QAMELEON: Multilingual QA with Only 5 Examples

Priyanka Agrawal[1]* Chris Alberti[2]* Fantine Huot[3] Joshua Maynez[1]
Ji Ma[5] Sebastian Ruder[4] Kuzman Ganchev[2] Dipanjan Das[2] Mirella Lapata[1]
[1]Google DeepMind, UK [2]Google DeepMind, USA
[3]Google DeepMind, The Netherlands [4]Google DeepMind, Germany [5]Google Research, UK
{priyankagr,chrisalberti,fantinehuot,joshuahm,
maji,ruder,kuzman,dipanjand,lapata}@google.com

Abstract

The availability of large, high-quality datasets has been a major driver of recent progress in question answering (QA). Such annotated datasets, however, are difficult and costly to collect, and rarely exist in languages other than English, rendering QA technology inaccessible to underrepresented languages. An alternative to building large monolingual training datasets is to leverage pre-trained language models (PLMs) under a few-shot learning setting. Our approach, QAMELEON, uses a PLM to automatically *generate* multilingual data upon which QA models are fine-tuned, thus avoiding costly annotation. Prompt tuning the PLM with only five examples per language delivers accuracy superior to translation-based baselines; it bridges nearly 60% of the gap between an English-only baseline and a fully-supervised upper bound fine-tuned on almost 50,000 hand-labeled examples; and consistently leads to improvements compared to directly fine-tuning a QA model on labeled examples in low resource settings. Experiments on the TYDIQA-GOLDP and MLQA benchmarks show that few-shot prompt tuning for data synthesis scales across languages and is a viable alternative to large-scale annotation.[1]

1 Introduction

Question answering (QA) has seen impressive progress in recent years enabled by the use of large pre-trained language models (Devlin et al., 2019; Lewis et al., 2020a; Raffel et al., 2020), and the availability of high-quality benchmarks (Rajpurkar et al., 2016; Trischler et al., 2017; Kwiatkowski et al., 2019). Many QA datasets frame the task as reading comprehension, where the question is about a paragraph or document and the answer is a span therein. Advances in QA modeling have been primarily reported for English, which offers a considerable amount of high-quality training data compared to other languages. More recently, efforts have focused on the creation of *multilingual* QA benchmarks such as TYDI QA (10 languages; Clark et al., 2020), MLQA (6 languages; Lewis et al., 2020b), and XQUAD (10 languages; Artetxe et al., 2020b). Among these, only TYDI QA is genuinely large-scale, MLQA and XQUAD are limited to an evaluation set due to the high cost and labor required to collect data across languages.

As a result, efforts to localize QA models to new languages have been primarily focusing on *zero-shot* approaches. Recent proposals include using machine translation to approximate training data for supervised learning (Lewis et al., 2020b), and data augmentation via generating synthetic questions for new languages (Riabi et al., 2021; Shakeri et al., 2021). Both approaches rely on transfer from English, which leads to a dependence on translation artifacts (Koppel and Ordan, 2011; Artetxe et al., 2020a) and a bias towards the linguistic characteristics of English, which is not the best source for all target languages (Lin et al., 2019). However, annotating a minimally sized data sample can potentially overcome these limitations while incurring significantly reduced costs compared to full dataset translation (Garrette and Baldridge, 2013).

In this paper, we argue that a few-shot approach in combination with synthetic data generation and existing high-quality English resources can mitigate some of the above-mentioned artifacts. Beyond question answering, multilingual approaches

*Equal contribution. See Contributions section for details.

[1]We release the multilingual QA synthetic data used for fine-tuning them at https://github.com/google-research-datasets/QAmeleon.

Transactions of the Association for Computational Linguistics, vol. 11, pp. 1754–1771, 2023. https://doi.org/10.1162/tacl_a_00625
Action Editor: Lidong Bing. Submission batch: 5/2023; Revision batch: 8/2023; Published 12/2023.

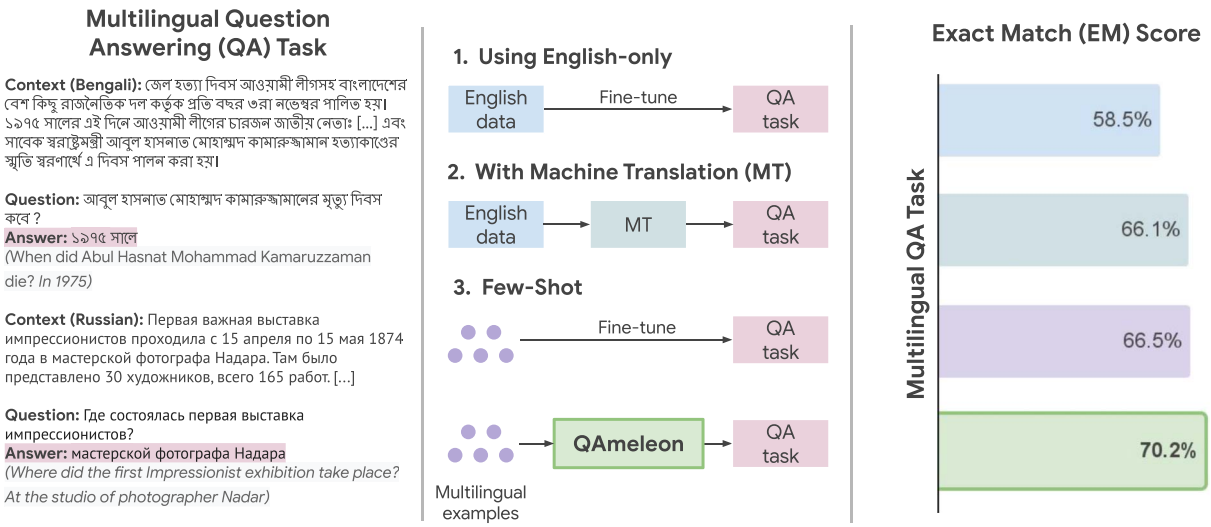


Figure 1: Synthetic data generation for multilingual question-answering (QA). Left: Examples of the multilingual QA task. Translations are added for readability. Middle: Strategies for localizing QA models to new languages: 1. Using English QA data as a zero-shot approach, 2. with Machine Translation (MT) to approximate training data for supervised learning, and 3. few-shot approaches with a handful of multilingual examples. Right: Model performance on the multilingual QA task. We report average Exact Match (EM) across all languages on the TyDiQA-GoldP dataset (Clark et al., 2020).

have succeeded at leveraging a small number of annotations within a variety of tasks (Zhao et al., 2021, *inter alia*) including natural language inference, paraphrase identification, and semantic parsing (Sherborne and Lapata, 2022). Existing work (Brown et al., 2020; Schick and Schütze, 2021, *inter alia*) has further shown that prompting pre-trained large language models (PLMs) can lead to strong performance on various tasks, including question answering (Khashabi et al., 2020; Chowdhery et al., 2022) and open-ended natural language generation (Tang et al., 2022; Yang et al., 2022). Investigations of prompting in multilingual settings have also shown strong few-shot performance in classification tasks (Winata et al., 2021), natural language inference (Zhao and Schütze, 2021), common sense reasoning (Shi et al., 2022), machine translation (Lin et al., 2022), and retrieval (Dai et al., 2022).

We synthesize these directions into QAmeleon, an approach for bootstrapping multilingual QA systems, with as few as five examples in a new target language (see Figure 1). We use gold annotations to prompt-tune a PLM in order to automatically generate multilingual QA data, which is then used to fine-tune a QA model. We find that QAmeleon delivers accuracy superior to zero-shot methods and competitive translation-based baselines, and in some cases competes with the fully supervised upper bound.[2] Experiments on the TyDi QA (Clark et al., 2020) and MLQA (Lewis et al., 2020b) benchmarks show that few-shot prompt tuning (Lester et al., 2021) scales across languages, significantly outperforms prompt engineering (Brown et al., 2020) with the same number of labeled examples, and is a viable alternative to large-scale annotation.

Our contributions include (a) a new approach to bootstrapping a multilingual QA system; QAmeleon prompt-tunes a PLM with as few as five gold examples to automatically generate multilingual QA data which is then used to fine-tune a QA model; (b) a series of experimental results showing significant improvements over existing approaches in the few-shot regime, ranging from 12% absolute accuracy on TyDiQA-GoldP (Clark et al., 2020) over an English-only baseline and 4% absolute accuracy over a competitive translate-train baseline; (c) extensive analysis of the behavior of QAmeleon in zero shot and low resource regimes, on different multilingual QA datasets, and in comparison to prompt-engineering.

[2]This is noteworthy as multilingual models fine-tuned on translated data—also known as translate-train—form the state of the art on most multilingual datasets (Ruder et al., 2021).

2 Synthetic Data Generation

Let $\mathcal{D}_l$ denote a QA dataset with examples provided by human annotators, where l is a *target* language in a set L of languages of interest. $\mathcal{D}_l$ consists of samples $(c, q, a)_l$, where c is a paragraph of text, q is a question, and a is an answer extracted from c (see Figure 1 left). We further use $\mathcal{D}_{l,n}$ to denote a dataset $\mathcal{D}_l$, but making n explicit, with n referring to the number of examples it contains. For instance, $\mathcal{D}_{\text{fr},5}$ denotes a French QA dataset with 5 examples. Finally, let $\mathcal{U}_l$ denote sets of *unlabeled* paragraphs in language l; we assume these are in-domain with respect to the paragraphs in $\mathcal{D}_l$ but are not accompanied by questions or answers.

Throughout this work, we will assume the availability of $\mathcal{D}_{\text{en}}$, a large QA dataset in English (*source* language). This assumption corresponds to the observation that most large-scale QA datasets (Rajpurkar et al., 2016; Yang et al., 2018; Bajaj et al., 2016; Kwiatkowski et al., 2019) contain examples exclusively in English. For languages other than English, we assume that only small datasets $\mathcal{D}_{l,n}$ are available for training (e.g., $n = 5$) (''Few-Shot'' scenario) or no data at all (''English-Only'' scenario). We will also assume that sets $\mathcal{U}_l$ of unlabeled passages are available for all target languages. Our task will be to synthesize QA data in each *target* language l in order to fine-tune QA models on l directly.

In the rest of this section we formally describe three ways of synthesizing QA data and give further details on the two scenarios we consider, ''English-Only'' and ''Few-Shot''.

2.1 Machine Translation (MT)

A widely adopted approach (Lewis et al., 2020b; Shakeri et al., 2020) makes use of a machine translation system $\mathcal{T}$ to automatically translate text from one language into another. Let $\mathcal{T}_{l'}(\mathcal{D}_l)$ denote the translation of dataset $\mathcal{D}_l$ from language l to language l'. The translation is performed by independently applying $\mathcal{T}$ to context c, question q, and answer a for each example in the source dataset (see approach 2 in Figure 1). A synthetic QA dataset $\mathcal{D}_{\text{MT}}$ is generated by translating the entire English dataset into each language of interest:

$$\mathcal{D}_{\text{MT}} = \mathcal{D}_{\text{en}} \cup \bigcup_{l \in L - \{\text{en}\}} \mathcal{T}_l(\mathcal{D}_{\text{en}}).$$

The approach described here is known as ''translate-train''. An alternative is ''translate-test'', where translation is employed during inference instead of training. Multilingual inputs are translated to English and inference is done via an English QA model. The English predictions are then translated back to the respective target language. We experimentally found ''translate-test'' to perform poorly on our task in comparison to translate-train due to its reliance on multiple noisy translation steps.

Note that fine-tuning on $\mathcal{D}_{\text{MT}}$ still relies on the support of the high-quality $\mathcal{D}_{\text{en}}$. Previous work (Kramchaninova and Defauw, 2022; Vu et al., 2022) has highlighted various limitations with multilingual approaches based on MT including (a) their dependence on the quality of available MT systems in a given language and in turn the availability of high-quality (expensive) parallel data, (b) a potential misalignment of answer spans after translation in context to the passage vs translation of answers independently, and (c) translationese artifacts and English-centric content topics (Clark et al., 2020).

2.2 Prompt Engineering (PE)

PLMs (Brown et al., 2020; Chowdhery et al., 2022) have recently shown unprecedented performance on a vast number of tasks, including natural language generation, without the need for modifying any of the model's parameters, simply by hand-designing a textual prompt that instructs the model to perform a certain task. Following Brown et al. (2020), we consider a class of hand-designed prompts referred to as ''prompting'' or ''in-context learning''. The prompt starts with a free form instruction, followed by a small number of instances exemplifying how the task is solved. An incomplete instance is then appended to this prompt and the PLM performs the task by completing that instance. We refer to this approach as ''prompt engineering'' (PE), since the input to the PLM has to be hand-engineered based on human intuition about the target task (see approach 3 in Figure 1).

In order to hand-engineer prompts for our task, we use a small set of parallel examples $\mathcal{C}_{l,n}$ consisting of passages, questions, and their answers in the English source and target language l. We discuss how we construct these examples shortly. For now, suffice it to say that we create two prompts

for answer and question generation, respectively.[3] Our first prompt is used to obtain an answer a_l in the target language l from passage c_l:

I will write potential answers
for the following passages.
  Passage: c_l
  Answer in English: a_{en}
  Answer in the original language: a_l
...

The second prompt generates question q_l, utilizing passage c_l and the previously predicted answer a_l:

I will write questions and answers
for the following passages.
  Passage: c_l
  Answer: a_l
  Question in English: q_{en}
  Question in the original language: q_l
...

We generate synthetic data instances $(c, q, a)_l$ where a and q are inferred by applying our two prompts consecutively on each passage $c_l \in \mathcal{U}_l$ (recall $\mathcal{U}_l$ is the set of unlabeled passages in target language l).

In the English-Only scenario, neither questions nor answers are available in target language; we obtain these by resorting to machine translation:

$$\mathcal{C}_{l,n}^{\text{en-only}} = \{(\mathcal{T}_l(c), q, a, \mathcal{T}_l(q), \mathcal{T}_l(a)) | (c, q, a) \in \mathcal{D}_{\text{en},n}\},$$

In the ''Few-Shot'' setting, we have access to n-labeled examples (questions and answers) in the target language, and translate these into English:

$$\mathcal{C}_{l,n}^{n\text{-shot}} = \{(c, \mathcal{T}_{\text{en}}(q), \mathcal{T}_{\text{en}}(a), q, a) | (c, q, a) \in \mathcal{D}_{l,n}\}.$$

Let $\mathcal{P}_l^e$ denote this prompting based generation. We can write the generated synthetic dataset as:

$$\mathcal{D}_{\text{PE}} = \mathcal{D}_{\text{en}} \cup \bigcup_{l \in L - \{\text{en}\}} \mathcal{P}_l^e(\mathcal{U}_l).$$

[3]We find that joint answer and question generation using single-stage prompting performs worse in comparison to two-stage generation.

Note that in the composition of the prompt, we always include English as an intermediate or ''bridge'', i.e., asking the model to predict questions and answers in English in addition to the ones in the target language, as we experimentally found it improves the quality of the generated data. The use of a bridge for this task can be thought of as an example of multilingual ''chain of thought'' prompting (Wei et al., 2022).

2.3 QAMELEON (PT)

In this approach, an optimizer is utilized to minimize the cross-entropy loss by updating the PLM's parameters for $P(a, q|c, l)$ over a training set containing examples $(c, q, a)_l$ for the languages in L. As with PE, we generate the training set for the PLM in two ways. For ''English-Only'' we construct the dataset as $\bigcup_{l \in L} \mathcal{T}(\mathcal{D}_{\text{en}})$, while for ''Few-Shot'' we use $\bigcup_{l \in L} \mathcal{D}_{l,n}$.

Given the small size of the training set in the ''Few-Shot'' setting and the large size of current models, we opt for using prompt tuning (PT; Lester et al., 2021), a parameter-efficient fine-tuning variant where we concatenate a soft prompt of length m tokens to the input of the PLM, where m is a hyperparameter always set to 50 in this work. Only the embeddings of these m prompt tokens are allowed to be modified by the optimizer. We note that in prompt tuning, like in prompt engineering, the parameters of the PLM remain unchanged. What is trained is only the short soft prompt that is prepended to the input embeddings at inference time.

We use $\mathcal{P}_l^t$ to denote the operation of generating question-answer pairs through greedy decoding on the prompt-tuned PLM, by taking an unlabeled passage $c_l \in \mathcal{U}_l$ as input, preceded by a few tokens encoding language l. We finally obtain the synthetic QA dataset $\mathcal{D}_{\text{PT}}$ as:

$$\mathcal{D}_{\text{PT}} = \mathcal{D}_{\text{en}} \cup \bigcup_{l \in L - \{\text{en}\}} \mathcal{P}_l^t(\mathcal{U}_l).$$

2.4 Data Assumptions

English-Only In this scenario, only training data in English is available, denoted as $\mathcal{D}_{\text{en}}$. Prompt Engineering (PE) assumes parallel exemplars are available, while Prompt Tuning (PT) requires exemplars in the target language only. Both are possible by translating examples of the English data

$\mathcal{D}_{\text{en}}$ into each target language. Machine Translation (MT) approaches in this work follow this scenario only.

Few-Shot In this scenario, a small number of examples (n-shot) are available in each target language, denoted as $\mathcal{D}_{\text{l},n}$. In this scenario, parallel exemplars for Prompt Engineering (PE) can be obtained by translating the target language data into English. Prompt Tuning (PT) only requires exemplars in the target language, which are readily available in this setting.

3 Experimental Setup

We evaluate the synthetic data generation approaches presented in Section 2 across various languages on two benchmark datasets, which we discuss below. We also describe various model configurations, and comparison systems before presenting our results.

3.1 Datasets

TyDi QA (Clark et al., 2020) is a multilingual extractive question answering dataset designed to represent a typologically diverse set of languages. Annotators were given a Wikipedia passage in the target language and asked to write a question that could not be answered by that passage. For each question, the top-ranked Wikipedia article was then retrieved via Google Search. Annotators were subsequently asked to answer the question given the retrieved Wikipedia article. As a result of this information-seeking task design, questions in TyDi QA are often without an answer. In this work we consider TyDiQA-GoldP: the Gold Passage version of TyDi QA where only questions with answers in the Wikipedia page are given and the model has to identify the answer in the passage that contains it (see Table 1 for statistics on this dataset).

MLQA (Lewis et al., 2020b) is an extractive question answering dataset, designed for evaluating multilingual and cross-lingual question answering models. MLQA does not publish a training split, but only development and test partitions. MLQA was created by aligning sentences in Wikipedia passages across different languages. Annotators then created questions based on English sentences, professional translators translated these questions to other languages, and finally annotators selected answers from passages containing sentences aligned to the translated questions. As in TyDiQA-GoldP, the task is to extract the answer from a passage given a question (dataset statistics are shown in Table 1).

	TyDiQA-GoldP		MLQA	
Language	Train	Eval	Dev	Test
Arabic	14,805	921	517	5,335
Bengali	2,390	113	–	–
Chinese	–	–	504	5,137
English	3,696	440	1,148	11,590
Finnish	6,855	782	–	–
German	–	–	512	4,517
Hindi	–	–	507	4,918
Indonesian	5,702	565	–	–
Kiswahili	2,755	499	–	–
Korean	1,625	276	–	–
Russian	6,490	812	–	–
Spanish	–	–	500	5,253
Telugu	5,563	669	–	–
Vietnamese	–	–	511	5,495
Total	49,881	5,077	4,199	42,245

Table 1: Number of question-answer pairs per language and data split for the datasets considered in this work.

Unlabeled Data We obtained paragraphs $\mathcal{U}_l$ in each target language from Wikipedia. Specifically, we pre-processed Wikipedia pages using WikiExtractor (Attardi, 2015). Paragraphs were sampled uniformly, with a length between 200 and 510 characters. The target language was determined based on the language code of the Wikipedia edition.

3.2 Model Configuration

Synthetic Data Generation In our TyDi QA experiments, we treat the English training data as the English source. For MLQA, we employ the English SQuAD (Rajpurkar et al., 2016) training data as the source. In the Few-Shot scenario, our human-annotated target-language examples $\mathcal{D}_{l,n}$ are taken from the training split of TyDiQA-GoldP and the validation split of MLQA.

For machine translation (MT), we employ the public Google Translate API (Wu et al., 2016)

while the PLM utilized in this work is PaLM-540B (Chowdhery et al., 2022). We perform heuristic checks to clean synthetic datasets $\mathcal{D}_{\text{PE}}$ and $\mathcal{D}_{\text{PT}}$. We only preserve a question-answer pair if the generated answer a is a substring of the given context c, but not a substring of the query q. We perform the first check as both TyDiQA-GoldP and MLQA are extractive QA datasets. We perform the latter check because we empirically found that some of the low quality generated question-answer pairs were trivially answered based on the content of the question alone, for example, q: ''where is X?'', a: ''X''.

In the construction of $\mathcal{D}_{\text{PE}}$, we additionally perform round-trip filtering (Alberti et al., 2019) as qualitative analysis of random QA pairs suggested a higher level of noise in the PE-generated data. This round-trip consistency check is done by comparing the originally generated answer a in $(c, q, a)_l$ with the predicted answer. This predicted answer is obtained by prompting the PLM to answer question q based on passage c. We also tried round-trip filtering for PT generated data, however, we did not observe any gains. We report detailed statistics of the synthetically generated datasets in Section 5.

In the construction of $\mathcal{D}_{\text{PT}}$, we prompt-tune the PLM on $\bigcup_{l \in L} \mathcal{T}(\mathcal{D}_{\text{en}})$ or $\bigcup_{l \in L} \mathcal{D}_{l,n}$ as detailed earlier. Prompt tuning is performed with the AdaFactor optimizer (Shazeer and Stern, 2018). We tune a prompt of length $m = 50$ tokens for up to 1,000 steps, evaluating every 50 steps, with a batch size of 16 examples, and learning rate of 0.3 with a linear warmup of 200 steps. We use early stopping to select the best prompt per language based on BLEU (Papineni et al., 2002) on a held-out dataset from the English TyDiQA-GoldP, translated to each target language.

Question Answering We fine-tuned an mT5-XL model (Xue et al., 2021) for question-answering to evaluate different synthetic data generation methods ($\mathcal{D}_{\text{MT}}$, $\mathcal{D}_{\text{PE}}$, and $\mathcal{D}_{\text{PT}}$). As a baseline, we further use mT5-XL fine-tuned on available training data. Specifically, in the English-Only scenario, Baseline mT5-XL is fine-tuned on the English QA data $\mathcal{D}_{en}$. In the Few-shot scenario, Baseline mT5-XL is fine-tuned on n human annotated examples in the target languages (same number given to PE and PT). We conducted experiments on TyDiQA-GoldP (Clark et al., 2020) and MLQA (Lewis et al., 2020b), see Section 3.1.

Throughout downstream QA evaluation, mT5-XL was fine-tuned with AdaFactor, with a learning rate of 0.0002, a batch size of 64, up to 3,000 and 5,000 steps of training for TyDiQA-GoldP and MLQA, respectively, evaluating every 50 steps. We measure QA performance with Exact Match (EM) and F1, and report the unweighted average across languages (excluding English). For TyDiQA-GoldP, we report results on the development split which is commonly used as an evaluation set since the test split is unavailable. We select mT5 checkpoints per language using EM, and report the average of 3 runs. For MLQA, we present results on the test split, selecting the best mT5 checkpoint based on the average EM on the MLQA dev set.

4 Results

QAmeleon (PT) Delivers the Best QA System Table 2 summarizes our results on TyDi QA for both English-only and Few-Shot scenarios. Overall, we find that a low resource setting with 5 human-annotated examples in the target language ($\mathcal{D}_{l,5}$) is useful for scaling QA to multiple languages. More specifically, 5-shot prompt tuning gives an EM improvement of 11.7% absolute (58.5% $\rightarrow$ 70.2%) in exact match answer accuracy on the TyDiQA-GoldP evaluation set over mT5 fine-tuned on English data only (Baseline), 3.7% (66.5% $\rightarrow$ 70.2%) over mT5 fine-tuned on 5 examples per language (Few-shot Baseline), and 4.1% (66.1% $\rightarrow$ 70.2%) over mT5 fine-tuned on the data obtained with the MT approach.

QAmeleon further improves over the few-shot results obtained by prompting code-davinci-002 (Chen et al., 2021), PaLM-540B (Chowdhery et al., 2022), and Flan-U-PaLM-540B (Chung et al., 2022), with a similar number of available labeled examples. These approaches directly employ extremely large PLMs for the task of QA, whereas QAmeleon leverages data synthesis to distill a PLM into a much smaller mT5-XL model. It also is important to note that QAmeleon as an approach is orthogonal and possibly complementary to any improvements due to more performant QA models and more sophisticated PLMs (e.g., Flan-U-PaLM-540B).

In both English-only and Few-shot resource scenarios, QAmeleon outperforms the other two data generation approaches, Machine Translation

Method	English-Only			Few-Shot		
	Translate	Avg EM	Avg F1	n-Shot	Avg EM	Avg F1
Baseline		$58.5_{(\pm 3.1)}$	$74.2_{(\pm 2.6)}$	5	$66.5_{(\pm 0.7)}$	$79.8_{(\pm 0.4)}$
MT	✓	$66.1_{(\pm 2.1)}$	$79.5_{(\pm 1.8)}$	5	–	–
PE	✓	$64.4_{(\pm 1.4)}$	$76.9_{(\pm 1.1)}$	5	$62.6_{(\pm 1.8)}$	$77.6_{(\pm 1.2)}$
PE + MT	✓	$\mathbf{69.4}_{(\pm 0.4)}$	$\mathbf{81.4}_{(\pm 0.4)}$	5	$67.9_{(\pm 0.2)}$	$80.5_{(\pm 0.6)}$
QAMELEON (PT)	✓	$65.5_{(\pm 0.7)}$	$79.4_{(\pm 0.7)}$	5	$70.2_{(\pm 0.2)}$	$81.7_{(\pm 0.1)}$
QAMELEON (PT)+MT	✓	$68.1_{(\pm 0.8)}$	$80.9_{(\pm 0.7)}$	5	$\mathbf{70.7}_{(\pm 0.9)}$	$\mathbf{82.2}_{(\pm 0.8)}$
code-davinci-002§		–	–	1	48.1	–
PaLM-540B†		–	–	1–10	60.0	–
Flan-U-PaLM-540B‡		–	–	1	68.3	–

Table 2: Synthetic question-answering data generation methods for training multilingual reading comprehension systems on TYDIQA-GOLDP. We report averages over 3 runs of fine-tuning mT5-XL on gold or synthetic data. Standard deviation is given in parentheses. Performance for individual languages (excluding English) is shown in Table 3. For comparison we also include recent few-shot prompting results with large language models on TYDIQA-GOLDP: Chen et al. (2021)§, Chowdhery et al. (2022)†, and Chung et al. (2022)‡.

Method	n-shot	Ar	Bn	Fi	Id	Ko	Ru	Sw	Te	Avg
Baseline	5	65.9	68.4	65.1	71.3	68.4	57.6	60.1	75.4	66.5
MT	0	66.3	62.2	65.2	72.4	63.9	61.1	70.5	67.0	66.1
PE	0	60.4	66.7	63.5	63.6	65.1	53.8	74.5	67.3	64.4
PE + MT	0	**68.1**	70.5	68.2	73.6	**68.5**	61.0	**78.4**	66.9	69.4
QAMELEON (PT)	5	65.4	**76.7**	**69.4**	69.0	67.6	61.5	75.6	**76.7**	70.2
QAMELEON (PT)+MT	5	67.9	72.6	69.2	**73.8**	65.1	**62.8**	77.7	76.1	**70.7**
Supervised	Multi-k	75.7	81.4	74.5	79.8	77.2	72.8	82.6	83.0	78.4
% tokens in PLM	–	*0.15*	*0.03*	*0.42*	*0.16*	*0.19*	*0.53*	*0.01*	*0.02*	–

Table 3: QA performance (Average EM over three runs) for individual languages on the TYDIQA-GOLDP evaluation set; the backbone of the QA model is mT5-XL fine-tuned on gold (Baseline, Supervised) or synthetically generated data. The final row displays the percent of tokens for each language in the PLM training data.

(MT), and Prompt Engineering (PE). Despite employing PE in two stages, chain-of-thought style, we observe that the generated data leads to lower QA performance. Moreover, we see better performance with using English-Only data in comparison to the Few-Shot scenario, suggesting that the PLM is able to better utilize high-quality English data rather than small amounts of labeled data (in other languages). Finally, augmenting PLM generated data (either via PE or PT) with data generated via MT leads to gains in QA performance over using any of these methods independently. This could be due to the coupling of diverse QA data, i.e., language-specific content and task-specific English-centric translated content.

Table 3 shows QA performance in individual languages, for each of the methods in Table 3 in their best performing setting: Few-shot Baseline, Machine Translation (MT), Prompt Engineering (PE), Prompt Tuning (PT), and augmenting PE and PT with MT. Data generated by QAMELEON (PT) using 5 target examples provides the best performance in Bengali, Finnish, and Telugu. A boost can be seen for Arabic, Indonesian, Russian, and Swahili when QAMELEON data is combined with MT data. Language distribution listed under '% tokens in PLM' reflects the extremely

Method	n-Shot	Avg EM
Baseline	1	63.7
QAMELEON (PT)	1	69.7
Baseline	5	66.5
QAMELEON (PT)	5	70.2
Baseline	50	69.3
QAMELEON (PT)	50	73.7
Baseline	100	70.6
QAMELEON (PT)	100	71.9
Supervised	Multi-k	78.4

Table 4: Comparison of QA performance from fine-tuning mT5-XL on 1 to 100 examples (Baseline), on synthetic data generated with prompt tuning (PT), or on the full TYDIQA-GOLDP training set (Supervised). Results are averaged across languages.

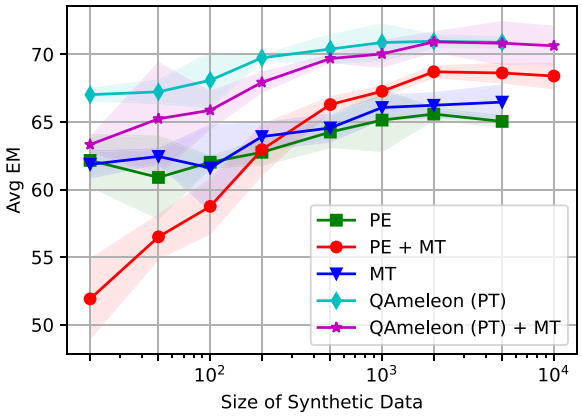


Figure 2: Effect of synthetic data size on downstream QA performance (Average EM on TYDIQA-GOLDP evaluation set); results shown for mT5-XL QA model fine-tuned via Machine Translation (MT), Prompt Engineering (PE), Prompt Tuning (QAMELEON (PT)), and combinations thereof (PE + MT and QAMELEON (PT) + MT).

low representation of many languages in the pre-training corpora of the PLM used in this work. As an upper bound, we additionally show the performance of supervised mT5-XL fine-tuned on large amounts of gold training data (see Table 1) to illustrate the remaining gap, which could potentially be bridged by increasing the number of labeled examples or by improved (e.g., more multilingual or FLAN-tuned) PLMs.

Increasing Labeled Examples Improves QA Performance So far, we have tested QAMELEON in an extremely low resource setting, using only 5 examples in the target language. We next examine its performance when we vary the number of annotated examples. Table 4 compares the performance of mT5-XL fine-tuned on 1 to 100 examples (Baseline), on synthetic QA datasets generated by QAMELEON using corresponding number of examples, and as an upper bound on the entire TYDIQA-GOLDP dataset. As can be seen, increasing the number of examples from 1 to 50 improves the performance of QA models. We observe however a decrease in performance at 100 examples, showing that further research will likely be needed to close the gap between our method and the fully supervised upper bound, while still only labeling a small number of examples. It is important to note that for all the amounts of available annotated data we considered, significant improvements in multilingual QA could be obtained by generating data with QAMELEON instead of fine-tuning the QA model directly on labeled data.

The Larger the Synthetic Data, the Better the QA Model We now study the impact of varying the size of the automatically generated datasets on QA performance. As shown in Figure 2, when larger amounts of synthetic data are used for fine-tuning the QA model, absolute accuracy increases. This upshot is higher when combining PLM-generated data with Translation data in comparison to individual datasets. This can be explained due to the increased diversity of the combined data, which include English-centric translated content and target language-specific content obtained from the PLM. Eventually, we observe a saturation effect, i.e., beyond $O(1,000)$ QA pairs in the target language improvements are limited.

BLEU Does not Correlate with Downstream QA Performance An interesting question is whether improvements in QA performance are due to better (e.g., more grammatical or diverse) questions. We assessed the quality of questions generated by QAMELEON (PT) on TYDIQA-GOLDP by measuring their similarity to gold standard questions. We compare this with an mT5-XL model for question generation fine-tuned in a Few-shot setting. Both QAMELEON (PT) and mT5-XL question generation models were given the same number of examples in each language. Table 5 reports BLEU (Papineni et al., 2002) scores for these two models; we additionally report question answering performance (in terms

Method	n-Shot	Avg BLEU	Avg EM
mT5-XL	5	24.74	57.3
QAMELEON (PT)	5	24.29	70.2

Table 5: BLEU scores and downstream QA performance on TYDIQA-GOLDP for questions generated by mT5-XL and QAMELEON (Few-shot setting, 5 examples in the target language).

of EM) via another set of mT5-XL models fine-tuned on the synthetic data generated by the respective models.

Even though mT5-XL produces questions with slightly higher BLEU score, QAMELEON generates QA data that leads to much higher QA performance. The result underscores the need for better trustworthy automatic evaluation metrics (Sellam et al., 2020) across languages.

Our Results Generalize to MLQA To validate the general applicability of our approach, we evaluate QAMELEON on MLQA (Lewis et al., 2020b). We prompt-tune the PLM on 5 examples per language taken from the MLQA development set, since MLQA does not provide training partitions. We generate synthetic datasets in all of the MLQA languages and compare an English-only baseline, MT, and QAMELEON (PT) approaches as we did previously for TYDIQA-GOLDP. We report results (EM and F1) using mT5-XL as the QA model in Table 6, where English is included in the average performance.

We find that the MT approach is very effective on MLQA, which is not surprising since MLQA questions are translated from English. QAMELEON (PT), however, still delivers an improvement in combination with MT synthetic data. Table 6 further reports comparisons with the state-of-the-art models of Xue et al. (2021) and Chi et al. (2022). The former is mT5-XL (3.7B parameters) fine-tuned on English data only, whereas XLM-E-XL (2.2B parameters) benefits from a different language model pretraining technique. The latter approach is orthogonal and potentially complementary to QAMELEON.

5 Data Analysis

Table 7 shows the size of synthetic data resources generated via Prompt Engineering (PE) and QAMELEON (PT), per language and in total.

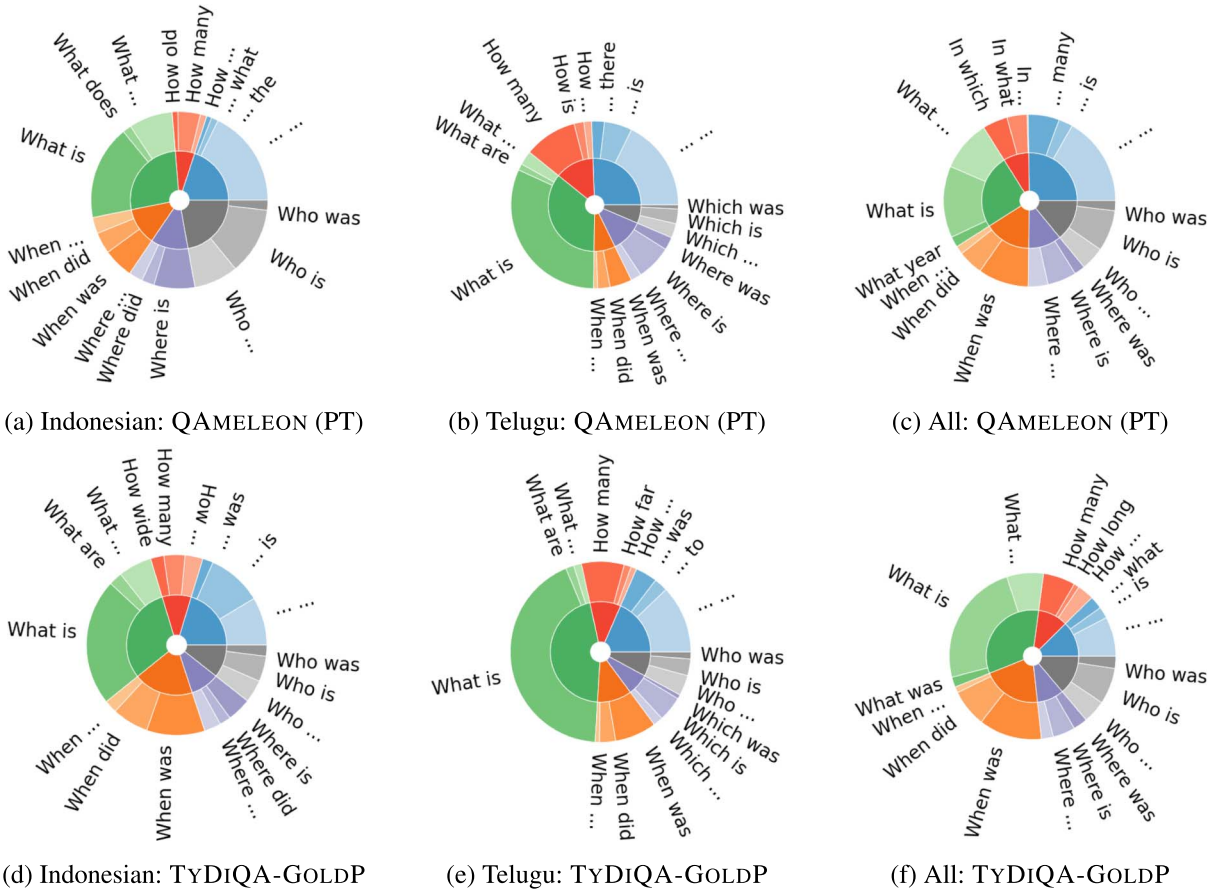
Method	Avg EM	Avg F1
English-Only	53.1	71.8
MT	56.4	74.8
QAMELEON (PT)	55.0	74.3
QAMELEON (PT) + MT	**56.8**	**75.3**
mT5-XL (Xue et al., 2021)	54.5	73.5
XLM-E-XL (Chi et al., 2022)	57.9	76.2

Table 6: Downstream QA performance on the MLQA test set with an mT5-XL model trained on SQUAD English data (English-Only), SQUAD translated to all MLQA languages (MT), on synthetic data generated by QAMELEON (5-shot) in all MLQA languages, or on a combination of data generated by MT and QAMELEON. Results for Xue et al. (2021) and Chi et al. (2022) are taken from the respective papers.

	TYDIQA-GOLDP		MLQA
Language	PE	QAMELEON	QAMELEON
Arabic	5,219	8,499	14,738
Bengali	5,948	8,036	–
Chinese	–	–	14,669
Finnish	8,062	5,943	–
German	–	–	11,186
Hindi	–	–	12,036
Indonesian	6,487	7,810	–
Kiswahili	8,003	8,041	–
Korean	5,229	7,906	–
Russian	5,619	7,441	–
Spanish	–	–	10,134
Telugu	2,742	5,222	–
Vietnamese	–	–	13,333
Total	47,309	52,955	89,344

Table 7: Number of synthetic question-answer pairs per language generated via Prompt Engineering (PE) and QAMELEON (PT) with 5 human-labeled examples.

These were in the range of 47,000–53,000 QA examples for TYDIQA-GOLDP, and 89,000 for MLQA. The varying size of the data across languages is due to the filtering described in Section 3. In some languages (e.g., Telugu) generation is more noisy leading to fewer data points. We conjecture this is due to the PLM being exposed



(a) Indonesian: QAMELEON (PT) (b) Telugu: QAMELEON (PT) (c) All: QAMELEON (PT)

(d) Indonesian: TYDIQA-GOLDP (e) Telugu: TYDIQA-GOLDP (f) All: TYDIQA-GOLDP

Figure 3: Distribution of question category for QAMELEON (PT) generated questions (a,b,c) and TYDIQA-GOLDP training questions (d,e,f). Category are obtained by translating the questions to English with Google Translate and grouping by the first two word tokens.

to less data representing these languages during pre-training. We further hypothesize that a more multilingual pre-training of PLMs could potentially lead to better quality data across all languages.

Machine translation (MT) creates the same number of data points as the source training set. For TYDIQA-GOLDP, the English training contains 3,696 data points (Table 1), leading to approximately 29,000 QA examples across 8 languages. In MLQA, machine translation (MT) uses SQuAD as the English dataset, consisting of $\sim$ 87,000 data points, leading to $\sim$ 522,000 QA examples across 6 languages.

Figure 3 shows the distribution of various question types for individual languages and on average across all languages. For each language, synthetically generated questions were first translated to English and then grouped into categories (inner circle) based on their first word and sub-categories (outer circle) based on the first two words. We find that QAMELEON (Figure 3(c)) generates more diverse questions in comparison to TYDIQA-GOLDP (Figure 3(f)). The distribution of question words varies across languages in both the datasets. For example, diversity is higher for Russian, Finnish, and Indonesian (Figure 3(a,d)); however, for Bengali and Telugu (Figure 3(b,e)), the distribution of questions is skewed towards a specific question type ('What' for Telugu and 'Other' for Bengali). This could be attributed to a lack of diversity in questions for these languages or poor translation quality leading to skewed utterances.

Table 8 illustrates randomly sampled examples of QA pairs generated by QAMELEON (PT) for passages in the TYDIQA-GOLDP eval set. For these passages, we also have access to human annotated QA pairs. As can be seen, QA pairs generated by QAMELEON are of similar quality and at times more diverse compared to the human-annotated dataset. Table 9 illustrates examples of QA pairs generated by QAMELEON from randomly selected Wikipedia passages.

Lang	Human Annotated		QAMELEON (PT)	
ar	**Q:** متى تأسست جامعة فرايبورغ؟	**A:** 1457	**Q:** متى تأسست جامعة فرايبورغ؟	**A:** 1457
	Q: When was the University of Freiburg established?	*A: 1457*	*Q: When was the University of Freiburg founded?*	*A: 1457*
sw	**Q:** Je, Kifaru ana urefu kiasi gani?	**A:** "mita 3.5–4.6	**Q:** Je, faru mweupe ana uzito gani?	**A:** kilogramu 3,500
	Q: How tall is a Rhino?	*A: 3.5–4.6 meters*	*Q: How much does a white rhino weigh?*	*A: 3,500 kilograms*
ru	**Q:** Когда был подписан Георгиевский трактат? **A:** 24 июля (4 августа) 1783 года		**Q:** В какой крепости был заключен Георгиевский трактат? **A:** Георгиевск (Северный Кавказ)	
	Q: When was the Treaty of Georgievsk signed?		*Q: In what fortress was the Treaty of St. George concluded?*	
	A: July 24 (August 4), 1783		*A: Georgievsk (North Caucasus)*	

Table 8: Examples of QA pairs from human-annotated TYDI QA and generated by QAMELEON (PT) on corresponding passages. English translations from Google Translate are added for readability.

Language	QAMELEON (PT)	
fi	**Q:** Milloin Tullintori valmistui?	**A:** *1990*
	Q: When was Tullintori completed?	*A: 1990*
ru	**Q:** Какой отраслью экономики преимущественно занято население Узбекистана? **A:** сфера обслуживания и туризма	
	Q: What sector of the economy is predominantly employed by the population of Uzbekistan? A: service and tourism	
sw	**Q:** Nyoka birisi iko katika familia gani?	**A:** Typhlopidae
	Q: Which family is the Birsi snake in?	*A: Typhlopidae*
ar	**Q:** في أي ولاية تقع بلدة فرانكلين ، مقاطعة مانيتووك؟	**A:** ويسكونسن
	Q: In which state is Franklin Township, Manitowoc County?	*A: Wisconsin*
id	**Q:** Siapa pencipta manga DN Angel?	**A:** Yukiru Sugisaki
	Q: Who created the manga DN Angel?	*A: Yukiru Sugisaki*

Table 9: QA pairs (random selection) generated by QAMELEON (PT) on Wikipedia passages. English translations from Google Translate are added for readability.

6 Related Work

Data Generation for QA Prior work on the generation of QA data has mostly focused on English and typically divides the task into answer extraction/generation and question generation, followed by some type of filtering. Alberti et al. (2019) employ round-trip consistency for filtering with BERT-based models. Other work (Shakeri et al., 2020) uses BART to jointly generate a question and its answer given an input passage, employing likelihood-based filtering. Lewis et al. (2021) use a RoBERTa-based passage selection model to identify interesting passages. Bartolo et al. (2021) additionally train the generation models on an adversarial QA dataset, while Yao et al. (2022) integrate a QA-pair ranking module.

The above approaches generally require large amounts of labeled QA data in the form of SQUAD (Rajpurkar et al., 2016) or Natural Questions (Kwiatkowski et al., 2019) to train passage selection and question generation models. In contrast, we only assume access to a few question-answer pairs per language.

Multilingual QA In this work we used mT5-XL (Xue et al., 2021) as our reference QA model. We note that a slightly more performant choice could have been ByT5 (Xue et al., 2022), which reports improvements on TYDIQA-GOLDP by operating directly on raw text instead of sentence pieces. Existing work on low resource multilingual QA has been relatively limited. Lee et al. (2018) propose to use automatically translated high-confidence QA examples for training, while other approaches (Kumar et al., 2019; Chi et al., 2020) only generate questions and require supervised training data in the target language. Other approaches (Riabi et al., 2021: Shakeri et al., 2021; Kramchaninova and Defauw, 2022) focus on zero-shot transfer, i.e., a multilingual model trained on QA data generation on SQUAD (and optionally automatically translated SQUAD data) is applied to other languages. Our work shows that few-shot settings result in better multilingual generation quality in comparison to zero-shot models.

Prompting Existing work (Brown et al., 2020; Schick and Schütze, 2021, *inter alia*) has shown

that prompting pre-trained large language models can lead to strong performance in a wide range of tasks including natural language generation and common sense reasoning. In the context of multilingual QA, Chowdhery et al. (2022) employ a single prompt and a few labeled examples in the target language. In contrast, we employ chain-of-thought prompting, and English answers and questions as a bridge. Moreover, our experiments with QAMELEON demonstrate that prompt tuning is superior and a viable alternative to large-scale annotation. Prompting in multilingual settings has achieved the best performance using English prompts and target language exemplars (Winata et al., 2021; Lin et al., 2022; Shi et al., 2022). We demonstrate that parameter-efficient methods such as prompt tuning using target language exemplars (Lester et al., 2021) is a superior choice.

7 Benefits and Limitations

The method proposed in this work, QAmeleon, prompt tunes large PLMs to generate multilingual synthetic question answering data. In this section we discuss its benefits over related approaches, but also drawbacks and limitations. The main benefits are large performance improvements over alternative methods, as borne out by our experiments, as well as surprising data efficiency achieved through large-scale pre-training and a few manual annotations. Alternative methods considered here are multilingual QA approaches for low resource languages, such as translate-test, translate-train, fine-tuning multilingual models directly on the small amount of available training data, performing multilingual QA directly through in-context learning, or even synthetic data generation with prompt engineered PLMs.

Another benefit of our approach stems from prompt tuning itself, which is able to learn from a tiny number of training examples, as low as one example per language in our experiments, whereas fine-tuning cannot be utilized as easily. Prompt tuning also affords the practical advantage of being space efficient; a fraction of a percent of the storage space is used to save the learned parameters, since only the learned soft prompt needs to be stored. Our evaluation methodology also provides benefits, since we measure question answer performance directly on downstream models instead of using a proxy like BLEU or ROUGE on generated questions. As shown in Table 5 proxy metrics can be misleading, and one might conclude that smaller models generate better questions than large PLMs if the evaluation were to consider only question BLEU scores.

The main drawback of our method is the high computational cost to prompt tune the PLM and to generate the synthetic data. While prompt tuning is not as expensive as fine-tuning, we still need to perform optimization on a model containing hundreds of billions of parameters. We estimate the cost of each prompt tuning and data generation experiment to be in the order of 256 TPU v3 chips for 12 hours. Another limitation of our experimental results is that they are fundamentally tied to a specific large PLM. PLMs are an area of active research, so any improvements in pre-training techniques, construction of pre-training sets, instruction tuning or reinforcement learning, are likely to translate in improvements for our synthetic data generation method. Promising areas of future work are parameter efficient techniques similar to prompt tuning, as well as analysis of data augmentations techniques like QAmeleon across different types and sizes of PLMs. Moreover, a more formal understanding of how the number of manual annotations (aka few shots) interacts with the quality of synthetic generation, would also be useful. Perhaps somewhat counter-intuitively, our experiments showed that QA performance does not drastically improve when scaling from 50 to 100 manual examples.

8 Data Release

To assist with the replicability of our results and to allow other researchers to benefit from our work, we will release a significant portion of the synthetic data generated by QAMELEON in the 5-shot scenario. To minimize the chance that question-answer pairs generated by the PLM contain sensitive, offensive or controversial material, we vetted each generated question with three human raters. We asked each rater to discard question-answer pairs that made generalized claims about groups, contained opinions that were potentially disparaging or embarrassing to one or more people, or names of individuals not related to media (e.g., film, TV) or sport. The release will contain 47,173 examples, each with a Wikipedia

passage, a question and an extractive answer, corresponding to 89% of the examples utilized in this work for the 5-shot scenario.

9 Conclusions

In this work, we examined the ability of pre-trained language models to generate synthetic data for bootstrapping multilingual QA systems, with as few as five examples in a new target language. We introduced QAMELEON, a parameter efficient approach which uses prompt tuning to automatically create multilingual QA data. Extensive experiments under different resource scenarios demonstrate that QAMELEON is superior to prompt engineering and competitive baselines based on machine translation. In the future, we would like to extend this approach to other multilingual tasks, including retrieval, summarization, and semantic parsing.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1620

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020a. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.618

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020b. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.421

Giuseppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.696

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage,

Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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. https://doi.org/10.1609/aaai.v34i05.6256

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, He-Yan Huang, and Furu Wei. 2022. Xlm-e: Cross-lingual language model pre-training via electra. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6170–6182. https://doi.org/10.18653/v1/2022.acl-long.427

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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 typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470. https://doi.org/10.1162/tacl_a_00317

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, Atlanta, Georgia. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a

single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.171

Moshe Koppel and Noam Ordan. 2011. Translationese and its dialects. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1318–1326, Portland, Oregon, USA. Association for Computational Linguistics.

Alina Kramchaninova and Arne Defauw. 2022. Synthetic data generation for multilingual domain-adaptable question answering systems. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 151–160, Ghent, Belgium. European Association for Machine Translation.

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, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1481

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466. https://doi.org/10.1162/tacl_a_00276

Kyungjae Lee, Kyoungho Yoon, Sunghyun Park, and Seung-won Hwang. 2018. Semi-supervised training data generation for multilingual question answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.243

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020b. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.653

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115. https://doi.org/10.1162/tacl_a_00415

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual language models. In *Proceedings of EMNLP 2022*.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell,

and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of ACL 2019*.

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. https://doi.org/10.3115/1073083.1073135

Colin Raffel, Noam Shazeer, Adam Roberts, Katherin Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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, Austin, Texas. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1264

Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2021. Synthetic data augmentation for zero-shot cross-lingual question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7016–7030, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.562

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.802

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.185

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892. https://doi.org/10.18653/v1/2020.acl-main.704

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, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.439

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Tom Sherborne and Mirella Lapata. 2022. Meta-learning a cross-lingual manifold for semantic parsing. https://doi.org/10.1162/tacl_a_00533

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. https://doi.org/10.48550/ARXIV.2210.03057

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Context-tuning: Learning

contextualized prompts for natural language generation. https://doi.org/10.48550/ARXIV.2201.08670

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-2623

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. https://doi.org/10.18653/v1/2022.emnlp-main.630

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.mrl-1.1

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306. https://doi.org/10.1162/tacl_a_00461

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. https://doi.org/10.18653/v1/2021.naacl-main.41

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. https://doi.org/10.18653/v1/2022.emnlp-main.296

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*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1259

Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Li, Mo Yu, and Ying Xu. 2022. It is AI's turn to ask humans a question: Question-answer pair generation for children's story books. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–744, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.54

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.672

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A closer look at few-shot crosslingual transfer: The choice of shots

matters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5751–5767, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.447