Exploring Expected Answer Types for Effective Question Answering Systems for low resource language

Chindukuri Mallikarjuna
NIT-Tiruchirappalli,Tamilnadu,India
malli.chindukuri@gmail.com

Sangeetha Sivanesan
NIT-Tiruchirappalli,Tamilnadu,India
sangeetha@nitt.edu

Abstract

Question-answering (QA) systems play a pivotal role in natural language processing (NLP), powering applications such as search engines and virtual assistants by providing accurate responses to user queries. However, building effective QA systems for Dravidian languages, like Tamil, poses distinct challenges due to the scarcity of resources and the linguistic complexities inherent to these languages. This paper introduces a novel method to enhance QA accuracy by integrating answer-type features alongside traditional question and context inputs. We fine-tuned both mono- and multilingual pre-trained models on the Extended Chaii dataset, which comprises Tamil translations from the SQuAD dataset, as well as on the SQuAD-EAT-5000 dataset, consisting of English-language instances. Our experiments reveal that incorporating answer-type features significantly improves model performance compared to using only question and context inputs. Specifically, for the Extended Chaii dataset, the MuRIL model achieved the highest F1 score of 53.89, surpassing other pre-trained models, while RoBERTa outperformed BERT on the SQuAD-EAT-5000 dataset with a score of 82.07. This research advances QA systems for Dravidian languages and underscores the importance of integrating linguistic features for improved accuracy.

1 Introduction

A question answering (QA) system is a challenging endeavor in the field of NLP, aiming to enable computers to derive final answers by reasoning from the semantic information of natural language queries(Zhang et al., 2023). QA systems have become a cornerstone in NLP, finding applications in search engines, virtual assistants, and various other domains where accurate and efficient information retrieval is crucial. QA is a job in NLP and Information Retrieval that tries to automatically answer a human-posed question in natural language(Aroussi et al., 2016). These systems are designed to interpret and respond to user queries with precise answers, thereby enhancing the user experience and accessibility of information. While significant progress has been made in developing QA systems for widely spoken languages, creating effective QA systems for less-resourced languages, such as Dravidian languages, poses unique challenges. Tamil, a prominent Dravidian language, exemplifies these challenges due to its rich linguistic structure and limited availability of annotated datasets.

Developing QA systems for Tamil is particularly challenging because of the inherent linguistic complexities and the scarcity of resources(Antony and Paul, 2023). The traditional approach of relying solely on question and context inputs often falls short in addressing these challenges effectively. To overcome these obstacles, this paper proposes a novel approach that integrates answer-type features into the QA system, providing an additional layer of contextual understanding that enhances accuracy.

In this study, we fine-tune several pre-trained models, both mono-lingual and multi-lingual, using a Tamil dataset translated from the widely recognized Squad dataset. By incorporating answer-type features, we aim to improve the performance of these models in predicting accurate answers. Our experimental results demonstrate that models utilizing answer-type features significantly outperform those that rely solely on question and context inputs.

1.1 Motivation

LLMs require comprehensive contextual information to deliver precise answers. Similar to how the hints provided to students help them to accurately answer the questions posed by teachers, LLMs benefit from enriched inputs in fetching accurate answers. With this inspiration, we fine-tune multiple PTLMs with the question, context, and expected answer type as a clue to guide the proposed QA system toward generating more precise and relevant answers. This research contributes to the advancement of QA systems in low-resource Dravidian languages. Integrating linguistic features such as expected answer type from an EAT classification model improved the performance of the QA system. By addressing the limitations associated with resource scarcity and linguistic complexity, this study paves the way for more accurate and effective QA systems in under-resourced languages.

This research contributes to the advancement of QA systems for Dravidian languages by highlighting the importance of integrating linguistic features such as answer type for improved performance. By addressing the limitations associated with resource scarcity and linguistic complexity, this study paves the way for more accurate and effective QA systems for under-resourced languages.

2 Related study

In this section, we describe some related works regarding Question answering systems in the Tamil language. Our literature review revealed that while English has several benchmark QA datasets, models have shown better performance in English than in other high-resource languages. Our literature review revealed that few initiatives have been implemented for Tamil question-answering tasks. There are few QA datasets available for Indic languages(Namasivayam and Rajan, 2023). Some of these works have been referenced here. Karthik Kumar (Kumar et al., 2022) implemented mBERT over Google Extended Chaii dataset which is a multi-lingual question-answering dataset consisting of a total of 1114 records of multiple languages like Hindi and Tamil. The authors implemented mBERT with a constrative training approach. Authors evaluated the performance of mBERT using the Jaccard similarity metric. Recent experiments (Thirumala and Ferracane, 2022) have investigated the application of transformer models pre-trained on multiple languages, specifically focusing on Hindi and Tamil question-answering (QA). These studies have demonstrated enhanced performance in extractive QA tasks. Ram Vignesh (Namasivayam and Rajan, 2023) implemented four approaches to solve answer prediction over the Chaii multi-linguaval dataset consisting of 746 data instances for Hindi and 368 data instances for Tamil. This work is implemented with fine-tuning of mBERT, XML-RoBERTa and MuRIL multi-lingual pre-trained models and evaluated the performance using Exact match and F1-score along with Jaccard similarity metrics. Rajendran et.al (Sankaravelayuthan et al., 2019) designed QA system for the tourism domain. This QA system uses dependency information while fetching the answers to the user queries. Srivatsun et.al (Srivatsun et al., 2022), implement the mBERT model over Extended Chaii dataset and evaluate the performance of the model using the BELU metric. The survey revealed that, despite the availability of several benchmark QA datasets for English, models perform well in English but not in other high-resource languages. There is a notable lack of benchmark datasets for Indic languages, which hinders the development of efficient QA systems. In addition, from the literature we found that QA system provides a precise answer to the user question based on having prior knowledge about the expected answer type. It also leads to improve the performance of the QA System(Mallikarjuna and Sivanesan, 2022). So in this research, by incorporating additional information alongside the usual input parameters to the QA system, we are aiming to build an efficient QA system for the Tamil language even with limited data.

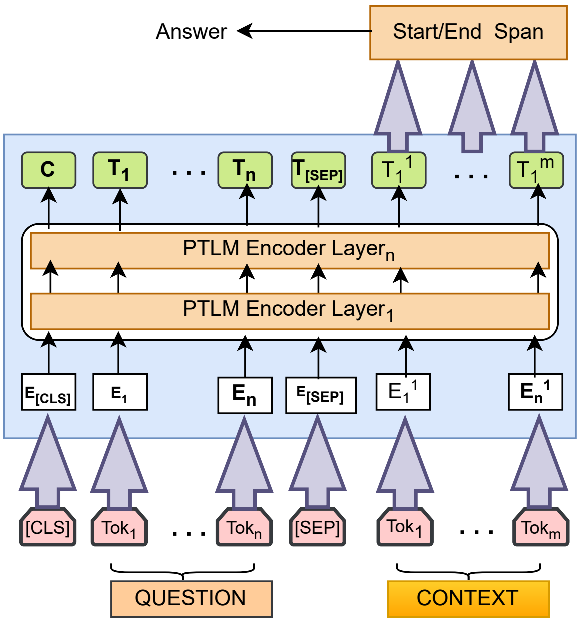


Figure 1: Architecture of traditional Question Answering System.

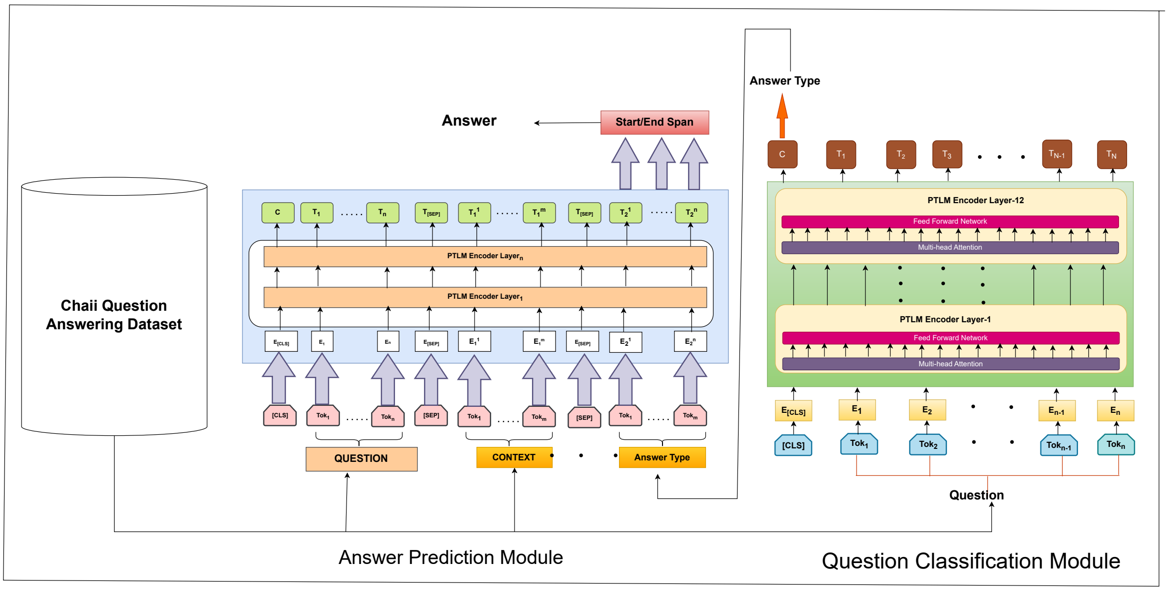
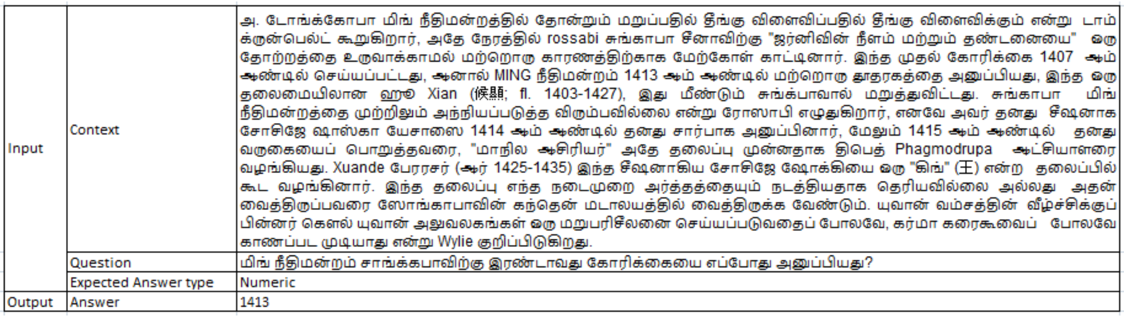


Figure 2: Architecture of Proposed Question Answering System.



Input	Context	அ. டோங்க்கோபா மிங் நீதிமன்றத்தில் தோன்றும் மறுப்பதில் தீங்கு விளைவிப்பதில் தீங்கு விளைவிக்கும் என்று டாம் க்றுன்பெல்ட் கூறுகிறார், அதே நேரத்தில் rossabi சுங்காபா சீனாவிற்கு "ஜர்னிவின் நீளம் மற்றும் தண்டனையை" ஒரு தோற்றத்தை உருவாக்காமல் மற்றொரு காரணத்திற்காக மேற்கோள் காட்டினார். இந்த முதல் கோரிக்கை 1407 ஆம் ஆண்டில் செய்யப்பட்டது, ஆனால் MING நீதிமன்றம் 1413 ஆம் ஆண்டில் மற்றொரு தூதரகத்தை அனுப்பியது, இந்த ஒரு தலைமையிலான ஹூ Xian (候顯; fl. 1403-1427), இது மீண்டும் சுங்க்பாவால் மறுத்துவிட்டது. சுங்காபா மிங் நீதிமன்றத்தை முற்றிலும் அந்நியப்படுத்த விரும்பவில்லை என்று ரோஸாபி எழுதுகிறார், எனவே அவர் தனது சீஷனாக சோசிஜே ஷாஸ்கா யேசாஸை 1414 ஆம் ஆண்டில் தனது சார்பாக அனுப்பினார், மேலும் 1415 ஆம் ஆண்டில் தனது வருகையைப் பொறுத்தவரை, "மாநில ஆசிரியர்" அதே தலைப்பு முன்னதாக திபெத் Phagmodrupa ஆட்சியாளரை வழங்கியது. Xuande பேரரசர் (ஆர் 1425-1435) இந்த சீஷனாகிய சோசிஜே ஷோக்கியை ஒரு "கிங்" (王) என்ற தலைப்பில் கூட வழங்கினார். இந்த தலைப்பு எந்த நடைமுறை அர்த்தத்தையும் நடத்தியதாக தெரியவில்லை அல்லது அதன் வைத்திருப்பவரை ஸோங்காபாவின் கந்தென் மடாலயத்தில் வைத்திருக்க வேண்டும். யுவான் வம்சத்தின் வீழ்ச்சிக்குப் பின்னர் கௌல் யுவான் அலுவலகங்கள் ஒரு மறுபரிசீலனை செய்யப்படுவதைப் போலவே, கர்மா கரைகூவைப் போலவே காணப்பட முடியாது என்று Wylie குறிப்பிடுகிறது.
	Question	மிங் நீதிமன்றம் சாங்க்கபாவிற்கு இரண்டாவது கோரிக்கையை எப்போது அனுப்பியது?
	Expected Answer type	Numeric
Output	Answer	1413

Figure 3: Sample input and output of Proposed QA System architecture.

3 Proposed Method

The proposed approach in this study aims to address the challenges of developing effective QA systems for Dravidian languages, specifically Tamil, by incorporating additional linguistic features into the model training process. The foundation of this approach lies in the creation of a Tamil dataset translated from the well-known Squad dataset. This translation ensures that the models are trained on high-quality, contextually relevant data, which is crucial for handling the unique linguistic characteristics of Tamil. The key innovation in the proposed approach is the inclusion of answer-type features alongside the traditional question and context inputs. Answer types classify the expected responses such as person, location, or date, providing the model with specific guidance that enhances its ability to generate accurate answers. This additional feature helps the model to better understand the nature of the question and focus its prediction efforts more effectively. Fig 2 represents the architecture of the proposed QA system. The proposed QA system architecture consists of two modules:

1. Question Classification module.

2. Answer Prediction module.

We describe each module of the proposed QA system in the following sections.

1. **Question Classification Module** Through this module, all questions in the QA dataset are classified based on the expected answer type. This involves categorizing questions into six types according to the Li and Roth Taxonomy: Abbreviation, Description, Entity, Human, Location, and Numeric (Li and Roth, 2006). This classification is a critical component of the QA system, as it guides the subsequent answer extraction process. The system

uses a transfer learning model fine-tuned for this question classification task, ensuring high accuracy in distinguishing between the different expected answer types.

The question classification process involves predicting the expected answer type for a given question. This process involves a series of steps. The question is first tokenized and embedded using a pre-trained language model. The model's output is then passed through a softmax layer to classify the question into one of the six expected answer types. Below is the mathematical representation of the question classification process as per expected answer type.

(a) **Input Representation** The input question $Q = [q_1, q_2, \ldots, q_{n_q}]$ consists of a sequence of tokens, where q_i represents the i-th token of the question, and n_q is the number of tokens. Each token q_i is mapped to a word embedding vector $\mathbf{e}_i \in \mathbb{R}^d$, where d is the dimension of the embedding space. This results in the embedding matrix for the question:

$$\mathbf{E}_Q = [\mathbf{e}_{q_1}, \mathbf{e}_{q_2}, \ldots, \mathbf{e}_{q_{n_q}}] \quad (1)$$

where $\mathbf{E}_Q \in \mathbb{R}^{n_q \times d}$ is the matrix of token embeddings.

(b) **Embedding:** The question tokens are embedded using pre-trained language models, producing contextualized embeddings as output after being forwarded through a series of encoder layers.

$$\mathbf{H}_Q = PTLM[\mathbf{E}_Q] \quad (2)$$

$$\mathbf{H}_Q = PTLM[\mathbf{h}_{q_1}, \mathbf{h}_{q_2}, \ldots, \mathbf{h}_{q_{n_q}}] \quad (3)$$

where $\mathbf{H}_Q \in \mathbb{R}^{n_q \times d}$ are the hidden states for the question, and d is the dimensionality of the embeddings.

(c) **Calculating probabilities** The [CLS] token's hidden state, $\mathbf{h}_{[CLS]}$, is used to classify the question into one of the six categories (which also serve as the expected answer types):

$$\mathbf{y}_{QC\_EAT} = \text{softmax}(W \cdot \mathbf{h}_{[CLS]} + b) \quad (4)$$

where:

- $W \in \mathbb{R}^{d \times 6}$ is the weight matrix.
- b is the bias vector.
- $\mathbf{y}_{QC\_EAT} \in \mathbb{R}^6$ represents the probability distribution over the six categories.

(d) **Predicting Question category:** Here, the category with maximum probability is predicted as the question category(e.g., “Person" “Location" etc.). This is mathematically represented as follows.

$$\text{Que\_Cat} = \arg\max(\mathbf{y}_{QC\_EAT}) \quad (5)$$

2. **Answer Prediction module** The goal of this module is to extract the relevant span from the context that best answers the given question. After the classification of each question in the QA dataset, a new feature called “Expected Answer Type" is added to the existing features in the QA dataset. This feature maps each question with the “Answer Type" value which is the outcome of the question classification module. After adding the additional feature to the existing QA dataset, both monolingual and multilingual pre-trained models are fine-tuned using the input data, which includes the question, context, and expected answer type for extracting the answers from the given context. In the following section, the series of steps involved in the answer prediction module is described.

Input Representation: Let the context be represented by a sequence of tokens: $C = [c_1, c_2, \ldots, c_{n_c}]$ where c_i is the i-th token in the context, and n_c is the number of tokens. The question is represented by $Q = [q_1, q_2, \ldots, q_{n_q}]$, and the unified label from the previous module is Expected Answer Type.

Concatenation and Embedding: The input to the model is the concatenation of the question, expected answer type, and the context:
Input $=$ [[CLS], $q\_1, \ldots, q\_n_q$, [SEP], Que_Cat, [SEP], $c\_1, \ldots, c\_n\_c$, [SEP]]

After passing this input through PTLM, we obtain hidden states for each token:

$$\mathbf{H} = [\mathbf{h}_{[CLS]}, \mathbf{h}_{\text{Que\_Cat}}, \mathbf{h}_{q_1}, \ldots, \mathbf{h}_{c_{n_c}}] \quad (6)$$

Answer Span Prediction (Start and End Positions): A linear layer predicts the start of the answer within the context:

$$P_s(i) = \text{softmax}(W_s \cdot \mathbf{h}_{c_i} + b_s) \quad (7)$$

where $W_s \in \mathbb{R}^{d\times 1}$, and $P_s(i)$ represents the probability that token c_i is the start of the answer. Similarly, the end position of the answer is predicted by:

$$P_e(i) = \text{softmax}(W_e \cdot \mathbf{h}_{c_i} + b_e) \quad (8)$$

where $P_e(i)$ represents the probability that token c_i is the end of the answer.

Incorporating expected answer type: The class label is used to refine the answer span prediction:

$$P_s(i) = \text{softmax}(\mathbf{h}^T_{\text{Category\_EAT}} W_s \mathbf{h}_{c_i}) \quad (9)$$

Answer Extraction: The final answer is extracted based on the predicted start and end positions:

$$\text{Start Index} = \arg\max(P_s) \quad (10)$$

$$\text{End Index} = \arg\max(P_e) \quad (11)$$

Figure 3 presents the sample input and output of the proposed QA system.

4 Experiments

4.1 Baseline Models

- MuRIL
  MuRIL (Khanuja et al., 2021), or Multilingual Representations for Indian Languages, is a pre-trained model developed by Google Research to understand and process text in multiple Indian languages. It's an extension of the popular BERT architecture, trained on a diverse dataset covering 17 Indian languages, enabling it to comprehend nuances and context specific to the Indian subcontinent. MuRIL empowers NLP tasks like sentiment analysis, language understanding, and text classification across a wide range of Indian languages, thereby fostering better accessibility and engagement in linguistic diversity.

- mBERT
  Google Multilingual BERT (Devlin et al., 2018) (Bidirectional Encoder Representations from Transformers) is a pre-trained language model developed by Google Research to understand and process text in multiple languages simultaneously. It builds upon the original BERT architecture, which is renowned for its effectiveness in NLP tasks. Multilingual BERT is trained on a vast and diverse dataset covering over a hundred languages, allowing it to capture cross-lingual patterns and representations. By learning shared representations across languages, Multilingual BERT enables effective transfer learning, where knowledge gained from one language can be applied to others, making it a valuable tool for multilingual NLP tasks such as translation, sentiment analysis, and named entity recognition.

- XLM-RoBERTa
  XLM-RoBERTa (Conneau et al., 2019), short for Cross-lingual Language Model - RoBERTa, is a pre-trained model developed by Facebook AI. It combines two powerful approaches: RoBERTa, an extension of BERT, and cross-lingual learning, to create a robust language model capable of understanding and generating text across multiple languages. XLM-RoBERTa is trained on a vast corpus of text from over 100 languages, allowing it to learn shared representations across different languages. This enables effective transfer learning, where knowledge gained from one language can be applied to others, making it useful for a wide range of multilingual natural NLP tasks such as translation, sentiment analysis, and language understanding.

- Tamil-BERT
  Tamil-BERT (Joshi, 2022) is a pre-trained language model developed specifically for the Tamil language by researchers or developers interested in fostering NLP tasks in Tamil. Like BERT, it is based on the Transformer architecture and is pre-trained on a large corpus of Tamil text, enabling it to learn contextual representations of Tamil words and sentences. This allows Tamil-BERT to effectively handle various NLP tasks, including sentiment analysis, text classification, and named entity recognition, in Tamil text.

4.2 Experimental Settings

We fine-tuned multiple mono and multilingual pre-trained models over multiple datasets. These are described as follows.

- **Extended Chaii Dataset**
  The Extended Chaii dataset contains questions and contexts in the Tamil language, sourced from the original Extended Chaii dataset [1] introduced during the Challenging Adversarial Hindi and Indic QA competition. In addition, it includes Tamil-translated questions and contexts from the SQuAD dataset. This dataset is specifically designed for question-answering tasks in low-resource Indic languages like Tamil, addressing the significant gap in resources for Tamil language processing. It focuses on enhancing machine comprehension of Tamil text and includes passages along with corresponding questions, answers, and expected answer types, making it well-suited for extractive question-answering tasks. Table 1 presents the statistics of the Extended Extended Chaii dataset.

- **SQuAD-EAT-5000 dataset**
  The SQuAD-EAT-5000 dataset contains 5,000 instances sourced from the SQuAD dataset[2], where all questions and contexts are provided in English. In addition to the questions and contexts, each question in the SQuAD-EAT-5000 dataset is also mapped to its corresponding expected answer type. Table 1 presents the statistics of the SQuAD-EAT-5000 dataset.

Table 1: Datasets Statistics

Dataset	Number of Instances		
	Train	Validation	Test
Extended Chaii Dataset	2855	460	250
SQuAD-EAT-5000	4000	600	400

We fine-tune the mono and multi-lingual pre-trained models with optimal hyperparameters. Table 2 presents the hyperparameters of different mono and multi-lingual pre-trained models for the translated Tamil Squad dataset. Table 3 presents the transfer learning models parameters and their corresponding sizes.

Table 2: Hyperparameters

Model	LR	BS	ML
mBERT	2.00E-05	16	512
XLM-RoBERTa	5.00E-05	8	512
Tamil-BERT	5.00E-05	16	512
MuRIL	2.00E-05	16	512

Table 3: PTLMs Statistics

Model	Size(In MB)	Parameters
Bert-base	440 MB	110 Million
RoBERTa-base	499 MB	355 Million
mBERT	672 MB	110 Million
MuRIL	953 MB	270 Million
XLM-RoBERTa	1120 MB	550 Million
Tamil-BERT	951 MB	110 Million

4.3 Evaluation Metrics

In our study, we selected Exact Match (EM) and F1 Score as the evaluation metrics for the extractive QA system because they directly assess the alignment between the predicted answer and the ground truth. EM guarantees the precise accuracy of the predicted answer, which is essential in tasks requiring high precision, such as fact-based QA. On the other hand, F1 Score strikes a balance between precision and recall, making it suitable for handling partially correct answers. This approach is particularly effective for evaluating extractive QA in low-resource scenarios where answer phrasing may vary.

4.4 Experimental Results

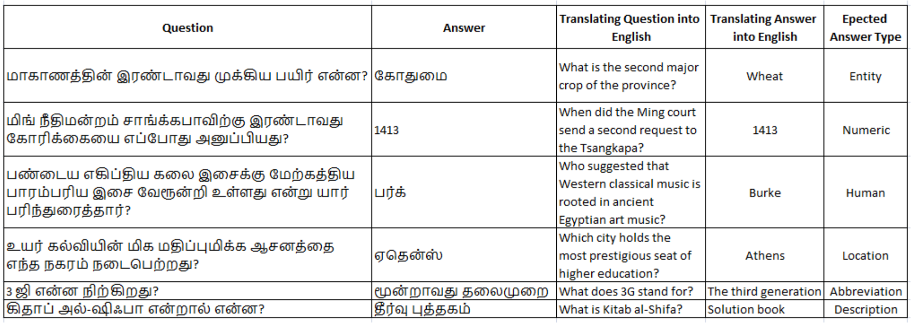
Here, we report dataset-wise experimental results for both monolingual and multilingual transfer learning models.

1. Extended Chaii Dataset
   We fine-tuned several mono and multi-lingual transfer learning models against the Translated Tamil question-answering dataset. We fine-tuned these pre-trained models in two scenarios. We evaluated the performance of the models against the above-mentioned datasets in terms of Average Exact Match score and F1 score. For this, we used Squad version2 metric [3] library in our experiments. Initially, we fine-tuned the models with Question and Context as input parameters and tried to predict the answers for the test data. Later in the proposed method, we fed "expected answer type" as an additional parameter along with question and context to the model during its

[1] https://www.kaggle.com/datasets/msafi04/squad-translated-to-tamil-for-Extended Chaii dataset/data

[2] https://rajpurkar.github.io/SQuAD-explorer/

[3] https://huggingface.co/spaces/evaluate-metric/squad_v2



Question	Answer	Translating Question into English	Translating Answer into English	Epected Answer Type
மாகாணத்தின் இரண்டாவது முக்கிய பயிர் என்ன?	கோதுமை	What is the second major crop of the province?	Wheat	Entity
மிங் நீதிமன்றம் சாங்க்கபாவிற்கு இரண்டாவது கோரிக்கையை எப்போது அனுப்பியது?	1413	When did the Ming court send a second request to the Tsangkapa?	1413	Numeric
பண்டைய எகிப்திய கலை இசைக்கு மேற்கத்திய பாரம்பரிய இசை வேரூன்றி உள்ளது என்று யார் பரிந்துரைத்தார்?	பர்க்	Who suggested that Western classical music is rooted in ancient Egyptian art music?	Burke	Human
உயர் கல்வியின் மிக மதிப்புமிக்க ஆசனத்தை எந்த நகரம் நடைபெற்றது?	ஏதென்ஸ்	Which city holds the most prestigious seat of higher education?	Athens	Location
3 ஜி என்ன நிற்கிறது?	மூன்றாவது தலைமுறை	What does 3G stand for?	The third generation	Abbreviation
கிதாப் அல்-ஷிஃபா என்றால் என்ன?	தீர்வு புத்தகம்	What is Kitab al-Shifa?	Solution book	Description

Figure 4: Mapping questions with expected answer type in Extended Chaii Dataset.

Table 4: Performance of mono and multilingual PTLMs against Extended Chaii Dataset

Model	Without EAT		With EAT	
	Avg Exact Match	F1-Score	Avg Exact Match	F1-Score
mBERT	14.73	46.82	38.245	49.43
XLM-RoBERTa	29.47	43.43	34.73	49.06
MuRIL	9.47	48.04	17.19	53.89
Tamil-BERT	17.89	41.24	21.05	45.84

fine-tuning process and tried to predict the answers for the test questions. We evaluate the models performance in terms of Exact match and F1-score.

Table 4 presents the fine-tuning results of the mono and multi-lingual pre-trained models for both scenarios. The models assessed include mBERT, XLM-RoBERTa, MuRIL, and Tamil-BERT, with their Average Exact Match and F1-Score metrics reported. The performance metrics of transfer learning models on the Translated Tamil Squad dataset(Extended Chaii Dataset) without Expected Answer Type (EAT) show varying strengths. mBERT achieves an Exact Match of 14.73 and an F1-Score of 46.83, indicating a substantial drop in precision without EAT but strong contextual understanding. XLM-RoBERTa scores 29.47 in Exact Match and 43.43 in F1-Score, demonstrating higher precision in exact answers compared to mBERT, albeit with slightly lower contextual comprehension. MuRIL exhibits the lowest Exact Match at 9.47 but the highest F1-Score at 48.04, suggesting superior contextual performance despite significant challenges in exact answer retrieval without EAT. Tamil-BERT records an Exact Match of 17.89 and an F1-Score of 41.24, indicating moderate precision and contextual comprehension, better in precision than MuRIL but lower in overall understanding. Overall, MuRIL excels in F1-Score, highlighting its contextual strength, while XLM-RoBERTa leads in Exact Match accuracy, and mBERT maintains a high F1-Score with balanced performance. Tamil-BERT shows moderate performance, suggesting the need for further optimization.

Table 4 also presents the performance metrics of various mono and multilingual transfer learning models when evaluated on the Translated Tamil Squad question answering dataset with Expected Answer Type(EAT) as an additional feature in addition to the "Question" and "context" inputs. The models under consideration are mBERT, XLM-RoBERTa, MuRIL, and Tamil-BERT, with their Exact Match and F1-Score metrics reported. The performance evaluation of various transfer learning models on the Translated Tamil Squad question answering dataset, with the inclusion of Expected Answer Type (EAT), reveals distinct strengths and weaknesses across the models. mBERT exhibits the highest Exact Match score of 38.245, indicating superior precision in exact answer retrieval, complemented by a robust F1-Score of 49.43, demonstrating balanced contextual understanding. XLM-RoBERTa, while achieving a slightly lower Exact Match score of

Table 5: Performance of transfer learning models against SQuAD-EAT-5000 dataset

Model	**Without EAT**		**With EAT**	
	Avg Exact Match	**F1-Score**	**Avg Exact Match**	**F1-Score**
BERT-base	32.25	76.11	35.5	77.87
RoBERTa-base	56	81	61.05	82.07

34.73, maintains a strong F1-Score of 49.06, reflecting its efficacy in nuanced semantic comprehension. MuRIL, tailored for Indian languages, leads in contextual performance with an F1-Score of 53.89, yet its Exact Match score of 17.19 underscores a significant gap in exactitude, suggesting a trade-off between context capture and precise answer generation. Tamil-BERT, despite being fine-tuned for the Tamil language, records a moderate Exact Match score of 21.05 and an F1-Score of 45.84, indicating its relatively limited effectiveness in both exact answer precision and contextual grasp. Collectively, these results underscore mBERT's and XLM-RoBERTa's robustness in multilingual scenarios, MuRIL's contextual superiority with precision trade-offs, and Tamil-BERT's potential for further optimization in Tamil-specific tasks.

2. SQuAD-EAT-5000 Dataset

   Table 5 presents fine-tuning results of BERT-base(Devlin et al., 2019) and RoBERTa-base(Liu et al., 2019) models against SQuAD-EAT-5000 Dataset. These results prove that both BERT-base and RoBERTa-base models perform better with Entity-Aware Training (EAT). For BERT-base, the Average Exact Match score increases from 32.25 to 35.5, and the F1-Score rises from 76.11 to 77.87. Similarly, for RoBERTa-base, the Exact Match score improves from 56 to 61.05, and the F1-Score goes up from 81 to 82.07, demonstrating enhanced performance with EAT for both models.

All things considered, the test of transfer learning models' performance using the Extended Chaii Dataset and SQuAD-EAT-5000 datasets shows that adding Expected Answer Type (EAT) greatly improves F1-score and Average Exact Match scores. All models show a decrease in F1-score and Average Exact Match in the absence of EAT. This study also demonstrates that adding the EAT feature will enhance the QA system's performance for both low-resource language like Tamil and rich-resource languages like English QA datasets.

5 Conclusion

In conclusion, this study addresses the challenges of developing effective question-answering (QA) systems for Dravidian languages, specifically Tamil, by introducing a novel approach that incorporates answer-type features alongside traditional question and context inputs. By fine-tuning both mono- and multi-lingual pre-trained models on a Extended Chaii Tamil QA dataset derived from the Squad dataset, the research demonstrates that the inclusion of answer-type features significantly enhances the accuracy of QA systems. Notably, the MURIL model achieved the highest F1 score of 53.89, outperforming other evaluated models. Similarly, over SQuAD-EAT-5000 dataset, RoBERTa model achieved highest F1-score with the inclusion of additional expected answer type information compared with BERT. This work underscores the importance of integrating linguistic features to improve the performance of QA systems for under-resourced languages, thereby contributing to the broader advancement of NLP technologies for Dravidian languages.

References

Betina Antony and NR Rejin Paul. 2023. Question answering system for tamil using deep learning. In *Speech and Language Technologies for Low-Resource Languages*, pages 244–252, Cham. Springer International Publishing.

Saïd Alami Aroussi, Nfaoui El Habib, and Omar El Beqqali. 2016. Improving question answering systems by using the explicit semantic analysis method. In *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–6. IEEE.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Gokul Karthik Kumar, Abhishek Singh Gehlot, Sahal Shaji Mullappilly, and Karthik Nandakumar. 2022. Mucot: Multilingual contrastive training for question-answering in low-resource languages. *arXiv preprint arXiv:2204.05814*.

Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chindukuri Mallikarjuna and Sangeetha Sivanesan. 2022. Question classification using limited labelled data. *Information Processing & Management*, 59(6):103094.

Ram Vignesh Namasivayam and Manjusha Rajan. 2023. Answer prediction for questions from tamil and hindi passages. *Procedia Computer Science*, 218:1985–1993.

Rajendran Sankaravelayuthan, M Anandkumar, V Dhanalakshmi, and SN Mohan Raj. 2019. A parser for question-answer system for tamil. *QA System Using DL*, 229:230.

G Srivatsun, S Thivaharan, Bharath Kumaar KS, and S Sudharsan. 2022. Machine comprehension system in tamil and english based on bert. In *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 847–854. IEEE.

Adhitya Thirumala and Elisa Ferracane. 2022. Extractive question answering on queries in hindi and tamil. *arXiv preprint arXiv:2210.06356*.

Jiahao Zhang, Bo Huang, Hamido Fujita, Guohui Zeng, and Jin Liu. 2023. Feqa: Fusion and enhancement of multi-source knowledge on question answering. *Expert Systems with Applications*, 227:120286.