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

Anonymous ACL submission

Abstract

Automatic question generation (QG) is used for various purposes, such as building question answering (QA) corpora, creating educational materials, and developing chatbots. However, despite its significance, the majority of existing datasets primarily focus on English, leaving a notable gap in data availability for other languages. Cross-lingual transfer for QG (XLT-QG) has addressed this concern by enabling the utilization of models trained with source language data in other languages. In this paper, we introduce a straightforward and efficient XLT-QG approach that enables the QG model to learn interrogative structures in the target language during inference. Our model is trained to leverage the interrogative patterns found in the given question exemplars to generate questions, using only English QA data. Experimental results demonstrate that the proposed method surpasses various XLT-QG baselines and achieves comparable performance to GPT-3.5-turbo. Moreover, the synthetic data generated by our models proves beneficial for training multilingual QA models. With significantly fewer parameters compared to large language models and without the need for additional training for new languages, our method offers an effective solution for performing QG and QA tasks across diverse languages.[1]

1 Introduction

Automatic question generation (QG) aims to generate questions from a given context. QG models have been utilized not only to augment question answering (QA) datasets but also to generate educational materials and develop chatbots, and various types of 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 QA datasets are in English, leaving a notable lack of data in languages other than English. Moreover, translating English datasets into other languages or crafting new QA dataset, despite the existence of similar English dataset, is deemed inefficient in terms of both time and financial resources.

Recently, researchers have focused on cross-lingual transfer (XLT) to solve these data deficiency 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 in cases where there is a limited or nonexistent availability of annotated data in the target language. Prior 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, integrating language-specific data during model training leads to inflexibility in language scalability, necessitating additional training efforts for application in new languages.

Furthermore, in recent years, multilingual large language models (mLLMs), such as the GPT series[2], BLOOM (Workshop et al., 2022), and PaLM (Chowdhery et al., 2023), have demonstrated remarkable performance across various natural language generation (NLG) tasks, often achieving high efficacy through zero or few-shot inference techniques. Yet, there remains a significant cost burden associated with utilizing commercial APIs, and employing open-source LLMs also necessitates substantial computing resources.

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 lan-

[1] We release our code and question exemplars used in the experiments at https://anonymous.4open.science/r/QuIST-DD51

[2] https://openai.com

guages (**QuIST**). Drawing inspiration from in-context learning (Brown et al., 2020), where the language model acquires knowledge from the input sequence and generates output without parameter updates, our model learns interrogative structures from question exemplars of the target language during inference. By training the model solely with English data, we ensure that our model generates questions in other languages without additional training.

QuIST comprises two stages: 1) Question Type Classification (QTC) and 2) QG utilizing question exemplars. We categorize questions into eight types based on English interrogative words (e.g., who, when, and where), and the QTC model determines the type of question to be generated considering the input context and answer. Once the question type is determined, it is utilized to select the question exemplars for the QG stage. With input comprising a context, answer, and question exemplars, the QG model generates a question. During the training stage using English data, the QG model learns to recognize the interrogative structure specific to the type of question from the provided question exemplars. This strategy empowers the model to generate questions that are not only semantically linked to the input context and answer but also syntactically akin to the question exemplars.

In our experiments, we evaluate the performance of QuIST across nine linguistically diverse languages. Through both automatic and human evaluation, we demonstrate that QuIST outperforms the 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 in target languages from question exemplars during inference.

- QuIST demonstrates high language scalability, as it can be readily applied to new languages with only a few question exemplars, without requiring additional parameter updates.

- QuIST generates questions of quality comparable to those generated by GPT-3.5-turbo, 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 proves to be more beneficial for data augmentation for multilingual QA compared to GPT-3.5-turbo.

2 Cross-lingual Transfer for Automatic Question Generation

In text classification tasks, the zero-shot XLT approach, which utilizes multilingual pretrained language models (mPLMs) fine-tuned solely on English data for the target language, has demonstrated promising performance (Conneau and Lample, 2019; Li and Murray, 2023). However, in NLG tasks, employing this transferring method results in catastrophic forgetting of the target language. To address this issue, Maurya et al. (2021) fine-tuned only the encoder layers of the mPLM while keeping the word embeddings and all the parameters of decoder layers 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: Examples of questions generated by mPLMs fine-tuned using English QA data. The questions typically include English interrogative expressions such as "How long" and "When did."

In our preliminary investigation, we observed that this training technique did not entirely mitigate code-switching in XLT-QG as shown in Figure 1. In particular, the models exhibited a deficiency in understanding interrogative structures in the target language, which we term "interrogative code-switching". In this study, we explore a method enabling the model to grasp interrogative structures without relying on data from the target languages in the training phase.

As depicted in Figure 2, we divide the task into two stages. In the QTC stage, the classification model determines the type of question to be generated. We focus on Wh-questions and classify

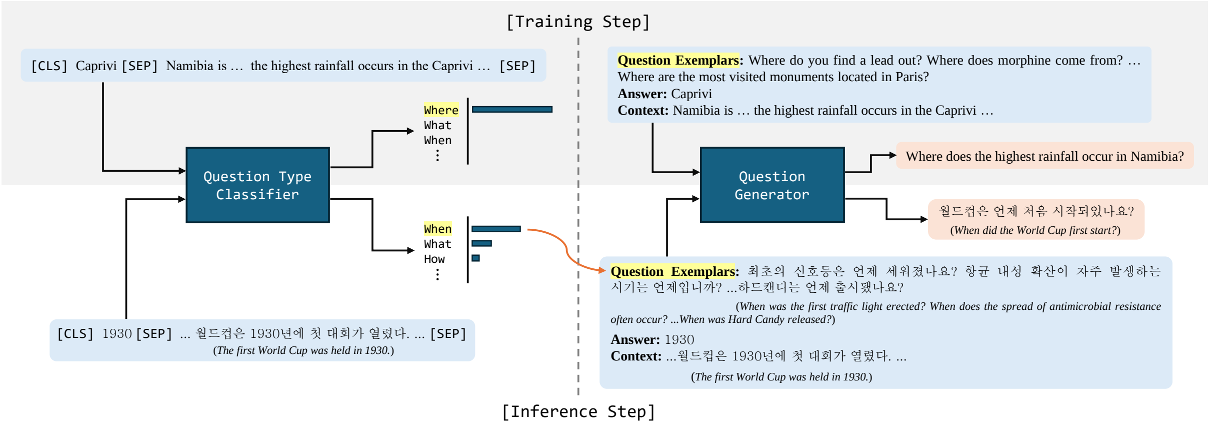
[Training Step]
[CLS] Caprivi [SEP] Namibia is … the highest rainfall occurs in the Caprivi … [SEP]
Question Type Classifier
Where
What
When
⋮
When
What
How
⋮
[CLS] 1930 [SEP] ... 월드컵은 1930년에 첫 대회가 열렸다. ... [SEP]
(The first World Cup was held in 1930.)
Question Exemplars: Where do you find a lead out? Where does morphine come from? … Where are the most visited monuments located in Paris?
Answer: Caprivi
Context: Namibia is … the highest rainfall occurs in the Caprivi …
Question Generator
Where does the highest rainfall occur in Namibia?
월드컵은 언제 처음 시작되었나요?
(When did the World Cup first start?)
Question Exemplars: 최초의 신호등은 언제 세워졌나요? 항균 내성 확산이 자주 발생하는 시기는 언제입니까? ...하드캔디는 언제 출시됐나요?
(When was the first traffic light erected? When does the spread of antimicrobial resistance often occur? ...When was Hard Candy released?)
Answer: 1930
Context: ...월드컵은 1930년에 첫 대회가 열렸다. ...
(The first World Cup was held in 1930.)
[Inference Step]

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.

the questions into eight types based on English interrogative words. While the type of question is primarily influenced by the type of answer, the model takes into account both the answer and context. This is essential because the same answer can yield various types of questions based on the context. For example, the number "911" could represent a count of objects, a historical year, or a proper noun.

The set of question exemplars corresponding to the question type determined by the QTC model is utilized in the QG stage. The question exemplars are pre-created for each question type and language, as explained in Section 3.1 for further details. Using the question exemplars, answer, and context, the QG model generates questions by leveraging the shared interrogative structure among the exemplars. The QTC and QG models are trained exclusively on English QA data and then deployed to new languages without requiring 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}.[3] To train the QTC model, we first annotate the types of questions in the English QA dataset. We only use questions that start with interrogative words and categorize them based on these words[4]. In particular, the questions beginning with "how" are classified into either How_{way} if the subsequent word is an auxiliary verb, or How_{number} if it is an adjective or adverb.

In this stage, we employ the zero-shot XLT approach, wherein the model trained only on English data is directly utilized for other languages. 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 processed through the feed-forward layer, followed by the softmax function, to calculate probabilities for the eight question types. We employed the cross-entropy loss between the predicted and ground-truth labels to update all model parameters. During the inference step in target languages, the fine-tuned model predicts the question type by taking into account the answer and context in those languages.

2.2 Question Generation with Question Exemplars

We use mT5 (Xue et al., 2021) as the backbone of our QG model and approach this task as sequence-to-sequence prediction. The model is trained to generate the ground-truth question given the question exemplars, answer, and context using the teacher-forcing technique. During training, the model learns to utilize syntax information from the question exemplars to generate questions that are also semantically appropriate for the given context and answer. During inference, the question exemplars corresponding to the question type predicted by the QTC model are fed into the QG model, aid-

[3] How_{way}-questions inquire about the manner of how something is done and How_{number}-questions seek information regarding a degree or a specific number.

[4] We only used the examples that fall into one of the eight types.

ing in understanding 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. Training details and evaluation metrics are explained in Appendix A and B.1.

3.1 Data

QA Datasets We utilized SQuAD1.1 (Rajpurkar et al., 2016) as the English QA data (***C-Q-A***$_{en}$) to train both QTC and QG models. For evaluation purposes, 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. Refer to Appendix C for more detailed information, including the number of examples and statistics on question types.

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. 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 five sets of 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 with several XLT-QG models that share the same backbone, mT5. All baselines treat the QG task as a sequence-to-sequence prediction, wherein the models are trained to generate the question given the concatenation of the input answer and context. The datasets used by each model for training and inference are summarized in Appendix C.

Baseline$_{EncDec}$ This baseline model was simply trained by fine-tuning all parameters of mT5 using *C-Q-A*$_{en}$. We employ this baseline to prove that updating the word embeddings and parameters in the decoder layers using English data leads to catastrophic forgetting for other languages.

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 mPLM, 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[5], 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.

4 Main Results

Comparison with Baselines Table 1 presents the performance of QuIST and baselines on nine target languages. According to the results, QuIST$_{15}$, which achieved the highest performance among our models using different numbers of question exemplars, outperforms several XLT-QG baselines,

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

Table 1: 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 E.

showing a margin of 6.00 points when 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 employed in new languages without the need for additional training. Therefore, QuIST stands out for its parameter efficiency and simplicity.

QuIST notably outperforms Baseline$_{Enc}$ across all languages. Interestingly, both models have an equal number of trainable parameters during the fine-tuning stage. From this result, we confirm that learning the interrogative structure of the target language from a small number of question exemplars is beneficial for generating high-quality questions.

Despite Baseline$_{Multi}$ learning questions in the target language via the denoising task, it displayed 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, thereby 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 D.

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 Hindi, Korean, Telugu, Swahili, and Chinese, while performing slightly behind in the remaining languages. 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 F.

	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 2: Human evaluation results.

Human Evaluation We conducted human evaluation in six languages where QuIST and GPT-

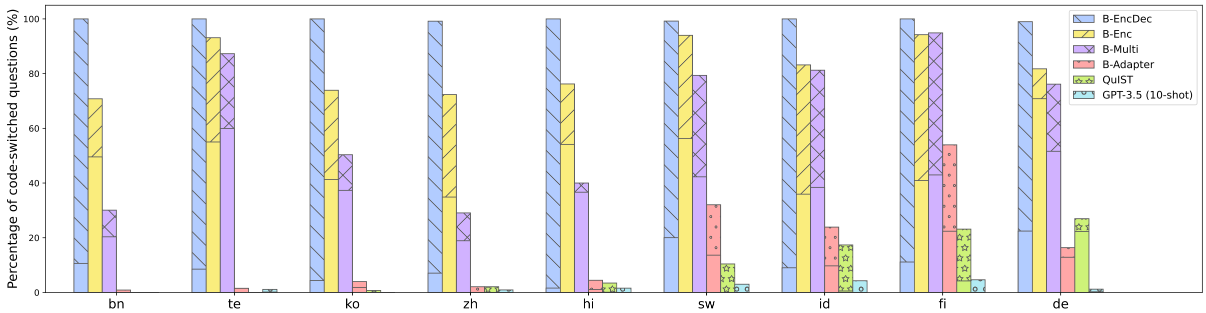

Figure 3: Percentage of questions that contain non-target languages. The lower part of the bar indicates the proportion of occurrences of interrogative code-switching.

3.5-turbo$_{10}$ exhibit similar automatic evaluation results, and we also evaluated the strongest baseline, Baseline$_{Adapter}$. We collected a total of 240 questions generated by the three models per language, then asked three native speakers to evaluate 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 for the remaining categories, the response was either yes or no. More information regarding these criteria is described in Appendix B.2.

Table 2 reports the majority responses from the three raters. In German, Finnish, and Indonesian, synthetic questions of QuIST and GPT-3.5-turbo$_{10}$ consistently achieved high scores across all criteria. Specifically, both models successfully generate questions appropriate for the given answers compared to Baseline$_{Adapter}$. In Bengali and Hindi, our model achieves lower overall scores compared to the previously mentioned languages. However, this performance degradation 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, considering that Baseline$_{Adapter}$ generates questions of significantly lower quality, despite outperforming all other baselines in automated evaluation, it is a meaningful finding 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 models[6]. 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 observed in low-resource languages such as Bengali and Swahili.

5.2 Data Augmentation for Question Answering

QA Data Synthesis Method	Average EM
English-only	49.86
Baseline$_{Enc}$	58.62
Baseline$_{Adapter}$	56.84
Prompt-tuned PaLM	59.54
GPT-3.5-turbo$_{10}$	57.79
QuIST	**59.65**

Table 3: Performance of multilingual QA models.

We examined the potential usefulness of QuIST in augmenting training data for multilingual QA

[6] We utilized cld3 (`https://github.com/google/cld3`) to identify the languages.

models[7]. We compared synthetic data from QuIST and baseline models to the multilingual QA dataset generated by Agrawal et al. (2023) using their prompt-tuned PaLM-540B model (QAMELEON). Table 3 presents the average EM scores across six languages (*bn, fi, id, ko, sw, te*) for the multilingual QA models. The training details can be seen in Appendix A.

According to the results, QuIST shows the best performance, outperforming GPT-3.5-turbo and prompt-tuned PaLM-540B. Interestingly, in contrast to the findings from automatic evaluation and interrogative code-switching analysis, Baseline$_{Enc}$ shows better efficacy in QA data augmentation compared to Baseline$_{Adapter}$. In the previous experiment, code-switching problems were observed in more than 70% of the questions generated by Baseline$_{Enc}$. However, unlike Baseline$_{Adapter}$, which relies solely on task-specific adapters to learn the QG task, Baseline$_{Enc}$ utilized all parameters in the encoder. Therefore, it is presumed that Baseline$_{Enc}$ is able to generate semantically higher quality questions.

5.3 Impact of Different Question Exemplars

Model	Characteristic of Question Exemplars		RL
	Train (en)	*Inference (tgt)*	
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 4: Performance of XLT-QG models using question exemplars in different ways.

We investigated the impact on performance when utilizing question exemplars constructed in ways different from our proposed approach. We compared these approaches to Baseline$_{Enc}$, wherein only the encoder is fine-tuned with English data without using additional data in target languages during training and inference. Table 4 shows the average ROUGE-L (RL) scores across nine languages of each model.

(1) QuIST employs human-written question exemplars of target languages during inference. Additionally, we assess the model's performance by employing exemplars translated from English questions using the Google Translation API. According to the result, exemplars obtained through machine translation are also beneficial for the generation of target language questions when compared to Baseline$_{Enc}$, although they do not exhibit as much effectiveness as when the human-written exemplars are used.

(2) We conducted training and inference using exemplars that cover all question types to ascertain the effectiveness of utilizing type-specific question exemplars. The exemplars consisted of two instances of each of the eight question types, and the QTC model was not employed in this setting. According to the results, there is a slight performance improvement compared to Baseline$_{Enc}$; however, the effect is marginal.

(3) We investigated whether input question exemplars in the inference stage are beneficial even without the training process for generating questions using question exemplars. The model was trained to generate a question from the given context and answer without utilizing the question exemplars, similar to Baseline$_{Enc}$, and only used the exemplars in the inference stage. The results show that the model exhibits lower performance compared to Baseline$_{Enc}$, which suggests that QuIST is trained to utilize question exemplars for QG.

5.4 Question Type Classification

Labeling Type	*en*	*tgt* (Avg)
Hard	62.92	52.86
Relaxed	96.38	91.13

Table 5: Performance of the QTC model.

To measure the zero-shot inference performance of the QTC model for the target languages, we initially annotated the ground-truth question types of the target language QA data. We translated the questions into English and conducted annotation as explained in Section 2.1 (i.e., hard labeling). Table 5 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 starting with "what" and "which,"[8] 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

[7] We generated questions based on the context and answer pairs within the dataset generated by QAMELEON.

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

90% of questions. Thus, it can be inferred that the error propagation resulting from misclassification in QTC is minor in the whole pipeline of QuIST.

5.5 Case Study

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.

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 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.

6 Related Work

Prior work on XLT for generation tasks has focused on training models using source language datasets while maintaining generation proficiency in the target language. To achieve this, Mallinson et al. (2020) and Chi et al. (2020) utilized parallel corpora to enhance alignment between source and target languages, enabling a more effective transfer of task-related knowledge. More recently, numerous researchers have investigated methods that enable models to learn language-specific and language-agnostic knowledge separately (Wang et al., 2021; Wu et al., 2022b; Deb et al., 2023; Pfeiffer et al., 2023).

Unlike most generation tasks, which typically produce declarative sentences, QG faces the challenge of generating interrogative sentences aimed at seeking specific information. In contrast to our approach, which abstains from training models with data from the target language, the majority of previous studies have relied on such data. Kumar et al. (2019) employed a blend of English Question-Answer (QA) data along with a restricted quantity of target language data. On the other hand, Shakeri et al. (2021) trained its model on a denoising task utilizing question corpus in the target language. Agrawal et al. (2023) fine-tuned the PaLM model with 540 billion parameters using five sets of target language QA data. Chi et al. (2020) further utilized language modeling with parallel corpora and restricted its vocabulary solely to tokens from the target language during the question decoding phase.

7 Conclusion

This paper introduces a straightforward and efficient XLT-QG method that utilizes English QA data and a small number of question exemplars in the target languages. Our model is trained to generate questions by leveraging the interrogative structures learned from the question exemplars. With this capability, it proficiently generates questions in a new language. Experimental results demonstrate that our method significantly outperforms XLT-QG baselines and achieves comparable results to GPT-3.5-turbo. Furthermore, we validate the effectiveness of our method's synthetic data for training multilingual QA models. Our approach exclusively utilizes English QA data during training, enabling scalability and parameter efficiency as it can seamlessly extend to new languages without additional parameter updates. Moreover, compared to LLMs, our method employs smaller-sized backbone models, making it easily deployable at a lower cost and with minimal computing power.

8 Limitations

The applicability of our model is restricted to languages on which the mPLMs had been pre-trained. However, it's noteworthy that the mT5 model utilized in our study was pre-trained on a wide spectrum of languages, totaling 101 in number. In addition, the phenomenon of interrogative code-switching is still present in some of the questions generated by QuIST. However, this can be addressed through a simple rule-based filtering method.

References

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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*.

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.

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 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[9]. 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 and reported the average scores obtained from utilizing three different random seeds.

B Metric

B.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), and SP-ROUGE (Vu et al., 2022) 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 that using SentencePiece subword tokenization (Kudo and Richardson, 2018).

B.2 Human Evaluation

We enlisted three native speakers for each language via Upwork[10] 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.

C Data Usage

We utilized SQuAD1.1 (Rajpurkar et al., 2016) as the English QA data $C\text{-}Q\text{-}A_{en}$ for training our models. As only training and validation sets are publicly available, we partitioned the training set

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

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

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 6.

What	Who	How-num	When	Which	Where	How-way	Why
33,777	7,951	5,657	4,780	3,931	2,953	1,600	1,054

Table 6: 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
Telugu	te	5,563	669
Swahili	sw	2,755	499

Table 7: 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 7 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 because XQuAD does not contain a training dataset. Training and test examples in other languages were obtained from TyDiQA (Clark et al., 2020).

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 8: Dataset used by QuIST and baselines.

Table 8 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-specific models were trained for these two baselines.

D Prompt Template for GPT-3.5-turbo

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)	41.13 (1.71)
	5	19.09 (0.85)	54.02 (1.11)	41.43 (1.27)
	10	**19.42** (1.02)	**54.37** (0.69)	**42.10** (1.01)

Table 9: Performance of GPT-3.5-turbo on the SQuAD1.1 validation set. The results of few-shot inference are presented in the form of *mean* (*standard deviation*).

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 9), 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.

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.

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.

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.

E Automatic Evaluation Results

Table 10, 11, and 12 show detailed results for the experiments in Section 4.

F 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 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 13, 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	te	sw	AVG
Baseline$_{EncDec}$	23.45	0.00	3.62	2.91	0.35	5.59	0.00	0.97	4.46	2.24
Baseline$_{Enc}$	23.72	5.64	13.57	6.27	10.01	10.11	4.38	3.64	5.80	7.43
Baseline$_{Multi}$	23.45	2.04	9.38	3.17	3.63	6.46	1.85	1.77	2.35	3.83
Baseline$_{Adapter}$	21.79	6.96	11.34	5.57	12.28	9.10	4.41	6.41	6.38	7.81
QuIST$_1$	22.32 ± 0.06	5.18 ± 0.72	13.02 ± 2.04	12.81 ± 0.88	8.24 ± 2.18	2.54 ± 1.50	3.41 ± 1.05	12.97 ± 0.39	7.78 ± 1.31	8.24
QuIST$_5$	22.20 ± 0.13	6.62 ± 0.97	20.50 ± 1.54	14.78 ± 0.79	15.07 ± 2.87	5.57 ± 4.21	9.38 ± 4.27	13.43 ± 0.30	7.84 ± 1.19	11.65
QuIST$_{10}$	22.17 ± 0.14	7.88 ± 0.70	19.71 ± 2.57	15.59 ± 0.94	18.29 ± 1.39	10.87 ± 1.97	13.19 ± 3.84	13.43 ± 0.26	9.44 ± 0.75	13.55
QuIST$_{15}$	21.90 ± 0.10	7.20 ± 0.75	20.46 ± 2.52	15.34 ± 1.38	17.34 ± 1.37	11.26 ± 1.07	13.83 ± 3.05	13.49 ± 0.27	9.15 ± 0.38	13.51
GPT-3.5-turbo$_{zero}$	12.27	7.76	11.53	11.84	7.53	11.25	5.40	4.59	10.90	8.85
GPT-3.5-turbo$_{10}$	15.50	7.77	12.40	15.45	7.30	12.84	7.82	5.30	11.55	10.05

Table 10: Automatic evaluation results using BLEU4.

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

Table 11: Automatic evaluation results using METEOR.

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

Table 12: Automatic evaluation results using ROUGE-L.

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

Table 13: Performance of GPT-3.5-turbo$_{10}$ employing the QTC model and question exemplars in target languages.