindiBERT

Table 7: **Kund Domain Model Scores**

Model	F1 Score	BLEU	RougeL
IndicBERT	5.357	0.352	14.374
BERT	72.14	64.559	14.791
LORA-bert(r2)	6.312	1.99	0.208
LORA-bert(r4)	11.258	10.105	0.208
LORA-bert(r8)	7.063	3.868	0
LORA-bert(r16)	7.506	4.231	0.416
LORA-bert(r32)	9.919	5.314	0.416
HindiBERT	59.995	42.809	13.958
Hindi-RoBERTa	69.207	56.497	14.583

had an F1 Score of 59.995, BLEU score was 42.809, and RougeL score was 13.958. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 69.207, 56.497 and 14.583 respectively.

Table 8: **Museum Domain Model Scores**

Model	F1 Score	BLEU	RougeL
IndicBERT	7.998	4.089	39.523
BERT	92.738	85.851	32.743
LORA-bert(r2)	57.118	48.933	24.007
LORA-bert(r4)	68.093	59.078	26.984
LORA-bert(r8)	65.344	57.676	26.825
LORA-bert(r16)	65.965	59.367	27.38
LORA-bert(r32)	67.663	59.484	27.936
HindiBERT	91.71	80.869	39.523
Hindi-RoBERTa	98.887	97.674	39.523

Table 8, for the Museum domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 7.998, 4.089, and 39.523 respectively. The BERT model had an F1 Score of 92.283, BLEU score was 85.851, and RougeL score was 32.743. Whereas in case of the LORA-bert(r2) model, the F1 Score was 57.118, BLEU score was 48.933, and RougeL score was 24.007. The LORA-bert(r4) model had an F1 Score of 968.093, BLEU score was 59.078, and RougeL score was 26.984. The LORA-bert(r8) model had an F1 Score of 65.344, BLEU score was 57.676, and RougeL was score 26.825. The LORA-bert(r16) model had an F1 Score of 65.965, BLEU score was 59.367, and RougeL score was 27.38. The LORA-bert(r32) model had an F1 Score of 67.663, BLEU score was 59.484, and RougeL score was 27.936. HindiBERT had an F1 Score of 91.71, BLEU score was 80.869, and RougeL score was 39.523. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 98.887, 97.674, and 39.523 respectively.

Table 9, for the General domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 8.353, 0.984, and 21.568 respectively. The BERT model had an F1 score of 92.715, the BLEU score was 84.223, and the RougeL score was 23.529. whereas in case of the LORA-bert(r2) model, the F1 score was 19.307, the BLEU score was 17.552, and the RougeL score was 10.924. The LORA-bert(r4) model had an F1 score of 15.142, the BLEU score was 18.916, and the RougeL score was 3.921. The LORA-bert(r8) model had an F1 score of 15.142, the BLEU score was 18.768, and the RougeL score was 3.921. The LORA-bert(r16) model had an F1 score of 15.142, the BLEU score was 18.768, and the RougeL score was 3.921. The LORA-bert(r32) model had an F1 score of 15.142, the BLEU score was 18.916, and the RougeL score was 3.921. HindiBERT had an F1 score of 71.698, the BLEU score was 66.071, and the

Table 9: General Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	8.353	0.984	21.568
BERT	92.715	84.223	23.529
LORA-bert(r2)	19.307	17.552	10.924
LORA-bert(r4)	15.142	18.916	3.921
LORA-bert(r8)	15.142	18.768	3.921
LORA-bert(r16)	15.142	18.768	3.921
LORA-bert(r32)	15.142	18.916	3.921
HindiBERT	71.698	66.071	23.529
Hindi-RoBERTa	73.103	74.271	23.529

RougeL score was 23.529. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 73.103, 74.271, and 23.529 respectively.

Table 10: Ashram Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	8.469	2.909	48.085
BERT	92.74	85.889	48.459
LORA-bert(r2)	90.846	84.657	46.825
LORA-bert(r4)	90.909	84.91	46.825
LORA-bert(r8)	90.908	84.74	46.778
LORA-bert(r16)	90.991	84.588	46.825
LORA-bert(r32)	90.934	84.765	46.732
HindiBERT	91.518	84.759	47.432
Hindi-RoBERTa	96.739	95.2	48.482

Table 10, for the Ashram domain, the F1, BLEU and RougeL scores for the IndicBERT model were found to be 8.469, 2.909 and 48.085, respectively. The BERT model had the F1 Score of 92.74, BLEU score was 85.889, and RougeL score was 48.459 whereas in case of the LORA-bert(r2) model, the F1 Score was 90.846, BLEU score was 84.657, and RougeL score was 46.825. The LORA-bert(r4) model had an F1 Score of 90.909, BLEU score was 84.91, and RougeL score was 46.825. The LORA-bert(r8) model had an F1 Score of 90.908, BLEU score was 84.74, and RougeL score was 46.778. The LORA-bert(r16) model had an F1 Score of 90.991, BLEU score was 84.588, and RougeL score was 46.825. The LORA-bert(r32) model had an F1 Score of 90.934, BLEU score was 84.765, and RougeL score was 46.732. HindiBERT had an F1 Score of 91.518, BLEU score was 84.759, and RougeL score was 47.432. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 96.739, 95.2, 48.482 respectively.

Table 11, for the Travel domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 1.299, 0.081, and 1.437 respectively. The BERT model had the F1 Score of 19.569, BLEU score 7.564, and RougeL score 1.437. Whereas in case of the LORA-bert(r2) model, the F1 Score was 19.983, BLEU score was 7.432, and RougeL score was 1.437. The LORA-bert(r4) model had an F1 Score of 19.865, BLEU score was 7.342, and RougeL score was 1.437. The LORA-bert(r8) model had an F1 Score of 19.989, BLEU score was 7.534, and RougeL score was 1.437. The LORA-bert(r16) model had an F1 Score of 19.999, BLEU score was 7.547, and RougeL score was 1.437. The LORA-bert(r32) model had an F1 Score of 19.982, BLEU score was 7.546, and RougeL score was 1.437. HindiBERT

Table 11: Travel Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	1.299	0.081	1.437
BERT	19.569	7.564	1.437
LORA-bert(r2)	19.983	7.432	1.437
LORA-bert(r4)	19.865	7.342	1.408
LORA-bert(r8)	19.989	7.534	1.437
LORA-bert(r16)	19.999	7.547	1.437
LORA-bert(r32)	19.982	7.546	1.437
HindiBERT	88.842	79.401	14.73
Hindi-RoBERTa	93.526	89.755	14.612

had an F1 Score of 88.842, BLEU score was 79.401, and RougeL score was 14.73. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 93.526, 89.755, and 14.612 respectively.

Table 12: Temple Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	8.21	2.533	34.75
BERT	84.232	77.012	35.718
LORA-bert(r2)	75.185	64.233	34.339
LORA-bert(r4)	75.442	64.958	34.312
LORA-bert(r8)	75.424	65.068	34.318
LORA-bert(r16)	75.279	64.623	34.183
LORA-bert(r32)	75.636	65.75	34.201
HindiBERT	75.562	63.918	34.187
Hindi-RoBERTa	86.493	81.747	35.434

Table 12, for the Temple domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 8.21, 2.533, and 34.75 respectively. The BERT model had the F1 Score of 84.232, BLEU score was 77.012, and RougeL score was 35.718. Whereas in case of the LORA-bert(r2) model, the F1 Score was 75.185, BLEU score was 64.233, and RougeL score was 34.339. The LORA-bert(r4) model had an F1 Score of 75.442, BLEU score was 64.958, and RougeL score was 34.312. The LORA-bert(r8) model had an F1 Score of 75.424, BLEU score was 65.068, and RougeL score was 34.318. The LORA-bert(r16) model had an F1 Score of 75.279, BLEU score was 64.623, and RougeL score was 34.183. The LORA-bert(r32) model had an F1 Score of 75.636, BLEU score was 65.75, and RougeL score was 34.201. HindiBERT had an F1 Score of 75.562, BLEU score was 63.918, and RougeL score was 34.187. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 86.493, 81.747, and 35.434 respectively.

Table 13, for the Merged domain, the F1, BLEU, and RougeL scores for the IndicBERT model were found to be 0.644, 0.043, and 1.073 respectively. The BERT model had an F1 Score of 13.825, BLEU score was 3.057, and RougeL score was 0.779. Whereas in the case of the LORA-bert(r2) model, the F1 Score was 21.082, BLEU score was 9.881, and RougeL score was 3.452. The LORA-bert(r4) model had an F1 Score of 20.942, BLEU score was 9.967, and RougeL score was 3.404. The LORA-bert(r8) model had an F1 Score of 14.209, BLEU score was 3.069, and RougeL was score 1.136. The LORA-bert(r16) model had an F1 Score of 14.139, BLEU score was 3.028, and RougeL score was 1.125. The LORA-bert(r32) model had an F1 Score of 14.106, BLEU score was 3.106, and RougeL score was 1.095.

Table 13: Merged Domain Model Scores

Model	F1 Score	BLEU	RougeL
IndicBERT	0.644	0.043	1.073
BERT	13.825	3.057	0.779
LORA-bert (r2)	21.082	9.881	3.452
LORA-bert (r4)	20.942	9.967	3.404
LORA-bert (r8)	14.209	3.069	1.136
LORA-bert (r16)	14.139	3.028	1.125
LORA-bert (r32)	14.106	3.106	1.095
HindiBERT	65.208	49.477	17.261
Hindi-RoBERTa	21.409	11.008	3.349

HindiBERT had an F1 Score of 65.208, BLEU score was 49.477, and RougeL score was 17.261. The F1, BLEU, and RougeL scores for the Hindi-RoBERTa model were found to be 21.409, 11.008, and 3.349, respectively.

Figure 4 and 5 show that models pretrained specifically on Hindi, such as HindiBERT and Hindi-RoBERTa, consistently outperform general-purpose models like BERT and IndicBERT across most domain-specific classification tasks, including Museum, Ashram, Temple, and Travel. Figure 6 shows Hindi-RoBERTa outperformed on the Toilet, Museum, Ashram, Temple, and Cruise domains, considering ROUGE-L score. This underscores the importance of language-specific pretraining for achieving robust performance in Indic NLP. While LoRA-based adaptations (r2 and r4) show improvements over vanilla BERT in selected domains, they still lag behind Hindi-specialized models, indicating that parameter-efficient tuning alone may not suffice without domain-relevant representation learning. IndicBERT, despite its multilingual training, performs poorly overall, suggesting that broader language coverage can dilute effectiveness in Hindi-specific tasks. Notably, the Merged category challenges all general models, with HindiBERT maintaining its advantage, highlighting its robustness in heterogeneous or cross-domain settings.

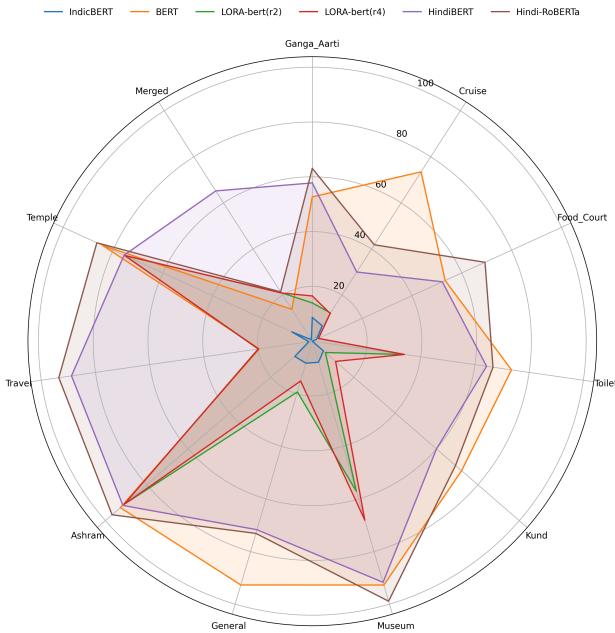


Figure 4: F1 score comparison of the models across 11 domain settings. Configurations with LoRA-based adapters at ranks r8, r16, and r32 consistently underperformed compared to r2 and r4; thus, they are omitted from this figure for clarity.

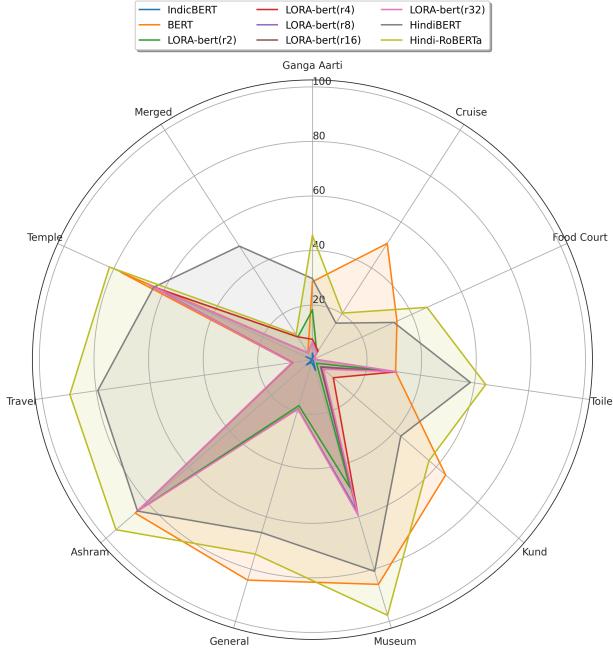


Figure 5: BLEU score comparison of the models across 11 domain settings

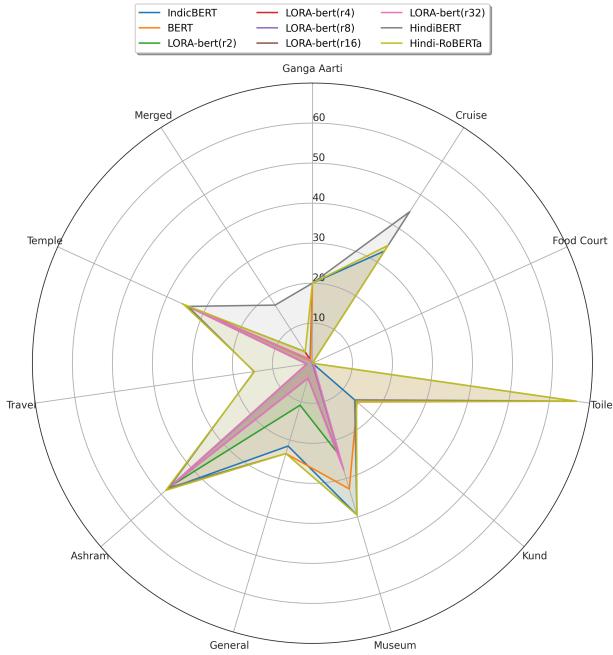


Figure 6: ROUGE-L score comparison of the models across 11 domain settings

10 Conclusion

Developing extractive QA systems for low-resource languages is particularly challenging in domain-specific settings. In this study, we manually prepared a Hindi QA dataset focused on Varanasi tourism—a cultural and spiritual hub renowned for its Bhakti-Bhaav—comprising 7,715 question-answer pairs obtained through extensive fieldwork using purposive and convenience sampling, and covering ten tourism-centric subdomains: Ganga Aarti, Cruise, Food Court, Public Toilet, Kund, Museum, General, Ashram, Travel, and Temple. Which was subsequently augmented to 27,455 pairs

using a Llama-based zero-shot prompting technique. We propose a framework leveraging foundation models-BERT and RoBERTa, fine-tuned using SFT and LORA, to optimize parameter efficiency and task performance. Multiple variants of BERT, including mBERT, Hindi-BERT, IndicBERT, etc., are evaluated to assess their suitability for low-resource domain-specific QA. In the Ganga Aarti, General, and Toilet sub-domains, the BERT model achieved the highest F1 scores, registering 51.223, 76.216, and 73.469, respectively, while in the Cruise, Kund, Museum, Ashram, Travel, Temple, and Food Court domains, the Hindi-RoBERTa model obtained F1 scores of 46.327, 66.639, 95.532, 96.575, 90.91, 86.493, and 66.987 respectively; as the RoBERTa model was pretrained on a large dataset, fine-tuning on the merged sub-domains enabled the Hindi-RoBERTa model to achieve an overall best F1 score of 89.75. As future work, we will leverage the existing multilingual LLM with RAG to enhance model robustness and effectively handle real-time diversified queries. Additionally, a similar Hindi QA dataset can be developed for other domains, such as education, agriculture, and health sciences.

A

Appendix-Sources

Table 14: List of sources and links

Source	Link
Tourism	http://pawanpath.up.gov.in/ https://yappe.in/uttar-pradesh/varanasi https://uptourism.gov.in/en/page/varanasi-sarnath https://uptourism.gov.in/en/article/year-wise-tourist-statistics https://upstdc.co.in/Web/varanasi_tourism https://varanasitemples.in/ https://kashiarchan.com/
Articles	https://www.lonelyplanet.com/articles/guide-to-varanasi https://travel.india.com/guide/destination/experience-the-magic-of-varanasis-silk-sarees-and-wooden-carvings-7358826/ https://timesofindia.indiatimes.com/travel/destinations/5-must-visit-places-in-varanasi/articleshow/115978318.cms https://www.varanasi.org.in/sankatha-temple-varanasi#google_vignette
Blogs	https://classynomad.com/best-street-foods-in-varanasi/ https://livingnomads.com/2023/03/varanasi-travel-blog/
Reviews	https://www.tripadvisor.in/Attractions-g297685-Activities-Varanasi_Varanasi_District_Uttar_Pradesh.html https://www.tripadvisor.com/Tourism-g297685-Varanasi_Varanasi_District_Uttar_Pradesh-Vacations.html
Books	Shiv Lings of Kashi
YouTube	https://youtu.be/890MOuBOUGE https://youtu.be/LLOEVk2FtU?si=PjTJBmYkJv5eJEY https://youtube.com/shorts/kZM9yQeP-UE?si=gaxSV6Cq16eYBpX4
Others	census data, public sector records

B

Appendix-Example of semantically repeated QA pairs

The text in red color (3-4) represents semantically related QA pairs that were dropped by annotators.

1. kyā durgā mamdira mēṁ tīna bāra āratī hōtī hay? [⁹Does three times aarti take place in Durga Mandir?] Ans. hāz, durgā mamdira mēṁ tīna bāra āratī hōtī hayl [Yes, three times aarti take place in Durga Mandir.]
2. durgā mamdira mēṁ kitānī bāra āratī hōtī hay? [How many times does aarti take place at Durga Mandir?] Ans. durgā mamdira mēṁ tīna bāra āratī hōtī hayl [Three times aarti take place at Durga Mandir.]
3. durgā mamdira mēṁ kitānī bāra āratī āyōjita hōtī hay? [How many times is aarti organised at Durga Mandir?] Ans. durgā mamdira mēṁ tīna bāra āratī āyōjita hōtī hayl [Three times aarti is organised at Durga Mandir.]

⁹The[] comprising translation of Hindi text

4. durgā māṇḍira mēṁ kītānī bārā āratī kā āyōjana hōtā hay? [How many times is aarti organised at Durga Mandir?]
 Ans. durgā māṇḍira mēṁ tīnā bārā āratī kā āyōjana hōtā hay! [Three times aarti is organised at Durga Mandir.]

Acknowledgment

We are grateful to acknowledge Transdisciplinary Research (TDR) Grant as a part of the Institute of Eminence, Banaras Hindu University (BHU), for providing the research grant that made this work in the Varanasi tourism domain possible. We have used DeepSeek and ChatGPT LLMs to refine our methodology and enhance the fluency of the text in the article. We sincerely thank Dr. S. Suresh, NIT, Kurukshetra and Dr. Jagdeesan T., BHU, Varanasi who were Co-Principal Investigators of the TDR Project. Because of their cooperation and coordination during the project tenure, we could work on this task. We thank Research Assistants of the TDR Project Shreya Pandey, Bhaskar Singh, and Aman Gupta for assisting in the creation of the Hindi QA dataset. We thank Himesh, and Abhilasha Master's students, BHU for extending their support in compiling the Hindi QA dataset for Varanasi tourism. We also thank Iram Ali Ahmad, Supriya Chauhan, and Jyoti Kumari for helping in refining the Hindi QA dataset.

References

- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017.
- Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, 2021.
- Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), page 219–224, New York, NY, USA, 1961. Association for Computing Machinery. ISBN 9781450378727. doi:10.1145/1460690.1460714. URL <https://doi.org/10.1145/1460690.1460714>.
- William A Woods. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450, 1973.
- Alexander R Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. Template-based question generation from retrieved sentences for improved unsupervised question answering. *arXiv preprint arXiv:2004.11892*, 2020.
- Claude Sammut and Ranjan B Banerji. Learning concepts by asking questions. *Machine learning: An artificial intelligence approach*, 2:167–192, 1986.
- Tianyu Liu, Bingzhen Wei, Baobao Chang, and Zhifang Sui. Large-scale simple question generation by template-based seq2seq learning. In *Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6*, pages 75–87. Springer, 2018.
- Janell Straach and Klaus Truemper. Learning to ask relevant questions. *Artificial Intelligence*, 111(1-2):301–327, 1999.
- Sanda Harabagiu, Andrew Hickl, John Lehmann, and Dan Moldovan. Experiments with interactive question-answering. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*, pages 205–214, 2005.
- Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In *Human language technologies: The 2010 annual conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, 2010.
- Roger Levy and Galen Andrew. Tregex and tsurgeon: Tools for querying and manipulating tree data structures. In *LREC*, pages 2231–2234. Genoa, 2006.
- Kaustubh D Dhole and Christopher D Manning. Syn-qg: Syntactic and shallow semantic rules for question generation. *arXiv preprint arXiv:2004.08694*, 2020.
- Xuchen Yao and Yi Zhang. Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 68–75. Citeseer, 2010.
- Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pages 5608–5617, 2017.

- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020a.
- Minh-Nam Tran, Phu-Vinh Nguyen, Long Nguyen, and Dinh Dien. Vimedaqa: A vietnamese medical abstractive question-answering dataset and findings of large language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 356–364, 2024.
- Vikas Yadav, Hyuk joon Kwon, Vijay Srinivasan, and Hongxia Jin. Explicit over implicit: Explicit diversity conditions for effective question answer generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6876–6882, 2024.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 7871. Association for Computational Linguistics, 2020b.
- Amer Farea and Frank Emmert-Streib. Experimental design of extractive question-answering systems: Influence of error scores and answer length. *Journal of Artificial Intelligence Research*, 80:87–125, 2024.
- Abhishek Kumar Pandey and Sanjiban Sekhar Roy. Extractive question answering over ancient scriptures texts using generative ai and natural language processing techniques. *IEEE Access*, 2024.
- Saptarshi Sengupta, Connor Heaton, Shreya Ghosh, Wepeng Yin, Preslav Nakov, and Suhang Wang. Top-training: Target-oriented pretraining for medical extractive question answering. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7035–7054, 2025.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016a. Association for Computational Linguistics. doi:10.18653/v1/D16-1264. URL <https://aclanthology.org/D16-1264/>.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.
- Benno Kruit, Yiming Xu, and Jan-Christoph Kalo. Retrieval-based question answering with passage expansion using a knowledge graph. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14063–14072, 2024.
- Kalyani Roy, Vineeth Balapanuru, Tapas Nayak, and Pawan Goyal. Investigating the generative approach for question answering in e-commerce. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 210–216, 2022.
- Ali Al-Laith. Exploring the effectiveness of multilingual and generative large language models for question answering in financial texts. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 230–235, 2025.
- Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. Realtime qa: What’s the answer right now? *Advances in neural information processing systems*, 36:49025–49043, 2023.
- Adhitya Thirumala and Elisa Ferracane. Extractive question answering on queries in hindi and tamil. *arXiv preprint arXiv:2210.06356*, 2022.
- Dhiraj Amin, Sharvari Govilkar, and Sagar Kulkarni. Question answering using deep learning in low resource indian language marathi. *arXiv preprint arXiv:2309.15779*, 2023.

- Maithili Sabane, Onkar Litake, and Aman Chadha. Breaking language barriers: A question answering dataset for hindi and marathi. *arXiv preprint arXiv:2308.09862*, 2023.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016b.
- Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemaraj, Mayank Jobanputra, Raghavan Ak, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10: 145–162, 2022.
- Abhishek Kumar Singh, Rudra Murthy, Jaydeep Sen, Ashish Mittal, Ganesh Ramakrishnan, et al. Indic qa benchmark: A multilingual benchmark to evaluate question answering capability of llms for indic languages. *arXiv preprint arXiv:2407.13522*, 2024.
- Arpita Vats, Rahul Raja, Mrinal Mathur, Vinija Jain, and Aman Chadha. Multilingual state space models for structured question answering in indic languages. *arXiv preprint arXiv:2502.01673*, 2025.
- Yuan Sui. Question answering system based on tourism knowledge graph. In *Journal of Physics: Conference Series*, volume 1883, page 012064. IOP Publishing, 2021.
- Jiahui Li, Zhiyi Luo, Hongyun Huang, and Zuohua Ding. Towards knowledge-based tourism chinese question answering system. *Mathematics*, 10(4):664, 2022.
- Qikai Wei, Mingzhi Yang, Jinqiang Wang, Wenwei Mao, Jiabo Xu, and Huansheng Ning. Tourllm: Enhancing llms with tourism knowledge. *arXiv preprint arXiv:2407.12791*, 2024.
- Hoon-chul Kang, Myeong-Gyun Kang, and Jeong-Woo Jwa. Development of a tourism information qa service for the task-oriented chatbot service. *International Journal of Advanced Culture Technology*, 12(3):73–79, 2024.
- Sonia Meyer, Shreya Singh, Bertha Tam, Christopher Ton, and Angel Ren. A comparison of llm finetuning methods & evaluation metrics with travel chatbot use case. *arXiv preprint arXiv:2408.03562*, 2024.
- İSMAİL Kırtilı, BEYKAN Çizel, İsmail Uzut, and SERDAR Uzun. Bridging the gap: Fine-tuning artificial intelligence (ai) chatbots for tourism. pages 127–130, 2024.
- Anna Scius-Bertrand, Michael Jungo, Lars Vögtlin, Jean-Marc Spat, and Andreas Fischer. Zero-shot prompting and few-shot fine-tuning: Revisiting document image classification using large language models. In *International Conference on Pattern Recognition*, pages 152–166. Springer, 2025.
- Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, et al. Lora: Low-rank adaptation of large language models.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, 2020.
- Raviraj Joshi. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*, 2022.