EXPLOITING ADAPTERS FOR QUESTION GENERATION FROM TAMIL TEXT IN A ZERO-RESOURCE SETTING

Shanmukanathan Purusanth

(209367E)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

October 2022

EXPLOITING ADAPTERS FOR QUESTION GENERATION FROM TAMIL TEXT IN A ZERO-RESOURCE SETTING

Shanmukanathan Purusanth

(209367E)

This dissertation submitted in partial fulfillment of the requirements for the Degree of MSc in Computer Science specializing in Data Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

October 2022

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgment, any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the University of Moratuwa, the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

I certify that the declaration above by the candidate is true to the best of my knowledge and that this report is acceptable for the evaluation of a Master's thesis.

Name of the Supervisor: Dr. Surangika Ranathunga

Signature of the supervisor: Date

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to Dr. Surangika Ranathunga for guiding me to initiate, finding a great research topic to conduct the research, and for continuing support and encouragement. Her supervision greatly helped me in setting goals and engaging in the research.

I would like to express my greatest gratitude to the Department of Computer Science and Engineering, the University of Moratuwa for providing the support to overcome this effort. Last but not least, my heartfelt gratitude goes to my parents and friends who supported me throughout this effort.

ABSTRACT

Automatic Question Generation focuses on generating questions from a span of text is a significant problem in Natural Language Processing (NLP). Question generation in low-resource languages is under-explored compared to high-resource languages. In the earlier work, all the parameters of a pre-trained multilingual language model were fine-tuned to perform a zero-shot question generation and other sequence-to-sequence (S2S) generation tasks. However, such full model fine-tuning is not computationally efficient. Recent research introduced a neural module called adapter to each Transformer layer of a pre-trained language model and fine-tuned only the adapter parameters to mitigate this issue. In this study, we explored single task adapter and adapter fusion on the pre-trained multilingual model mBART to generate questions from Tamil text. Our best model produced a Rough-1 (F1) score of 16.9. Furthermore, we obtained a similar result with two variants of adapters called Houlsby adapter [1] and Pfeifer adapter [1], which resemble the result of adapters for other S2S tasks[2].

Keywords: Automatic Question Generation, Pre-trained language models, Adapters, Tamil.

TABLE OF CONTENTS

Declaration of the candidate and the supervisor I
Acknowledgements II
Abstract III
Table of content IV
List of Figures VI
List of Tables VII
List of Abbreviation VIII
1. Introduction 1
1.1 Background 1
1.2 Research Problem 1
1.3 Research Questions 2
1.4 Objective 2
1.5 Thesis Structure 3
2. Literature Review 4
2.1 Overview 4
2.2 Datasets 4
2.3 Rule-based approach 4
2.4 Machine Learning-based approach 5
2.4.1 Single-hop question generation 6
2.4.2 Multi-hop question generation 8
2.5 Pre-trained sequence-to-sequence modes 8
2.5.1 Adapters and Adapter Fusion 9
2.6 Summary 10
3. Methodology 11
3.1 Experimental setups 11
3.2 Baseline 12
3.2.1 Full model fine-tuning(mBART50) 12
3.2.2 Language model fine-tuning(T5) 12
3.2.3 Freeze token embedding and position of mBART50 12
3.2.4 Zero-shot questions generator 12
3.2.5 Multiple adapters for cross-lingual transfer learning 12
3.2.6 Prefix tuning 13
3.3.1 Proposed Method 13
3.3.1 Adapter fine-tuning with Houlsby adapter 13

3.3.2 Adapter fine-tuning with Pfeiffer adapter 13
3.3.3 Adapter Fusion 13
4. Evaluation Results 14
4.1 Evaluation metrics 14
5.1 Datasets 14
5.2 Results 15
6. Conclusion and Future work 18
Reference List 19

LIST OF FIGURES

Figure	description	Page
Figure 2.4.1	BERT-QG architecture	7
Figure 2.5.1	Houlsby adapter	9
Figure 2.5.2	Pfeifer adapter	9
Figure 2.5.3	Architecture of Adapter fusion inside a Transformer layer	11

LIST OF TABLES

Table	Description	Page
Table 1.1	Example of a question generated by MulQG	1
Table 2.4.1	Distribution of question words in SQuRD and MARCO dataset	7
Table 3.1.1	An example of formatted input and its formatted target for mBART	11
Table 3.1.2	An example of formatted input and its formatted target for mT5	11
Table 4.2.1	An example question form the test dataset	15
Table 4.2.2	Average lengths of context, question, answer	15
Table 4.3.1	Models and their parameters and trained epochs	16
Table 4.3.2	Result	17

LIST OF ABBREVIATION

Abbreviation	Description
S2S	sequence-to-sequence
SOTA	state-of-the-art
POST	part-of-speech tagging
NLP	Natural language processing

1 CHAPTER 1: INTRODUCTION

1.1 Background

Automated question generation from a given span of text is helpful in the automation of reading comprehension assessment, intelligent tutoring systems, chat-bot applications, dataset development for question answering, and system authentication [2], [3], [4]. For example, in an authentication system, rather than using a fixed set of questions, generating questions from a user-given context is more secure since little or no prior knowledge of the user's context makes it hard for a hacker to answer those questions [4]. Table 1.1 shows an example question generated by humans and a MulQG [5] (discussed later) given the context and answer as input.

Context: **Paragraph A**: House of Many Ways is a young adult fantasy novel written by Diana Wynne Jones. The story is set in the same world as "Howl's Moving Castle" and "Castle in the Air". **Paragraph B**: Howl's Moving Castle is a fantasy novel by British author Diana Wynne Jones. ... In 2004 it was adapted as an animated film of the same name, which was nominated for the academy award for the best-animated feature.
Answer: academy award for best-animated feature
The question generated by MulQG: what award was the author of the book house of many ways nominated for?
Human-generated question: house of many ways is a young adult fantasy novel set in the same world as a novel that was adapted as an animated film of the same name and nominated for what?

Table 1.1: Example question generated by MulQG

Source: [5]

1.2 Research problem

Automated question generation is not newborn research in Natural language processing (NLP). Initial work of Deep Learning based approaches used RNN to generate questions

[6]. Then, They were improved with an attention mechanism, copying mechanism, answer embedding, POS tag embedding, and NER embedding [6]. Although these techniques improved the performance, the complexity of the model increased drastically [6]. The major limitation of RNN-based models is that they cannot process a long context even though theoretically they can. On the other hand, pre-trained Transformer models provided outcomes comparable to sequence-to-sequence (S2S) models despite being a single model [6].

There is many research focused on question generation for high resource languages, but a limited number of works focused on generating questions in low-resources languages. The lack of annotated and unlabelled data is a remarkable challenge in developing an NLP system for low-resource languages. To the best of our knowledge, only Vignesh et al [7] focused on generating questions in Tamil. Their approach tagged each word in the sentence into a predefined set of tags and replaced a word with a specific tag with a corresponding question word to generate the question. However, this method of generating questions is confined to a specified kind of questions that are easy to answer and are not fully automated. Hence, it is essential to see how pre-trained Transformer models can be tailored for low resource languages. Many research has studied S2S pre-trained language models such as T5 [7], and MT5 [8], and encoder-only models such as BERT [9]. However, to the best of our knowledge, no study has used sequence-to-sequence models such as mBART for question generation.

1.3 Research Questions

The main research question in this research is formulated as follows: **Under limited computational resources, how can we fine-tune a pre-trained multi-lingual model for question generation?**

1.4 Objective

The objective of this research is to exploit adapters to fine-tune a pre-trained language model in a limited resource setting to generate questions from Tamil text.

1.5 Thesis Structure

The rest of the thesis contains six chapters. A brief description of the related work done previously is discussed in Chapter 2. Next, Chapter 03 will explain the methodology proposed in this research. Next, Chapter 04 will discuss the evaluation methods and findings from our research. Finally, Chapter 05 will discuss the conclusion and future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview

Prior work of automatic question generation can be categorized into rule-based and Machine Learning. Machine Learning-based approaches can further be categorized into single-hop question generation and multi-hop question generation. In this chapter, we will discuss the prior work in question generation using these techniques, Datasets used in previous studies, and S2S pre-trained language models.

2.2 Datasets

Many large-scale question-answering datasets are available, which can be repurposed for question generation in English [6]. SQuAD [11], RACE [2], and HotpotQA [5] are some of such question-answering datasets repurposed for question generation. The SQuAD is made up of 100,000 questions annotated by crowd workers from Wikipedia articles, where answer to each question is a section of the text from the article. HotpotQA is also generated from the Wikipedia articles; however, it requires complex reasoning to answer or generate the question. On the other hand, RACE is collected from real exams.

2.3 Rule-based Approaches

The earlier studies of question generation used rule-based approaches to generate questions. While rule-based approaches are labor-intensive, domain-specific and limited to a few styles of questions, they have strengths over the Neural Network models since they are easy to interpret and allow the developer to have greater control over the model behavior. Furthermore, they typically require fewer data to generate questions than the complex Neural Network to achieve the same level of performance [10]. On the other hand, the usefulness of this approach depends on the quality of the rules, which is often dependent on extensive knowledge about the domain.

Vignesh et al. [10] tagged each word in the sentence into a set of tags and replaced words having a specific tag with their corresponding question words to generate a question. This

is the only study that investigates question generation in Tamil to the best of our knowledge. Since it replaces a single word with a single question word, it produces syntactically and semantically correct questions. However, a question generated from a single sentence is easy to answer and limits the real word application like reading comprehension assessments where it is required to generate questions that require complex reasoning to answer. Dhole et al. [13] developed the Syng-QG model, where the raw declarative sentence is transformed into a question-answer pair using shallow semantic parsing, transparent syntactic rules, and lexical resources. PropBank argument descriptions and VerbNet state predicates to exploit shallow semantics and custom rule to generate questions, enabling them to generate a wide range of questions; this is not the case in [10]. Moreover, questions generated by Syng-QG are not suitable for applications that require complex reasoning to answer, as in [10]. However, questions generated by the approaches proposed in Vignesh et al. [10], and Dhole et al. [13] can be used to augment the question answering datasets because generated questions are syntactically, and semantically correct.

2.4 Machine Learning Approaches

Neural Network models can be trained end-to-end to generate questions [13]. The choice of Neural Network architecture should have the capacity to process sequence data because this is a S2S generation problem. The typical choices of building blocks are RNN [13] and Transformers [14]. In the earlier stage of neural question generation, two modules of the RNN networks work together to perform an end-to-end question generation. These end-to-end models do not depend on human-generated heuristics. However, they can not process long contexts [14], [17], [3]. The self-attention mechanism enables pre-trained Transformer models to learn a long context better and produces SOTA results in many NLP tasks [6]. Though it produces excellent results, it requires a lot of data and computational power to train. Therefore, categorizing the works in the Neural Network based approaches as single-hop question generation and multi-hop question generation is worth it.

In single-hop question generation, questions are generated from a single sentence or a paragraph. Since this is a moderately easy problem to solve compared to multi-hop question generation, the majority of work in the literature is focused on generating questions from a single-hop [14], [17], [3].

2.4.1 Single-hop Question Generation

Unlike multi-layer perceptrons, recurrent Neural Networks use their internal state to process a sequence of input. Hence, it makes them applicable to many NLP applications. Many kinds of recurrent Neural Networks have been proposed to address the vanishing gradient and exploding gradient drawbacks of basic Recurrent Neural Networks.

One of the significant contributions to automated question generation literature is the max-pointer approach introduced by Zhao et al. [14]. Prior to this study, many studies concentrated on generating questions from a single sentence, and inferior results were reported when a complete paragraph was used as input [14]. However, it is often necessary to use the entire paragraph to produce high-quality questions. Upon its publication, many researchers employed the max-pointer mechanism as part of their models. The limitation of this study is that from a paragraph, we can generate many questions, but as ground truth, we have a single question focused on a given answer. Therefore, a mechanism to quantify the interaction of answer and passage is essential to make a reasonable evaluation of question generation. Most of the prior work used a bit or additional word embedding to incorporate the answer focus into the model. In contrast, Song et al. [17] proposed an explicit mechanism to measure the interaction between the answer and the passage.

Table 2.4.1 shows the distribution of question words from SQuAD and MARCO. Unfortunately, the question word type is skewed and difficult to predict accurately. To mitigate this challenge, Zhon et al. [18] first generate question words from an answer-focused context enriched by lexical embedding and use the generated question type to steer the question generation. On the other hand, Sun et al. [3] incorporate relative distance

between the answer and context word as an additional technique to avoid copying the context word that is far and irrelevant to the answer. This modeling is done with the intuition that the context close to the answer can be the best source of information to generate question words. However, such intuition may not apply in all circumstances, particularly in phrases that include complicated answer-relevant relationships. Li et al. [19] address this difficulty by concurrently learning the unstructured sentence and structured answer relevant connection.

TYPE	**SQuAD**	**MARCO**
what	43.26%	44.29%
Who	9.39%	1.47%
How	9.12%	16.28%
When	6.26%	16.28%
Which	4.78%	1.11%
Where	3.76%	4.16%
Why	1.37%	1.88%
Others	21.83%	29.29%

Table 2.4.1: Distribution of question words in SQuRD [18] and MARCO[19] dataset

Source: [3]

In all the above studies, representation of the text is learned with the single objective of generating a question. In contrast to that Zhou et al [22] proposed a hierarchical multi-task learning model that learns the representation of the text at multiple levels, and it helped the model outperform the pointer generator that enriches with features on the SQuRD dataset.

Chan et al. [23] proposed a restructure of the BERT deployment to incorporate teacher force learning for the question generation. The proposed model is shown in Figure 2.4.7

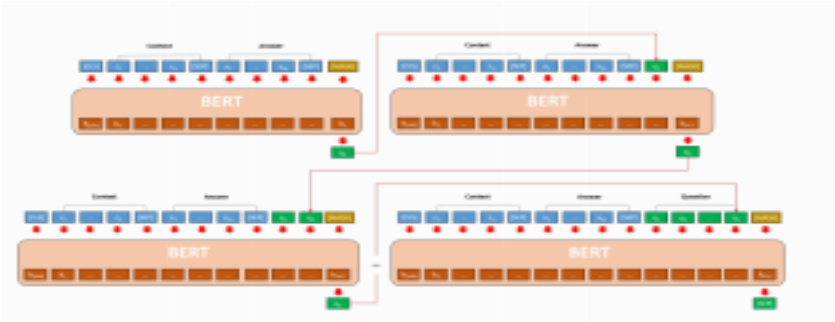

Figure 2.4.1: BERT-SQG architecture
Source:[19]

On top of having the benefit of contextual embedding, Scialom et al. [14] explored the copying mechanism, placeholder, and contextual word embedding to mitigate the challenges except for the given vocabulary words.

2.4.2 Multi-hop Question Generation

The intuition for multi-hop questions comes from the human process of generating multi-hop questions. We begin by skimming the material to gain a broad comprehension of it based on the provided answers and context. Then, given the context, we look for entities included in or connected to the answers and assess related phrases to extract helpful evidence. Finally, we aggregate the knowledge gained in earlier rounds and begin developing questions.

Su et al. [5] proposed a question-generating network based on a multi-hop encoder fusion network (MulQG). It employs a graph convolutional network to perform answer-aware entity embedding in multi-hops, and it is the first study to focus on multi-hop question synthesis. The effectiveness of this approach depends on how well the graph was created, but they used simple rules to make it. Therefore improving the graph construction may lead to better performance. On the other hand, A rich form of the questions with structure patterns on syntax and semantics is used to generate questions in [9]. A neural semi-hidden Markov model uses latent variables to estimate these patterns.

2.5 Pre-trained sequence-to-sequence modes

Fine-tuning a model that has been pre-trained on a large amount of unlabeled data

produced a SOTA result for many NLP tasks. The intuition behind such a transfer learning approach is that pre-training enables the model to learn common knowledge about the language, and fine-tuning enables the model to learn about the task-specific knowledge. There are encoder only, the decoder only, and encoder-decoder Transformer models pre-trained on a single language or many languages. BART [24], mBART [25], T5 [7], mT5 [8] are some of the encoder-decoder models that are suitable for sequence-to-sequence generation tasks. However, they have a huge number of parameters and require a lot of computation power to fine-tune [1].

2.5.1 Adapters and Adapter Fusion

Recently adapters have been introduced as a lightweight alternative to the full model fine-tuning that delivers comparable performance [1]. The adapter has several advantages over fine-tuning a model, including scalability, modularity, and composition [1]. Houlsby adapter [1], and Pfeifer adapter [1] are two popular variants of adapters where the former has two modules of trainable parameters while the latter has a single module of trainable parameters. However, the performance of the Pfeifer adapter is similar to Houlsby. Figures 2.5.1 and 2.5.2 below the Houlsby adapter [1] and Pfeifer adapter [1].

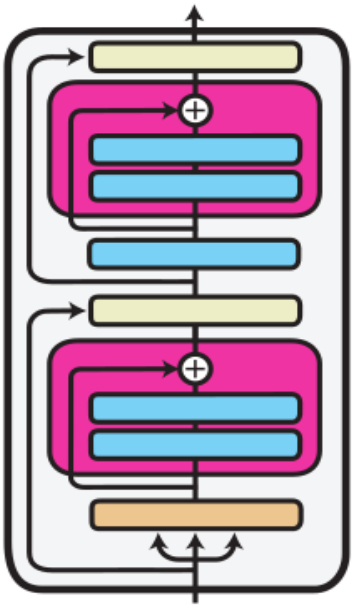

Figure 2.5.1 Houlsby adapter

Source: [1]

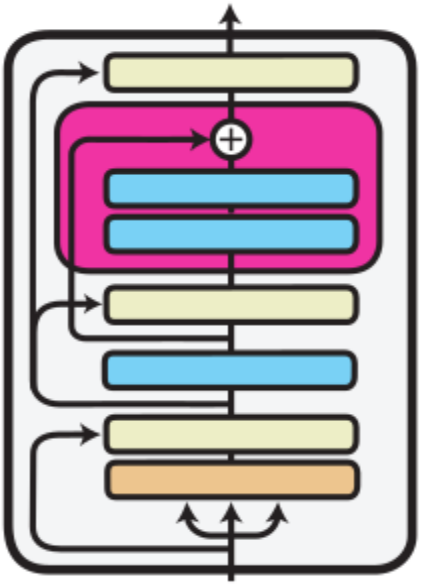

Figure 2.5.2 Pfeifer adapter

Source:[1]

Multi-task learning and sequential fine-tuning are two approaches for incorporating

perception from many NLP tasks. Pretrained sequence-to-sequence models are explored for multi-task learning. However, they struggle with catastrophic forgetting and data balance issues. Adapter Fusion [26], a new two-stage learning algorithm that uses knowledge from multiple tasks, is proposed to address this problem. Figure 2.5.3 below shows the architecture within a Transformer layer.

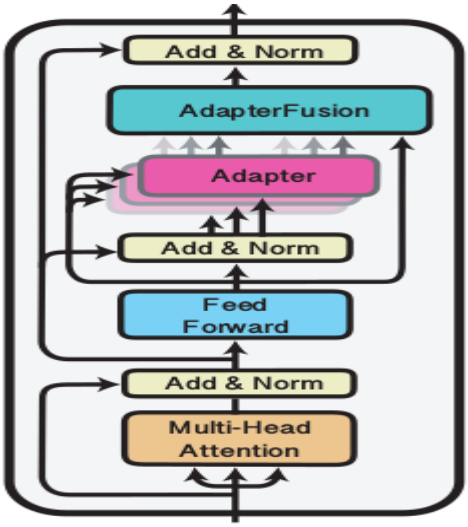

Figure 2.5.3: Architecture of Adapter fusion inside a Transformer layer.

Source: [26]

2.6 Summary

Rule-based approach can be used to generate syntactically and semantically correct but limited types of questions. On the other hand, RNNs are capable of generating a wide variety of questions. There are many techniques along with RNN used to process the long context effectively. Fine-tuning a pre-trained S2S model for text generation produced SOAT results. However, such fine-tuning is not possible in a computational resource-limited setting. Recently, adapter fine-tuning, and adapter fusion were proposed as an alternative for full model fine-tuning.

CHAPTER 3: METHODOLOGY

We exploit adapter modules to fine-tune pre-trained S2S models. Exploiting adapters enables the model to learn the target task with fewer parameters, reducing the computational complexity. Moreover, we train adapters in an end-to-end fashion. Hence, it does not require any heuristic defined by humans.

3.1 Experiment Setup

Many questions can be generated from a given paragraph. Generating a question focused on a given answer is essential to measure the quality of the generated question. In order to generate a question that focused on the given answer, we pre-pad the answer token to the context. Table 3.1.1 shows an example of formatted input and its formatted target for mBART-based models and Table 3.1.2 shows a formatted input and formatted target for the mT5-based model.

Input	[SEP]லாமா[SEP]உள்ளடக்கம்: திபெத்திய பௌத்தத்தில் திபெத்தில் தர்ம போதகர்கள் பொதுவாக லாமா என்று அழைக்கப்படுகிறார்கள். ஃபோவா மற்றும் சித்தி மூலம் மீண்டும் பிறக்க வேண்டும் என்று மனப்பூர்வமாக தீர்மானித்த ஒரு லாமா, பல முறை, போதிசத்வா சபதத்தைத் தொடர, துல்கு என்று அழைக்கப்படுகிறார்.[SEP]
Target	[SEP]திபெத்திய புத்த மதத்தில் ஆசிரியரின் பெயர் என்ன?[[SEP]

Table 3.1.1: Example of formatted input and its formatted target for mBART

Target	[SEP]பதில் : லாமா [SEP] உள்ளடக்கம்: திபெத்திய பௌத்தத்தில் திபெத்தில் தர்ம போதகர்கள் பொதுவாக லாமா என்று அழைக்கப்படுகிறார்கள். ஃபோவா மற்றும் சித்தி மூலம் மீண்டும் பிறக்க வேண்டும் என்று மனப்பூர்வமாக தீர்மானித்த ஒரு லாமா, பல முறை, போதிசத்வா சபதத்தைத் தொடர, துல்கு என்று அழைக்கப்படுகிறார்.[SEP]
Target	[SEP]திபெத்திய புத்த மதத்தில் ஆசிரியரின் பெயர் என்ன?[SEP]

Table 3.1.2: Example of formatted input and its formatted target for mT5

Rest of this chapter, we discuss the implementation details of our baselines and proposed models.

3.2 Baseline

Since only one prior work studied question generation in Tamil and does not report any of the automatics evaluation metrics, we developed standard approaches used in S2S generation as baselines. The following subsection will describe each of these techniques in detail.

3.2.1 Full Model Fine-tuning (mBART50)

A mBART50 model having 610M parameters is prepared to fine-tune on a translated Tamil training data set to generate questions.

3.2.2 Language Model Fine-tuning (T5)

A T5 base model is fine-tuned on a translated Tamil training dataset to generate questions.

3.2.3 Freeze Token Embedding and Position of mBART50

Token embedding and position embedding of the mBart50 pre-trained model's encoder and decoder is frozen, while other parameters are trainable

3.2.4 Zero-shot Questions Generator

A Houslby adapter with the bottleneck layer size of 264 is added to the pre-trained mBART50 and fine-tuned to generate English questions. Model performance is evaluated on a Tamil test set to measure Zero-Shot question generation.

3.2.5 Multiple Adapters for Cross-lingual Transfer Learning

It follows the MAD-X approach proposed in [26]. In the first stage, English and Tamil language adapters are learned parallelly with a Houslby adapter with a bottleneck size of 254. In the second stage, the English adapter and another newly added adapter called "question generator" are loaded to the model, and only the parameters of the "question generator" are fine-tuned to generate questions in English. After fine-tuning, the English language adapter is replaced by the Tamil language adapter to generate the question Tamil.

3.2.6 Prefix Tuning

Prefix Tuning is a shallow substitute for full language model fine-tuning. It freezes all the pre-trained language model parameters and introduces continuous task-specific vectors called Prefix. by optimizing 0.1% of the parameters, obtained SOTA results in a low resource setting [28].

3.3 Proposed Method

In this section we will discuss the implementation of our proposed models.

3.3.1 Adapter Fine-tuning with Houlsby Adapter

For adapter fine-tuning, we added Houslby adapters with a bottleneck layer size of 254 to the pre-trained mBART50 and fine-tuned it on the Translated Tamil dataset to generate questions in Tamil.

3.3.2 Adapter Fine-tuning with Pfeiffer Adapter

For the adapter fine-tuning, we added Pfeiffer adapters with a bottle layer size of 64 to the pre-trained mBART50 and fine-tuned it on the Translated Tamil dataset to generate questions in Tamil.

3.3.3 Adapter Fusion

In the first stage of the adapter fusion, the following adapters are trained parallelly.

1. **English to Tamil translation adapter:** A Houslby adapter with the bottleneck layer size of 264 is added to the pre-trained Transformer. Adapter weights are learned to translate text from English to Tamil.
2. **English question generation adapter**: A Houslby adapter with a bottleneck layer size of 254 is added to the pre-trained mBART50, and adapter weights are optimized to generate questions in English.
3. **Tamil question generation adapte**r: A Houslby adapter with a bottleneck layer size of 254 is added to the pre-trained mBART50, and adapter weights are optimized to generate questions in Tamil

In the second stage, all the above three adapters are fused to generate questions in Tamil.

CHAPTER 4: EVALUATION AND RESULT

This research Rouge metric to evaluate the proposed approaches. Next section variants of Rouge metrics that are appropriate to quantify the model quality.

4.1 Evaluation Metrics

To measure the quality of generated questions, we use the following metrics.

Rouge: Rouge is a set of metrics for evaluating machine-generated text. It works by comparing a machine-generated text aging a hypothesis text

- Precision: The percentage of n-gram that appears in the prediction is covered in the hypothesis
- Recall: The percentage of n-gram that appears in the ground truth covered in the prediction
- F1-score: is computed by precision and recall

ROUGE-L: Calculates the longest word sequence that matches.

4.2 Datasets

As discussed in section 2.2 many large-scale question answering datasets are available, which can be repurposed for question generation in English. However, only one dataset with 380 questions is available for Tamil question answering [26]. Therefore, to create a Tamil question generation training dataset, we randomly sampled 10,000 examples from the SQUAD [16] and translate them with Google Translate. Furthermore, we translate the 1190 questions from the XQUAD as a Test dataset. Table 5.1 shows an example from the test dataset, and Table 5.1 shows the average number of tokens in the training and test dataset.

Input	திபெத்திய பௌத்தத்தில் திபெத்தில் தர்ம போதகர்கள் பொதுவாக லாமா என்று அழைக்கப்படுகிறார்கள். ஃபோவா மற்றும் சித்தி மூலம் மீண்டும் பிறக்க வேண்டும் என்று மனப்பூர்வமாக தீர்மானித்த ஒரு லாமா, பல முறை, போதிசத்வா சபதத்தைத் தொடர, துல்கு என்று அழைக்கப்படுகிறார்.
Target	திபெத்திய புத்த மதத்தில் ஆசிரியரின் பெயர் என்ன?
Answer	லாமா

Table 4.2.1: Example question from the test dataset.

Data split	Average passage length	Average question length	Average answer length
train	290	19	9
test	303	20	9

Table 4.2.2: Average length of context, question, answer.

4.3 Result

Under a limited computational resource condition (Tesla V100), we were not able to perform the mBAT50 full model fine-tuning. Table 5.2 shows the result for the rest of the baselines and the proposed approaches. Our two models, the zero-shot question generator and the MAD-X style zero-shot question generator, produce a score of 0 for all the evaluation metrics. We examine the questions generated by these approaches, and it reveals that the questions generated by the model are in English rather than Tamil. Therefore task adapters learned in Zero-shot question generation not only learned to generate the questions but also learned English; hence it washed out the Tamil language during zero-shot inference.

Model	**Number of Total Parameters**	**Number of Trainable Parameters**	**Number of Epoch**
Language model fine-tuning(T5)	300M	300M	10
Freeze token embedding and position of mBART50	610M	352M	10
Zero-Shot questions generator	611M	420K	10
MAD-X style question generation	611M	420K	10
Adapter fine-tuning with Houlsby adapter	611M	420K	10
Adapter fine-tuning with Pfeiffer adapter	611M	420K	10
Adapter Fusion	763.5M	151M	30
Prefix Tuning	650M	39.5 M	10

Table 4.3.1: Model's and their parameters and trained Epochs

Model	Rouge-1			Rouge-2			Rouge-L		
	precision	recall	F1 score	precision	recall	F1 score	precision	recall	F1 score
Zero-shot question generation (mBART50)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MAD-X (mBART50)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Frozen embedding(mBART50)	2.30	1.07	1.09	0.05	0.03	0.04	2.3	1.7	1.9
Prefix tuning(mBART50	2.50	1.70	1.09	0.120	0.05	0.10	2.04	1.70	1.90
Language model fine-tuning(mT5)	4.10	3.30	3.50	0.40	0.40	0.30	4.00	3.20	3.50
Adapter fusion(mBART50	16.00	11.50	13.00	6.00	4.00	4.00	15.00	11.00	12.00
Adapter fine-tuning Pfeiffer(mBART50)	16.90	14.70	15.20	6.30	5.30	5.50	16.30	14.20	14.60
Adapter fine-tuning Houlsby(mBART50)	**19.00**	**15.50**	**16.60**	**7.10**	**5.50**	**6.00**	**18.40**	**15.00**	**16.10**

Table 4.3.2: Result

The frozen embedding language model fine-tuning produces slightly better results than Zero-Shot question generators. Next, adapter fine-tuning with the Pfeiffer adapter produces a marginally better result than the zero-shot question generators and the language model fine-tuning. Next, Adapter fine-tuning with Houlsby adapter produces 1.5 absolute improvements over the fine-tuning of the Pfeiffer adapter. Finally, adapter fusion produces a significantly better result than the zero-shot question generators and full model fine-tuning; however, its performance is below the Adapter fine-tuning. We suspect that this may be because of two reasons: the first one is English to Tamil translation; English question generation adapters are trained on the dataset other than SQUAD, which leads to a domain miss-match. The second reason may be that 10,000 training examples in the training set are insufficient to train the 151 million parameters introduced by the adapter fusion layer to combine the adapters. The performance of the prefix tuning, Language model fine-tuning and frozen embedding is very low maybe because of the low number of training examples and a low number of epochs. Table 5.5 shows the total number of parameters, Trainable parameters and the Number of Epochs for all the models.

CHAPTER 5: CONCLUSION AND FUTURE WORK

We explored computationally efficient approaches to generate questions in a zero-resource setting for the Tamil language. Under limited resources, we were not able to perform the full model fine-tuning with mBART50. By introducing standalone adapters and adapter fusion, we mitigate the computational challenge of full model fine-tuning and produce a significant result for Tamil question generation. The empirical result shows that adapter fine-tuning could produce SOTA results for Tamil question generation. In addition, Houlsby adapter and Pfeiffer adapter produce a similar result. Adapter fine-tuning with Houlsby produces the best result with 19 rouge. For future work, we recommend training an English to Tamil question translation adapter and an English question generation adapter on the SQUAD data prior to generating questions in Tamil with adapter fusion. We believe this will help the model to understand the target domain.

REFERENCES

[1] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, I. Gurevych, "AdapterHub: A Framework for Adapting Transformers", "Proceedings of the 2020 EMNLP (Systems Demonstrations), 2020, pp. 46–54.

[2] X. Jia, W. Zhou, X. Sun, and Y. Wu, "Eqg-race: Examination-type question generation", AAAI, 2021, pp. 13143-13151.

[3] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, "Answer-focused and position-aware neural question generation", Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3930–3939.

[4] S. Woo, Z. Li, and J. Mirkovic, "Good automatic authentication question generation", *Proceedings of the 9th International Natural Language Generation conference*, 2016, pp. 203–206.

[5] D. Su, Y. Xu, W. Dai, Z. Ji, T. Yu, and P. Fung, "Multi-hop question generation with graph convolutional network", The 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, pp. 16-20.

[6] L. E. Lopez, D. K. Cruz, J. C. B. Cruz, and C. Cheng, "Simplifying Paragraph-Level Question Generation via transformer Language Models", 8th Pacific International Conference on Artificial Intelligence, 2020, pp. 323-334.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer", Journal of Machine Learning Research, 2020, pp. 1-67.

[8] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua C. Raffel, "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer", Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 483–498

[9] Y.-H. Chan and Y.-C. Fan, “Bert for question generation”, *Proceedings of the 12th International Conference on Natural Language Generation*, 2019, pp. 173–177.

[10] N. Vignesh, S. Sowmya “Automatic question generation in Tamil”, “International Journal of Engineering Research and Technology.

[11] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for Machine Learning comprehension of text”, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp2383-23-92.

[12] V. Harrison and M. Walker, “Neural generation of diverse questions using answer focus, contextual and linguistic features”, *Proceedings of The 11th International Natural Language Generation Conference*, 2018, pp 296-306.

[13] K. D. Dhole and C. D. Manning, “Syn-qg: Syntactic and shallow semantic rules for question generation”, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp 752-765.

[14] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, “Paragraph-level neural question generation with maxout pointer and gated self-attention networks”, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3901–3910.

[15] X. Du, J. Shao, and C. Cardie, “Learning to ask: Neural question generation for reading comprehension”, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1342-1352.

[16] T. Scialom, B. Piwowarski, and J. Staiano, “Self-attention architectures for answer-agnostic neural question generation”, *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, 2019, pp. 6027–6032.

[17] L. Song, Z. Wang, W. Hamza, Y. Zhang, and D. Gildea, “Leveraging context information for natural question generation”, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 569–574.

[18] *W. Zhou, M. Zhang, and Y. Wu, "Question-type driven question generation," Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 6032-6037.*

[19] J. Li, Y. Gao, L. Bing, I. King, and M. R. Lyu, "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 3216-3226.

[20] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for Machine Learning comprehension of text", Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, Pp2383-23-92.

[21] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, T. Wang, "MS MARCO: A Human Generated Machine Reading COmprehension Dataset", "30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain."

[22] W. Zhou, M. Zhang, and Y. Wu, "Multi-task learning with language modeling for question generation", *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp, 3494-3499.

[23] J. Yu, W. Liu, S. Qiu, Q. Su, K. Wang, X. Quan, and J. Yin, "Low-resource generation of multi-hop reasoning questions", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6729–6739.

[24] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad,A. Mohamed, O. Levy,V. Stoyanov, L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880".

[25] Y. Liu, J Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, L Zettlemoyer, “Multilingual Denoising Pre-training for Neural Machine Translation”, “Transactions of the Association for Computational Linguistics, 2020, pp. 726–742”.

[26] J. Pfeiffer, A. Kamath, A. Ruckle, K. Cho, “AdapterFusion: Non-Destructive Task Composition for Transfer Learning”, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pp. 487–503.

[27] J. Pfeiffer, I. Vulić, I. Gurevych, S. Ruder, MAD-X: “An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer”, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020. pp. 7654–7673.

[28] X. Li, P. Liang, “Prefix-Tuning: Optimizing Continuous Prompts for Generation”,Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics,and the 11th International Joint Conference on Natural Language Processing”, pp 4582–4597.

[29] https://www.kaggle.com/c/chaii-hindi-and-tamil-question-answering.