Cross-lingual Transfer for Automatic Question Generation by Learning Interrogative Structures in Target Languages

Seonjeong Hwang[♠], **Yunsu Kim**[♣], **Gary Geunbae Lee**[♠,◇]
[♠] Graduate School of Artificial Intelligence, POSTECH, South Korea
[◇] Computer Science and Engineering, POSTECH, South Korea
[♣] aiXplain, Inc. Los Gatos, CA, USA
seonjeongh@postech.ac.kr, yunsu.kim@aixplain.com, gblee@postech.ac.kr

Abstract

Automatic question generation (QG) serves a wide range of purposes, such as augmenting question-answering (QA) corpora, enhancing chatbot systems, and developing educational materials. Despite its importance, most existing datasets predominantly focus on English, resulting in a considerable gap in data availability for other languages. Cross-lingual transfer for QG (XLT-QG) addresses this limitation by allowing models trained on high-resource language datasets to generate questions in low-resource languages. In this paper, we propose a simple and efficient XLT-QG method that operates without the need for monolingual, parallel, or labeled data in the target language, utilizing a small language model. Our model, trained solely on English QA datasets, learns interrogative structures from a limited set of question exemplars, which are then applied to generate questions in the target language. Experimental results show that our method outperforms several XLT-QG baselines and achieves performance comparable to GPT-3.5-turbo across different languages. Additionally, the synthetic data generated by our model proves beneficial for training multilingual QA models. With significantly fewer parameters than large language models and without requiring additional training for target languages, our approach offers an effective solution for QG and QA tasks across various languages[1].

1 Introduction

Automatic question generation (QG) aims to generate questions based on a given context. QG models have been employed not only to augment question-answering (QA) datasets but also to generate educational materials and develop chatbots. Several QA datasets have been proposed, including SQuAD (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018), and QuAC (Choi et al., 2018). However, the majority of these datasets are in English, resulting in a significant lack of data for other languages. Moreover, translating English datasets into other languages or creating new QA datasets, despite the availability of similar English datasets, is often inefficient in terms of both time and financial resources.

Recently, researchers have concentrated on cross-lingual transfer (XLT) to address data deficiencies in non-English languages (Sherborne and Lapata, 2022; Wu et al., 2022a; Vu et al., 2022; Deb et al., 2023; Pfeiffer et al., 2023). XLT involves deploying models trained on English datasets to other languages when annotated data in the target language is limited or unavailable.

Additionally, in recent years, multilingual large language models (mLLMs), such as GPT-4 (Achiam et al., 2023), BLOOM (Workshop et al., 2022), and PaLM (Chowdhery et al., 2023), have exhibited remarkable performance across various natural language generation (NLG) tasks, often achieving high efficacy through zero or few-shot inference. However, significant cost burdens are associated with utilizing commercial APIs, and employing open-source LLMs requires substantial computational resources. Previous studies on XLT for QG (XLT-QG) have typically utilized target language data, such as monolingual corpora, source-target parallel corpora, or a limited number of QA examples (Kumar et al., 2019; Chi et al., 2020; Shakeri et al., 2021; Wang et al., 2021; Agrawal et al., 2023). Nevertheless, incorporating language-specific data during model training can lead to inflexibility in language scalability, necessitating additional training efforts for applications in new languages.

In this paper, we present a simple and efficient XLT-QG method that generates **Qu**estions by learning **I**nterrogative **S**tructures in **T**arget languages (**QuIST**). QuIST comprises two stages: 1) Ques-

[1]We release our code and question exemplars used in our experiments at https://github.com/SeonjeongHwang/QuIST.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3194–3208
November 12-16, 2024

tion Type Classification (QTC) and 2) QG utilizing question exemplars. We categorize questions into eight types based on English interrogative words, and the QTC model determines the type of question to be generated based on the input context and answer. Once the question type is identified, it is used to select the corresponding question exemplars for the QG stage.

The QG model generates questions based on a given input context, answer, and question exemplars. During training with English data, the model learns to identify the interrogative structures specific to each question type from the provided exemplars. This approach enables the model to generate questions that are not only semantically aligned with the input context and answer but also syntactically similar to the exemplars. By training exclusively on English data, we ensure that the model can generate questions in other languages without requiring additional training.

In our experiments, we evaluate the performance of QuIST across nine linguistically diverse languages. Through both automatic and human evaluation, we show that QuIST outperforms various XLT-QG baselines and achieves performance comparable to GPT-3.5-turbo in several languages. Furthermore, we confirm that synthetic questions generated by QuIST are more effective for training high-performance multilingual QA models than those generated by GPT-3.5-turbo.

Our contributions can be summarized as follows:

- We introduce QuIST, a straightforward and efficient XLT-QG method that leverages interrogative structures from question exemplars in target languages during inference.
- QuIST exhibits high language scalability, as it can be readily applied to new languages with only a few question exemplars, without requiring additional parameter updates.
- Despite utilizing relatively smaller language models, such as mBERT (Devlin et al., 2018) with 110 million parameters and mT5 (Xue et al., 2021) with 1.2 billion parameters, QuIST generates questions of quality comparable to those produced by GPT-3.5-turbo.
- QuIST demonstrates greater effectiveness for data augmentation in multilingual QA compared to other XLT-QG baselines.

2 Cross-lingual Transfer for Automatic Question Generation

The zero-shot XLT approach–leveraging multilingual pretrained language models (mPLMs) fine-tuned exclusively on English data for target languages–has shown promising performance across various classification tasks (Liu et al., 2019; Conneau and Lample, 2019; Gritta and Iacobacci, 2021; Wu et al., 2022a; Li and Murray, 2023). However, when applied to natural language generation (NLG) tasks, this approach often results in catastrophic forgetting of the target language. To mitigate this issue,Maurya et al. (2021) proposed fine-tuning only the encoder layers of mPLMs while keeping the word embeddings and all decoder layer parameters frozen.

Finnish
Synthetic Question (mT5) : How long is Pyhäjärven pituus?
Synthetic Question (mBART) : How pitkä on Pyhäjärven muoto?
Ground Truth : Kuinka pitkä Pyhäjärvi on?
Translation: How long is Pyhäjärvi?

Korean
Synthetic Question (mT5) : When did 카를 마르크스 죽었다?
Synthetic Question (mBART) : When was 카를 하인리히 마르크스's birthday?
Ground Truth : 마르크스는 언제 사망하였는가?
Translation: When did Marx die?

Telugu
Synthetic Question (mT5) : How many వేరు వేరు మహాసముద్రాలు ఉన్నాయి?
Synthetic Question (mBART) : How many మహాసాగరాలుగా గుర్తిస్తారు?
Ground Truth : మహా సముద్రాలు ఎన్ని ఉన్నాయి?
Translation: How many great oceans are there?

Figure 1: Questions generated by mT5 (Xue et al., 2021) and mBART (Liu et al., 2020) fine-tuned on English QA datasets. The questions often contain English interrogative expressions such as "How long" and "When did."

In our preliminary investigation, we found that this technique did not completely prevent code-switching in XLT-QG, as shown in Figure 1. Specifically, the models struggled to fully grasp interrogative structures in the target language, a phenomenon we refer to as "interrogative code-switching." In this study, we propose a method that enables small mPLMs to learn interrogative structures without relying on target language data during training.

As illustrated in Figure 2, we divide the task into two stages. In the QTC stage, a classification model identifies the type of question to be generated. We focus on Wh-questions, categorizing them into eight types based on English interrogative words. While the type of question is primarily influenced by the type of answer, the model considers both the answer and the context. This is crucial,

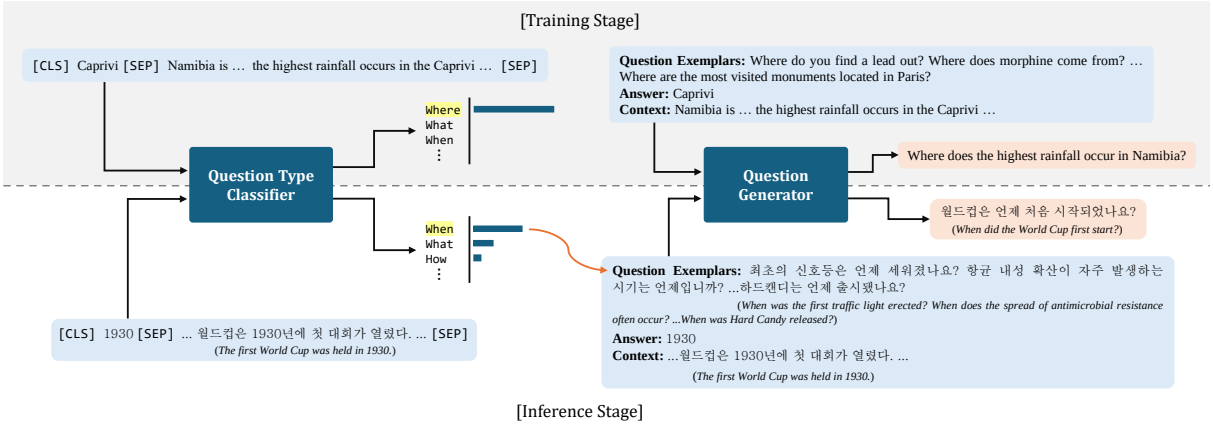

Figure 2: Overview of our proposed method: The QG model generates questions utilizing the question exemplars corresponding to the question type determined by the QTC model.

as the same answer can result in different types of questions depending on the context. For example, the number "911" could refer to a quantity, year, or proper noun.

The set of question exemplars corresponding to the question type identified by the QTC model is used in the QG stage. These exemplars are pre-created for each question type and language, as detailed in Section 3.1. By leveraging the shared interrogative structures among the exemplars, the QG model generates questions using the provided answer and context. Both the QTC and QG models are trained exclusively on English QA data and can be deployed to new languages without the need for additional training with target language data.

2.1 Question Type Classification

We categorize questions into eight types: *When*, *Where*, *What*, *Which*, *Who*, *Why*, How_{way}, and How_{number}[2]. To train the QTC model, we first annotate the question types in the English QA dataset, considering only those questions that fit into one of the eight categories. Specifically, questions starting with "how" are classified as How_{way} if followed by an auxiliary verb, or as How_{number} if followed by an adjective or adverb.

In this stage, we apply the zero-shot XLT approach. We fine-tune mBERT (Devlin et al., 2018) with a feed-forward classification layer using English QA data. The concatenation of the answer and context, separated by special tokens (i.e., `[CLS] answer [SEP] context [SEP]`), is fed into the QTC model. After encoding the input sequence using mBERT, the output hidden vector corresponding to the `[CLS]` token is passed through a feed-forward layer, followed by the softmax function, to compute probabilities for the eight question types. We use cross-entropy loss between the predicted and ground-truth labels to update all model parameters. During inference in target languages, the fine-tuned model predicts the question type by considering the answer and context in those languages.

2.2 Question Generation with Question Exemplars

We employ mT5 (Xue et al., 2021) as the backbone of our QG model, framing the task as a sequence-to-sequence prediction problem. The model is trained using the teacher-forcing technique to generate the ground-truth question based on the provided question exemplars, answer, and context. During training, the model learns to leverage the syntactic information from the question exemplars to generate questions that are both syntactically correct and semantically appropriate for the given context and answer. During inference, the question exemplars corresponding to the question type predicted by the QTC model are input into the QG model, helping it comprehend the interrogative structures of the target language.

3 Experimental Setup

In this section, we describe the datasets and baseline models we used in our experiments. Details regarding the implementation and evaluation metrics are provided in Appendices B and C.1, respectively.

[2] How_{way}-inquire about the manner in which something is done, while How_{number}-questions seek information regarding a degree or specific number.

3.1 Data

QA Datasets We used SQuAD1.1 (Rajpurkar et al., 2016) as the English QA dataset (***C-Q-A***$_{en}$) to train both the QTC and QG models. For evaluation, we collected QA examples in nine target languages (***C-Q-A***$_{tgt}$) from multilingual human-annotated QA datasets, including TyDiQA (Clark et al., 2020), XQuAD (Artetxe et al., 2020) and MLQA (Lewis et al., 2020). These datasets consist of *context–question–answer* triplets, where the answer is a span within the context. Details about these datasets are provided in Appendix D.

Question Exemplars The English question exemplars (***Q***$_{en}$) were randomly selected from the questions in the training set of *C-Q-A*$_{en}$ after labeling question types as described in Section 2.1[3]. To gather question exemplars in the target languages (***Q***$_{tgt}$) written by native speakers, we utilized the questions from the training set of *C-Q-A*$_{tgt}$. After translating these questions into English using Google Translation API, we constructed the question exemplars in the same manner as for English.

We experimented with several versions of question exemplars containing different number of questions: {1, 5, 10, 15}. In addition, we randomly sampled each version of the exemplars five times using different random seeds. Consequently, we trained five distinct QuIST models using different English question exemplars. During the inference stage, five sets of exemplars for each target language were utilized for evaluation. As a result, in Section 4, we report the average of 25 automatic evaluation results.

3.2 Baselines

We compared our QuIST method with several XLT-QG models that share the same backbone, mT5. All baselines treat the QG task as a sequence-to-sequence prediction, training the models to generate questions based on the concatenation of the input answer and context.

Baseline$_{EncDec}$ This model was simply trained by fine-tuning all parameters of mT5 using *C-Q-A*$_{en}$. This approach was introduced to examine the effect of training the parameters of the embedding layer and decoder on English data regarding catastrophic forgetting in the target language.

Baseline$_{Enc}$ Unlike Baseline$_{EncDec}$, only parameters of the encoder layers of mT5 were updated for this baseline model. This training technique was also employed to train QuIST, but the two models differ in whether the question exemplars are utilized.

Baseline$_{Multi}$ Inspired by the method proposed by Shakeri et al. (2021), we adopt multi-task fine-tuning, where mT5 simultaneously learns the English QG task and the question denoising task. The denoising task aims to restore questions with randomly masked tokens and we used Q_{tgt} with 15 exemplars for each question type (i.e., 120 questions) for a fair comparison with QuIST. We use this baseline to check whether utilizing a small number of question exemplars during the fine-tuning stage is also effective in XLT-QG. As this baseline learned language-specific data during training, we constructed different models for each language.

Baseline$_{Adapter}$ We implemented the Adapter-based mT5, which have been recently utilized in XLT for various NLP tasks (Pfeiffer et al., 2020; Deb et al., 2023; Pfeiffer et al., 2023; Wu et al., 2023). After training language-specific adapters using monolingual corpora[4], we trained the task-specific adapters using *C-Q-A*$_{en}$, where the English adapters are incorporated. While updating each type of adapter, we froze all other model parameters. In contrast to QuIST, this baseline does not utilize Q_{tgt}, but instead requires large-scale monolingual corpora in target languages.

Model	*Training*	*Inference*
Baseline$_{EncDec}$	*C-Q-A*$_{en}$	*C-Q-A*$_{tgt}$
Baseline$_{Enc}$	*C-Q-A*$_{en}$	*C-Q-A*$_{tgt}$
Baseline$_{Multi}$	*C-Q-A*$_{en}$, Q_{tgt}	*C-Q-A*$_{tgt}$
Baseline$_{Adapter}$	*C-Q-A*$_{en}$, S_{tgt}	*C-Q-A*$_{tgt}$
QuIST	*C-Q-A*$_{en}$, Q_{en}	*C-Q-A*$_{tgt}$, Q_{tgt}

Table 1: Data utilized by QuIST and baseline models.

Table 1 summarizes the datasets utilized by each model during both the training and inference stages. As indicated in the table, QuIST, Baseline$_{EncDec}$, and Baseline$_{Enc}$ are exclusively trained on English datasets. In contrast, Baseline$_{Multi}$ and Baseline$_{Adapter}$ make use of language-specific data during training. Consequently, distinct language-

[3]In preliminary experiments, we observed that using fixed exemplars was more effective than configuring random exemplars for each training example. A detailed analysis of this finding is provided in Appendix A.

[4]We extracted 50k raw sentences for each language from the Wikipedia dump (https://dumps.wikimedia.org) using WikiExtractor (https://github.com/attardi/wikiextractor), and the language-specific adapters were updated through a text denoising task.

Model	en	bn	de	fi	hi	id	ko	sw	te	zh	Avg
Baseline$_{EncDec}$	44.25	0.72	10.11	14.48	2.11	13.33	2.17	16.07	3.92	27.63	10.06
Baseline$_{Enc}$	44.45	14.53	25.00	19.95	23.45	20.37	11.76	16.72	14.79	40.83	20.82
Baseline$_{Multi}$	41.84	6.23	19.11	15.65	15.12	15.92	7.92	13.65	8.72	30.93	14.81
Baseline$_{Adapter}$	44.16	19.29	23.44	20.26	**31.41***	22.73	15.75	21.09	22.21	44.60	24.53
QuIST$_1$	43.48	14.96	25.75	27.73	21.82	23.06	11.51	20.84	10.44	42.40	22.06
QuIST$_5$	43.47	17.47	26.80	37.89	22.44	27.04	15.90	27.82	20.57	46.09	26.89
QuIST$_{10}$	43.40	**20.23**	**27.08**	38.36	27.26	28.32	23.86	**31.32***	29.98	**47.82***	30.47
QuIST$_{15}$	43.08	19.07	26.84	**38.79**	27.56	**28.36**	**25.14***	30.59	**30.74***	47.71	**30.53***
GPT-3.5-turbo$_{zero}$	33.98	21.30	27.76	35.55	24.84	31.18	18.56	27.90	17.31	41.67	27.34
GPT-3.5-turbo$_{10}$	37.63	21.51*	29.49*	39.41*	26.60	32.54*	22.28	30.12	23.13	44.47	29.95

Table 2: Automatic evaluation results for the nine target languages. This table shows the ROUGE-L performance of the models (SP-ROUGE (Vu et al., 2022) scores for Chinese). The best scores among mT5-based models are in **bold** and the highest scores among all models are marked with $\star$. We also report BLEU4 and METEOR scores and standard deviations in Appendix F.

specific models were trained for these two baselines.

4 Main Results

Comparison with Baselines Table 2 presents the performance of QuIST and the baseline models across nine target languages. The results show that QuIST$_{15}$, which achieved the highest performance among our models with varying numbers of question exemplars, outperforms several XLT-QG baselines, demonstrating a margin of 6.00 points compared to the most robust baseline, Baseline$_{Adapter}$. While adapting Baseline$_{Adapter}$ to a new language necessitates training language-specific adapter modules, our model can be readily deployed in new languages without the need for additional training.

QuIST notably outperforms Baseline$_{Enc}$ across all languages. Note that both models have the same number of trainable parameters during the fine-tuning stage. These results indicate that exposing the model to interrogative structures during the inference stage significantly enhances its ability to generate questions in the target language.

Despite Baseline$_{Multi}$ learning questions in the target language via the denoising task, it exhibited poor performance, even scoring lower than Baseline$_{Enc}$. Upon reviewing the generated results of Baseline$_{Multi}$, we frequently observed instances where the questions were unrelated to the input context or answer. These findings suggest that utilizing a small number of question exemplars during the training stage may lead to overfitting, resulting in a decline in model performance.

Comparison with LLMs We also compared QuIST and GPT-3.5-turbo, which stands out as a relatively cost-effective option among various commercial LLMs and demonstrates satisfactory results using only a few examples. We evaluated the performance of GPT-3.5-turbo through zero-shot inference and 10-shot inference, using prompts that included 10 English examples sampled from *C-Q-A$_{en}$*. The prompt templates we used are provided in Appendix E.

According to the results, QuIST$_{15}$ shows higher scores on average than the zero-shot and 10-shot inference of GPT-3.5-turbo. In detail, our model exhibits better performance in several languages, particularly in Hindi, Korean, Telugu, Swahili, and Chinese. Additionally, we investigated the few-shot inference of GPT-3.5-turbo that utilized our QTC model and question exemplars. The results are reported in Appendix G.

Human Evaluation We conducted a human evaluation in six languages where QuIST and GPT-3.5-turbo$_{10}$ exhibited similar automatic evaluation results, and we also evaluated the strongest baseline model, Baseline$_{Adapter}$. We collected a total of 240 questions generated by the three models per language and asked three native speakers to assess the question quality based on five criteria: *Interrogative Sentence* (*I*), *Grammatical Correctness* (*G*), *Clarity* (*C*), *Answerability* (*A*), *Answer-Match* (*A.M.*). The first two metrics were rated on a scale of 0, 1, 2, while responses for the remaining categories were binary (yes or no). More information regarding these criteria is described in Appendix C.2.

Table 3 presents the majority responses from three raters. For the criteria of clarity, answerability, and answer-match, we report the percentage of 'yes' responses. In German, Finnish, and In-

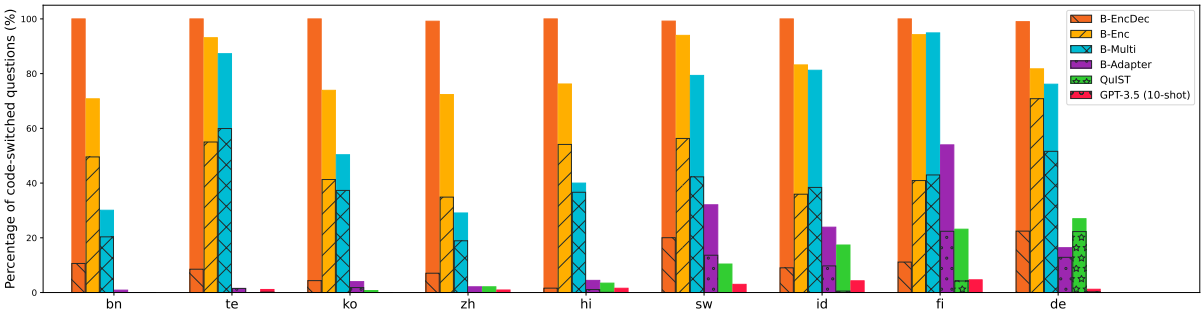

Figure 3: Percentage of code-switched synthetic questions. The patterned lower section of each bar represents the proportion of questions with only interrogative code-switching, while the full bar indicates the total proportion of all questions involving any form of code-switching.

	I	*G*	*C*	*A*	*A.M.*
			bn		
Baseline$_{Adapter}$	1.10	1.60	**76.6**	**76.1**	**72.3**
QuIST	1.05	1.65	73.8	70.5	68.2
GPT-3.5-turbo$_{10}$	**1.69**	**1.82**	64.7	64.7	64.7
			de		
Baseline$_{Adapter}$	1.62	1.48	79.2	77.9	55.1
QuIST	1.88	1.94	97.4	96.2	**96.2**
GPT-3.5-turbo$_{10}$	**1.96**	**2.00**	**100**	**98.8**	95.0
			fi		
Baseline$_{Adapter}$	0.82	1.08	**100**	**100**	73.8
QuIST	1.97	1.91	**100**	**100**	**100**
GPT-3.5-turbo$_{10}$	**2.00**	**1.98**	**100**	**100**	98.2
			hi		
Baseline$_{Adapter}$	1.83	1.84	31.3	32.3	20.7
QuIST	1.73	1.50	28.6	26.5	**25.7**
GPT-3.5-turbo$_{10}$	**1.99**	**1.96**	**32.5**	**32.9**	24.6
			id		
Baseline$_{Adapter}$	1.78	1.86	89.2	77.0	47.3
QuIST	1.96	**2.00**	**100**	98.7	97.5
GPT-3.5-turbo$_{10}$	**2.00**	**2.00**	**100**	**100**	**98.8**
			sw		
Baseline$_{Adapter}$	1.36	1.73	42.4	33.9	6.8
QuIST	1.94	1.82	82.5	76.3	55.0
GPT-3.5-turbo$_{10}$	**2.00**	**1.95**	**98.8**	**98.8**	**96.3**

Table 3: Human evaluation results.

donesian, the questions generated by QuIST and GPT-3.5-turbo$_{10}$ consistently received high scores across all criteria. Specifically, both models effectively generate questions that align with the given answers, outperforming Baseline$_{Adapter}$. In contrast, our model achieves lower overall scores in Bengali and Hindi compared to the previously mentioned languages. However, this performance decline is also observed in GPT-3.5-turbo$_{10}$ and Baseline$_{Adapter}$.

In Swahili, QuIST lagged significantly behind GPT-3.5-turbo$_{10}$ in terms of "Answerability" and "Answer-Match." However, given that Baseline$_{Adapter}$ generates questions of significantly lower quality–despite outperforming all other baseline models in automated evaluation–it is noteworthy that our model can generate Swahili questions of acceptable quality without any specific training in the target language.

5 Analysis

5.1 Interrogative Code-switching

We investigated the frequency of interrogative code-switching occurrence in questions generated by different XLT-QG methods[5]. As depicted in Figure 3, interrogative code-switching is observed in the majority of questions generated by Baseline$_{EncDec}$. This phenomenon can be attributed to catastrophic forgetting in target languages, as both the encoder and decoder were fine-tuned using English data. In Baseline$_{Enc}$, where only the encoder was fine-tuned, this issue is slightly alleviated; nevertheless, more than half of the synthetic questions still exhibit this code-switching problem.

Through the results of Baseline$_{Multi}$, we confirm that interrogative code-switching is alleviated in numerous languages due to the impact of the question denoising task specific to the target language. Both QuIST and Baseline$_{Adapter}$ prove comparable effectiveness in mitigating interrogative code-switching, surpassing other baseline approaches. Specifically, our model demonstrates effective in alleviating interrogative code-switching

[5]We used cld3 (https://github.com/google/cld3) to identify the languages. If the target language comprised less than 70% of the generated question, it was classified as code-switching. If the target language accounted for more than 70% but included English interrogative words, it was classified as interrogative code-switching.

Method	bn	fi	id	ko	sw	te	Avg
English-only	33.63	54.05	55.75	49.03	50.30	56.40	49.86
Baseline$_{Enc}$	56.34	53.71	57.52	56.04	60.12	68.01	58.62
Baseline$_{Adapter}$	54.87	50.85	58.29	52.90	59.72	64.43	56.84
Prompt-tuned PaLM	54.57	**54.14**	59.18	56.16	64.00	**69.21**	59.54
GPT-3.5-turbo$_{10}$	54.28	53.28	56.34	53.87	**64.06**	64.92	57.79
QuIST	**59.59**	53.33	**59.53**	**57.37**	60.05	68.01	**59.65**

Table 4: Exact match scores of multilingual QA models trained on datasets synthesized using different methods.

Model	*Training* (en)	*Inference* (tgt)	Avg ROUGE
QuIST	Human & Classified	Human & Classified	**30.53**
(1)	Human & Classified	Translator & Classified	27.65
(2)	Human & Typeless	Human & Typeless	23.59
(3)	×	Human & Classified	16.96
Baseline$_{Enc}$	×	×	20.82

Table 5: Performance of XLT-QG models using question exemplars in different ways.

observed in low-resource languages such as Bengali and Swahili.

5.2 Data Augmentation for Question Answering

We explored the potential of QuIST for augmenting training data for multilingual QA models. Specifically, we compared synthetic data generated by QuIST and baseline models[6] with the multilingual QA dataset generated by Agrawal et al. (2023), which used their PaLM-540B model prompt-tuned with five QA examples from target languages. Table 4 presents the average exact match (EM) scores across six languages for the multilingual QA models. The training details can be seen in Appendix B.

According to the results, QuIST achieves the best performance, surpassing GPT-3.5-turbo$_{10}$ and prompt-tuned PaLM-540B. Interestingly, contrary to the findings from the automatic evaluation and interrogative code-switching analysis, Baseline$_{Enc}$ demonstrates greater effectiveness in QA data augmentation compared to Baseline$_{Adapter}$. In the earlier experiment, over 70% of the questions generated by Baseline$_{Enc}$ exhibited code-switching issues. However, unlike Baseline$_{Adapter}$, which depends solely on task-specific adapters for learning the QG task, Baseline$_{Enc}$ leverages all encoder parameters. This suggests that Baseline$_{Enc}$ may be capable of producing questions with higher semantic quality.

[6]The questions were generated based on the context and answer pairs within the synthetic dataset released by Agrawal et al. (2023).

5.3 Impact of Different Question Exemplars

We investigated the impact of utilizing different methods for constructing question exemplars compared to our proposed approach. These approaches were compared to Baseline$_{Enc}$, where only the encoder is fine-tuned on English data, without using additional data from target languages during both training and inference. Table 5 presents the average ROUGE scores across nine languages.

(1) QuIST utilizes human-written question exemplars in target languages during inference. In this experiment, we evaluate the model's performance using exemplars translated from English questions via the Google Translation API. The results show that while machine-translated exemplars improve target language question generation compared to Baseline$_{Enc}$, they are less effective than human-written exemplars.

(2) We conducted training and inference using exemplars that covered all question types to evaluate the effectiveness of type-specific question exemplars. The exemplars included two instances of each of the eight question types, totaling 16 questions, and the QTC model was not used in this setting. The results indicate a slight performance improvement compared to Baseline$_{Enc}$; however, this effect is marginal.

(3) We investigated whether input question exemplars during the inference stage are beneficial, even without the training process for generating questions using question exemplars. The model was trained to generate a question based on the given context and answer without utilizing the question

exemplars, similar to Baseline$_{Enc}$, and only used the exemplars in the inference stage. In this setting, question exemplars in the target language were not helpful, meaning that QuIST learns how to utilize question examples for QG during training.

5.4 Question Type Classification

Labeling Type	en	tgt
Hard	62.92	52.86
Relaxed	96.38	91.13

Table 6: Performance of the QTC model.

To measure the zero-shot inference performance of the QTC model for the target languages, we first annotated the ground-truth question types of the target language QA data. We translated the questions into English and conducted annotation as detailed in Section 2.1 (i.e., hard labeling). Table 6 displays the macro F1 scores of the QTC model, measured based on ground-truth labels constructed in two ways. Since most Wh-questions can be paraphrased into questions beginning with "what" and "which,"[7] we also evaluate the QTC performance in a setting where "what" and "which" are accepted as additional gold labels (i.e., relaxed labeling). According to the results measured with the relaxed labels, the model correctly classified more than 90% of questions. This suggests that the error propagation resulting from misclassification in QTC is minimal throughout the entire pipeline.

5.5 Case Study

We analyzed the questions generated by the models we used in the experiments, particularly focusing on Swahili, where our model received lower rating than GPT-3.5-turbo$_{10}$ in human evaluation. In Figure 4, we can see that the question generated by QuIST is insufficient to explain the given answer, and these incorrect generations resulted in the low "Answer-Match" score. We also note that Baseline$_{EncDec}$ and Baseline$_{Enc}$ encounter code-switching issues, and the question generated by Baseline$_{Multi}$ contains information that is not present in the context. Furthermore, the question generated by Baseline$_{Adapter}$ was assessed as not being a question, as it is a descriptive sentence ending with a question mark.

[7]For example, "How large is the Mupartifad village?" is equivalent to "What is the area of Mupartifad village?"

Context: Malawi, Zambia na Zimbabwe wakati mwingine zinehesabiwa kuwa sehemu ya Afrika ya Kusini (zamani zilikuwa pamoja kama Rhodesia ya Kusini, Rhodesia ya Kaskazini na Unyasa katika Shirikisho la Afrika ya Kati)
(Malawi, Zambia and Zimbabwe are sometimes considered part of South Africa (they used to be together as Southern Rhodesia, Northern Rhodesia and Nyasa in the Central African Federation))
Answer: Zambia

Baseline$_{EncDec}$	Along with Malawi, Zimbabwe and Zimbabwe, which nations sometimes zinehesabiwa sehemu ya Africa ya Kusini? *(Along with Malawi, Zimbabwe and Zimbabwe, which nations are sometimes considered part of South Africa?)*
Baseline$_{Enc}$	What nchi zinahesabiwa kuwa sehemu ya Afrika ya Kusini? *(What countries are considered to be part of South Africa?)*
Baseline$_{Multi}$	Malawi, Zimbabwe, na eneo gani lilikuwa "mji mzuri zaidi nchini Uingereza" mbele ya Southampton mnamo 2007? *(Malawi, Zimbabwe, and which region was the "best city in England" ahead of Southampton in 2007?)*
Baseline$_{Adapter}$	Malawi, Zimbabwe na Zimbabwe wakati mwingine zinahesabiwa kuwa sehemu ya Afrika ya Kusini *(Malawi, Zimbabwe and Zimbabwe are sometimes considered part of South Africa?)*
GPT-3.5-turbo$_{10}$	Ni nchi gani inahesabiwa kuwa sehemu ya Afrika ya Kusini pamoja na Malawi na Zimbabwe? *(Which country is considered part of South Africa along with Malawi and Zimbabwe?)*
QuIST	Ni nchi ipi iliyohesabiwa kuwa sehemu ya Afrika ya Kusini? *(Which country is considered part of South Africa?)*
Ground-Truth	Je, Rhodesia ya Kaskazini ina jina gani kwa sasa? *(What is the current name of Northern Rhodesia?)*

Figure 4: Examples of synthetic questions in Swahili.

6 Related Work

Prior work on XLT for NLG tasks has primarily focused on training models with source language datasets while maintaining the ability to generate outputs in the target language. For example, Mallinson et al. (2020) and Chi et al. (2020) leveraged parallel corpora to improve the alignment between source and target languages, facilitating a more effective transfer of task-related knowledge. Similarly, Maurya et al. (2021) enhanced the mPLM model through an auxiliary task closely related to the downstream task, using only monolingual data, and applied it to various NLG tasks in the XLT setting. In another approach, Vu et al. (2022) demonstrated that prompt-tuning effectively mitigated catastrophic forgetting of the target language in zero-shot cross-lingual summarization. More recently, researchers such as Wu et al. (2022b), Deb et al. (2023), and Pfeiffer et al. (2023) have explored methods to separate the acquisition of language-specific knowledge from language-agnostic knowledge, aiming to improve cross-lingual performance.

Unlike most generation tasks that focus on pro-

ducing declarative sentences, QG involves the additional complexity of generating interrogative sentences designed to elicit specific information. While our approach avoids training models using target language data, much of the prior research has relied on such data. For instance, Kumar et al. (2019) utilized a combination of English QA data and a limited amount of target language data. In contrast, Shakeri et al. (2021) trained their model using a denoising task on a question corpus in the target language. Additionally, Agrawal et al. (2023) prompt-tuned the PaLM-540B model using five sets of target language QA examples and used the model to synthesize multilingual QA dataset. Finally, Chi et al. (2020) adopted a language modeling approach with parallel corpora and restricted the question decoding phase to tokens from the target language vocabulary.

7 Conclusion

In this paper, we proposed a simple yet effective XLT-QG approach, where the question generation model is trained solely on an English QA dataset and leverages a small set of target language questions during inference. By incorporating question exemplars from target languages, our method enables the model to learn the interrogative structures of those languages, effectively addressing the issue of code-switching.

Experimental results demonstrate that this approach significantly outperforms several XLT-QG baselines and achieves performance comparable to GPT-3.5-turbo across a variety of languages. Additionally, we validated the utility of our method's synthetic data for training multilingual QA models.

A key strength of our method lies in its scalability and parameter efficiency, as it relies exclusively on English QA data during training. This enables the seamless extension to new languages without the need for additional parameter updates. Moreover, in contrast to LLMs, our approach employs smaller backbone models, offering the advantages of lower deployment costs and reduced computational requirements, making it more accessible for practical use in diverse multilingual settings.

8 Limitations

While our model demonstrates strong cross-lingual capabilities, its applicability remains constrained to the languages on which the mPLMs have been trained. Although the mT5 model employed in our study was pre-trained on a diverse set of 101 languages, there remain many underrepresented or low-resource languages where the model's performance may be limited.

Another limitation is the instability in model performance, which can vary depending on the configuration of the question exemplars in the target language. Some questions generated by the model continue to exhibit code-switching issues. While this issue may affect the grammatical and linguistic consistency of the outputs, it can be mitigated through the use of a simple rule-based filtering technique. Nonetheless, this solution may not entirely eliminate the problem and could require further refinement, particularly in more complex multilingual contexts.

Acknowledgements

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00437866) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) and also by the Technology Innovation Program (20015007, Development of Digital Therapeutics of Cognitive Behavioral Therapy for treating Panic Disorder) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2023. Qameleon: Multilingual qa with only 5 examples. *Transactions of the Association for Computational Linguistics*, 11:1754–1771.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7570–7577.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in neural information processing systems*, 32.

Ujan Deb, Ridayesh Parab, and Preethi Jyothi. 2023. Zero-shot cross-lingual transfer with learned projections using unlabeled target-language data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 449–457.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Milan Gritta and Ignacio Iacobacci. 2021. Xeroalign: Zero-shot cross-lingual transformer alignment. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 371–381.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. 2019. Cross-lingual training for automatic question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4863–4872.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. Mlqa: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330.

Tianjian Li and Kenton Murray. 2023. Why does zero-shot cross-lingual generation fail? an explanation and a solution. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Ngan Fung. 2019. Zero-shot cross-lingual dialogue systems with transferable latent variables. In *EMNLP-IJCNLP 2019-2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2020. Zero-shot crosslingual sentence simplification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5109–5126.

Kaushal Kumar Maurya, Maunendra Sankar Desarkar, Yoshinobu Kano, and Kumari Deepshikha. 2021. Zmbart: An unsupervised cross-lingual transfer framework for language generation. *arXiv preprint arXiv:2106.01597*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jonas Pfeiffer, Francesco Piccinno, Massimo Nicosia, Xinyi Wang, Machel Reid, and Sebastian Ruder. 2023. mmt5: Modular multilingual pre-training solves source language hallucinations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1978–2008.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siamak Shakeri, Noah Constant, Mihir Kale, and Linting Xue. 2021. Towards zero-shot multilingual synthetic question and answer generation for cross-lingual reading comprehension. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 35–45.

Tom Sherborne and Mirella Lapata. 2022. Zero-shot cross-lingual semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4134–4153.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9300.

Bingning Wang, Ting Yao, Weipeng Chen, Jingfang Xu, and Xiaochuan Wang. 2021. Multi-lingual question generation with language agnostic language model. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2262–2272.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Han Wu, Haochen Tan, Kun Xu, Shuqi Liu, Lianwei Wu, and Linqi Song. 2022a. Zero-shot cross-lingual conversational semantic role labeling. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 269–281.

Ting-Wei Wu, Changsheng Zhao, Ernie Chang, Yangyang Shi, Pierce Chuang, Vikas Chandra, and Biing Juang. 2023. Towards zero-shot multilingual transfer for code-switched responses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7551–7563.

Xianze Wu, Zaixiang Zheng, Hao Zhou, and Yong Yu. 2022b. Laft: Cross-lingual transfer for text generation by language-agnostic finetuning. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 260–266.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

A Static and Dynamic Exemplars

Since gathering sufficient question samples in the target languages is challenging, we used fixed question exemplars during inference. In contrast, English question exemplars can be easily sourced from QA datasets. Thus, we experimented with two approaches for creating question exemplars to train the QG model: (1) Static exemplars, which use fixed exemplars across all training examples, and (2) Dynamic exemplars, which are sampled from the English QA dataset for each training example.

Language	Dynamic	Static
bn	**20.40** $\pm$ 0.40	20.13 $\pm$ 0.71
de	26.53 $\pm$ 0.19	**26.84** $\pm$ 0.26
fi	35.91 $\pm$ 1.35	**43.09** $\pm$ 2.18
hi	26.97 $\pm$ 0.44	**27.72** $\pm$ 0.28
id	27.42 $\pm$ 2.04	**29.96** $\pm$ 2.22
ko	23.35 $\pm$ 0.37	**27.01** $\pm$ 0.83
sw	27.57 $\pm$ 0.86	**32.01** $\pm$ 1.09
te	27.64 $\pm$ 0.99	**32.96** $\pm$ 1.17
zh	47.29 $\pm$ 0.22	**47.64** $\pm$ 0.26
AVG	29.23	**31.93**

Table 7: Comparison of models using dynamic and static exemplars during training. We report SP-ROUGE scores for Chinese and ROUGE-L scores for other languages. The scores for the static setting are based on the English exemplars, representing median performance.

As shown in Table 7, both approaches demonstrate effective performance in target languages compared to the existing XLT-QG baseline models (Table 2). However, the static exemplar method achieves better overall performance across various languages. During training, our model generates questions by leveraging the syntactic information from the exemplars while utilizing the semantic information from the input context and answer. We hypothesize that the model trained with static exemplars was better able to focus on the syntactic structures of the example questions, leading to improved performance. Consequently, we utilized static exemplars in all our experiments.

B Implementation Details

We utilized a single NVIDIA Tesla A100-80GB GPU for model training. The QTC and QG models were initialized using `bert-base-multilingual-cased` with 110M parameters and `google/mt5-large` with 1.2B parameters, sourced from HuggingFace[8]. Training was conducted employing stochastic gradient descent with the AdamW optimizer (Loshchilov and Hutter, 2018) coupled with a linear learning rate scheduler encompassing 1000 warm-up steps. Batch sizes and learning rates were set as (8, 1e-5) and (16, 5e-5) for QTC and QG, respectively. Training ceased upon optimization of the models on the validation set.

Due to variations in the number of examples across different question types, we employed data upsampling based on the type with the highest number of examples for training the QTC model. During the inference stage, we determined the question type with the highest predicted probability from the QTC model and generated questions using the beam search algorithm with a beam size of 4.

To train multilingual QA models in Section 5.2, we adopted the methodologies used by Agrawal et al. (2023). Each QA model underwent training using a combination of English data sourced from the TyDiQA training set and synthetic data for all languages, generated by each XLT-QG model. Given the unavailability of the TyDiQA test set, we evaluated the validation performance instead. The backbone of the QA model consisted of `google/mt5-xl` with 3.7B parameters, fine-tuned with a learning rate of 2e-4 and a batch size of 64. We selected the model checkpoint yielding the highest EM score for each language, following the strategy of Agrawal et al. (2023), and reported the average scores obtained from utilizing three different random seeds.

C Metric

C.1 Automatic Evaluation

In accordance with previous studies on QG, we use BLEU4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004) as automatic evaluation metrics. These metrics measure the n-gram similarity between model predictions and references. However, these evaluation metrics are not suitable for Chinese (zh), where words are not separated by white space. Therefore, we additionally used SP-ROUGE (Vu et al., 2022) that using SentencePiece sub-word tokenization (Kudo and Richardson, 2018).

C.2 Human Evaluation

We enlisted three native speakers for each language via Upwork[9] to evaluate the quality of our synthetic questions. The questions were rated based on five criteria:

- *Interrogative Sentence* evaluates whether the question has an interrogative structure.
  0: This is not a question.
  1: This is a question, but it doesn't have the typical structure of an interrogative sentence.
  2: This is a natural interrogative structure.

- *Grammatical Correctness* evaluates the grammatical accuracy of the question.
  0: Numerous grammatical errors make the question unacceptable.
  1: Some errors exist but do not hinder understanding of the question.
  2: The question is grammatically correct.

- *Clarity* determines whether the question is clear and easily understandable given the context. Answer **yes** or **no**.

- *Answerability* determines whether the question can be answered using information from the context. Answer **yes** or **no**.

- *Answer-Match* determines whether the input answer could be a valid answer to the question considering the content of the provided context. Answer **yes** or **no**.

If a score of "0" is assigned to the *Interrogative Sentence* category, evaluations for the remaining categories did not conducted. Additionally, if a score of 0 is rated in *Grammatical Correctness*, or if "no" is selected for *Clarity*, *Answerability*, or *Answer-Match* categories, subsequent evaluations can not be carried out. Therefore, in this case, the lowest scores were assigned for these criteria.

D Data Usage

We used SQuAD1.1 (Rajpurkar et al., 2016) as the English QA data *C-Q-A*$_{en}$ for training our models. As only training and validation sets are publicly

[8] `https://huggingface.co`

[9] `https://www.upwork.com`

available, we partitioned the training set and employed a portion of the examples for validation purposes. The original validation set served as our test set. The training, validation, and test sets comprised 79,321, 8,283, and 1,190 examples, respectively. Furthermore, the distribution of examples by question type is summarized in Table 8.

Question Type	# Examples
What	33,777
Who	7,951
How_{number}	5,657
When	4,780
Which	3,931
Where	2,953
How_{way}	1,600
Why	1,054

Table 8: Number of examples by question type in training set of *C-Q-*A_{en}.

Language	Code	# Examples	
		Train	Test
Bengali	bn	2,390	113
Chinese	zh	5,137	1,190
German	de	4,517	1,190
Finnish	fi	6,855	782
Hindi	hi	4,918	1,190
Indonesian	id	5,702	565
Korean	ko	1,625	276
Swahili	sw	2,755	499
Telugu	te	5,563	669

Table 9: Language codes and the number of examples in *C-Q-*A_{tgt} dataset. In our method, only a small portion of the training examples are used as question exemplars.

Table 9 presents the statistics of target language QA data *C-Q-*A_{tgt} utilized by our models during inference. Note that training examples were solely employed for sampling question exemplars Q_{tgt}. Test examples in Chinese, German, and Hindi were collected from the XQuAD (Artetxe et al., 2020) test set, whereas training examples were sourced from the MLQA (Lewis et al., 2020) validation set, as XQuAD does not provide a training set for the target languages. Training and test examples in other languages were obtained from TyDiQA (Clark et al., 2020).

E Prompt Template for GPT-3.5-turbo

We evaluated the zero-shot and few-shot performance of `gpt-3.5-turbo-0125` model. We extracted sets with different numbers of examples: 1, 3, 5, and 10, from *C-Q-*A_{en} to employ for few-shot inference. In addition, we used five versions of each set, varying the random seed. Based on the English validation set, we determined the optimal number of examples (see Table 10), and used the set with the median performance as the component in the few-shot prompt. Subsequently, we conducted zero-shot and 10-shot inference for various languages using the prompts described in Figure 5 and 6, respectively.

Prompt Type		BLEU-4	METEOR	ROUGE-L
Zero-shot		15.01	53.28	40.32
Few-shot	1	17.58 ± 3.04	52.99 ± 0.80	40.20 ± 3.11
	3	18.28 ± 1.82	53.43 ± 1.01	41.10 ± 1.71
	5	19.09 ± 0.85	54.02 ± 1.11	41.40 ± 1.27
	10	**19.42** ± 1.02	**54.37** ± 0.69	**42.10** ± 1.01

Table 10: Performance of GPT-3.5-turbo on the SQuAD1.1 validation set. We report the mean and standard deviation of the few-shot inference results.

Input Template

Considering the given context, generate a question for the given answer in the same language as the given context:
Context: `{context}`
Answer: `{answer}`
Question:

Model Prediction

`{question}`

Figure 5: The input and output template for zero-shot inference of GPT-3.5-turbo.

Additionally, we empirically observed that specifying the language of the questions to be generated is essential for effective few-shot inference. Even when the input context and answer are in non-English languages, the model frequently generated English questions when the language to be generated was not specified.

F Automatic Evaluation Results

Table 11, 12, and 13 show detailed results for the experiments in Section 4.

G GPT-3.5-turbo few-shot Inference with Question Type Classification

We additionally investigated whether the QTC model and question exemplars are beneficial for few-shot inference of GPT-3.5-turbo. In this experiment, we utilized the exemplar set that exhibited the best performance for each language in

Input Template
Considering the given context, generate a question for the given answer in the same language as the given context: [Example 1] Context: … In total, Afrikaans is the first language in South Africa alone of about 6.8 million people and is estimated to be a second language for at least 10 million people worldwide, compared to over 23 million and 5 million respectively, for Dutch. Answer: 6.8 million English question: About how many South Africans speak Afrikaans as their primary language? … [Example 10] Context: … In ring-porous species, such as ash, black locust, catalpa, chestnut, elm, hickory, mulberry, and oak, the larger vessels or pores (as cross sections of vessels are called) are localised in the part of the growth ring formed in spring, thus forming a region of more or less open and porous tissue. The rest of the ring, produced in summer, is made up of smaller vessels and a much greater proportion of wood fibers. … Answer: ring-porous English question: What species of hardwood are hickory and mulberry trees? [Example 11] Context: `{context}` Answer: `{answer}` `{language}` question:
Model Prediction
`{question}`

Figure 6: The input and output template for 10-shot inference of GPT-3.5-turbo.

our method. We supplemented these exemplars with the statement "The followings are examples of `language` questions:" placed before the prompt in Figure 6. According to the results in Table 14, leveraging the QTC model and question exemplars leads to particularly improved performance in low-resource languages such as Bengali, Telugu, and Swahili.

Model	en	bn	de	fi	hi	id	ko	sw	te	Avg
Baseline$_{EncDec}$	23.45	0.00	3.62	2.91	0.35	5.59	0.00	4.46	0.97	2.24
Baseline$_{Enc}$	23.72	5.64	13.57	6.27	10.01	10.11	4.38	5.80	3.64	7.43
Baseline$_{Multi}$	23.45	2.04	9.38	3.17	3.63	6.46	1.85	2.35	1.77	3.83
Baseline$_{Adapter}$	21.79	6.96	11.34	5.57	12.28	9.10	4.41	6.38	6.41	7.81
QuIST$_1$	22.32 ± 0.06	5.18 ± 0.72	12.97 ± 0.39	13.02 ± 2.04	7.78 ± 1.31	12.81 ± 0.88	2.54 ± 1.50	8.24 ± 2.18	3.41 ± 1.05	8.24
QuIST$_5$	22.20 ± 0.13	6.62 ± 0.97	13.43 ± 0.30	20.50 ± 1.54	7.84 ± 1.19	14.78 ± 0.79	5.57 ± 4.21	15.07 ± 2.87	9.38 ± 4.27	11.65
QuIST$_{10}$	22.17 ± 0.14	7.88 ± 0.70	13.43 ± 0.26	19.71 ± 2.57	9.44 ± 0.75	15.59 ± 0.94	10.87 ± 1.97	18.29 ± 1.39	13.19 ± 3.84	13.55
QuIST$_{15}$	21.90 ± 0.10	7.20 ± 0.75	13.49 ± 0.27	20.46 ± 2.52	9.15 ± 0.38	15.34 ± 1.38	11.26 ± 1.07	17.34 ± 1.37	13.83 ± 3.05	13.51
GPT-3.5-turbo$_{zero}$	12.27	7.76	11.53	11.84	7.53	11.25	5.40	10.90	4.59	8.85
GPT-3.5-turbo$_{10}$	15.50	7.77	12.40	15.45	7.30	12.84	7.82	11.55	5.30	10.05

Table 11: Automatic evaluation results using BLEU4.

Model	en	bn	de	fi	hi	id	ko	sw	te	Avg
Baseline$_{EncDec}$	50.98	6.95	16.09	21.72	6.29	25.25	10.38	22.85	13.06	15.32
Baseline$_{Enc}$	50.68	22.21	31.23	27.92	27.73	35.10	17.78	25.79	23.05	26.35
Baseline$_{Multi}$	50.99	11.68	24.88	23.16	18.24	28.36	14.99	18.76	16.93	19.63
Baseline$_{Adapter}$	48.11	24.96	31.30	29.47	33.47	36.57	16.04	28.03	23.50	27.92
QuIST$_1$	48.67 ± 0.12	21.69 ± 1.60	30.57 ± 0.74	34.14 ± 2.87	25.15 ± 3.01	36.99 ± 1.74	17.26 ± 1.01	31.73 ± 3.01	22.09 ± 1.69	27.45
QuIST$_5$	48.56 ± 0.14	23.66 ± 1.23	31.39 ± 0.40	41.57 ± 1.60	25.36 ± 2.69	40.85 ± 1.07	19.94 ± 4.24	40.19 ± 3.25	27.59 ± 3.99	31.32
QuIST$_{10}$	48.51 ± 0.19	25.22 ± 1.28	31.33 ± 0.40	41.78 ± 1.59	28.85 ± 1.60	41.66 ± 1.96	24.74 ± 3.52	43.89 ± 1.31	30.62 ± 3.18	33.51
QuIST$_{15}$	48.22 ± 0.12	24.49 ± 1.45	31.43 ± 0.47	42.38 ± 2.64	29.51 ± 0.79	42.38 ± 2.39	27.65 ± 2.47	43.15 ± 1.80	32.65 ± 1.77	34.21
GPT-3.5-turbo$_{zero}$	47.61	27.08	35.50	41.48	28.84	45.81	23.19	41.10	24.16	33.40
GPT-3.5-turbo$_{10}$	49.29	26.82	37.43	44.72	30.16	47.05	27.98	40.96	27.49	35.33

Table 12: Automatic evaluation results using METEOR.

Model	en	bn	de	fi	hi	id	ko	sw	te	zh	Avg
Baseline$_{EncDec}$	44.25	0.72	10.11	14.48	2.11	13.33	2.17	16.07	3.92	27.63	10.06
Baseline$_{Enc}$	44.45	14.53	25.00	19.95	23.45	20.37	11.76	16.72	14.79	40.83	20.82
Baseline$_{Multi}$	41.84	6.23	19.11	15.65	15.12	15.92	7.92	13.65	8.72	30.93	14.81
Baseline$_{Adapter}$	44.16	19.29	23.44	20.26	31.41	22.73	15.75	21.09	22.21	44.60	24.53
QuIST$_1$	43.48 ± 0.04	14.96 ± 2.05	25.75 ± 0.87	27.73 ± 3.87	21.82 ± 3.50	23.06 ± 2.14	11.51 ± 1.07	20.84 ± 2.44	10.44 ± 3.22	42.40 ± 2.32	22.06
QuIST$_5$	43.47 ± 0.07	17.47 ± 1.49	26.80 ± 0.61	37.89 ± 2.37	22.44 ± 3.08	27.04 ± 1.09	15.90 ± 5.63	27.82 ± 3.56	20.57 ± 7.14	46.09 ± 2.24	26.89
QuIST$_{10}$	43.40 ± 0.11	20.23 ± 1.14	27.08 ± 0.52	38.36 ± 1.92	27.26 ± 1.78	28.32 ± 1.76	23.86 ± 2.51	31.32 ± 2.38	29.98 ± 3.29	47.82 ± 0.61	30.47
QuIST$_{15}$	43.08 ± 0.06	19.07 ± 1.47	26.84 ± 0.49	38.79 ± 3.36	27.56 ± 0.63	28.36 ± 2.63	25.14 ± 1.69	30.59 ± 1.39	30.74 ± 2.02	47.71 ± 0.41	30.53
GPT-3.5-turbo$_{zero}$	33.98	21.30	27.76	35.55	24.84	31.18	18.56	27.90	17.31	41.67	27.34
GPT-3.5-turbo$_{10}$	37.63	21.51	29.49	39.41	26.60	32.54	22.28	30.12	23.13	44.47	29.95

Table 13: Automatic evaluation results using ROUGE-L and SP-ROUGE.

Model	bn	de	fi	hi	id	ko	sw	te	zh	Avg
GPT-3.5-turbo$_{10}$	21.51	**29.49**	**39.41**	**26.60**	32.54	**22.28**	30.12	23.13	**44.47**	29.95
w/ QTC & Target language Question Exemplars	**21.97**	28.08	38.99	26.01	**34.63**	20.15	**32.43**	**26.46**	43.16	**30.21**

Table 14: Performance of GPT-3.5-turbo$_{10}$ employing the QTC model and question exemplars in target languages.