PAXQA: Generating Cross-lingual Question Answering Examples at Training Scale

Anonymous ACL submission

Abstract

Existing question answering (QA) systems owe much of their success to large, high-quality training data. Such annotation efforts are costly, and the difficulty compounds in the cross-lingual setting. Therefore, prior cross-lingual QA work has focused on releasing evaluation datasets, and then applying zero-shot methods as baselines. In this work, we propose a synthetic data generation method for cross-lingual QA which leverages indirect supervision from existing parallel corpora. Our method termed PAXQA (Projecting annotations for cross-lingual (x) QA) decomposes cross-lingual QA into two stages. In the first stage, we apply a question generation (QG) model to the English side. In the second stage, we apply annotation projection to translate both the questions and answers. To better translate questions, we propose a novel use of lexically-constrained machine translation, in which constrained entities are extracted from the parallel bitexts. We release cross-lingual QA datasets across 4 languages, totaling 662K QA examples. We then show that extractive QA models fine-tuned on these datasets outperform both zero-shot and prior synthetic data generation models, showing the sufficient quality of our generations. We find that the largest performance gains are for cross-lingual directions with non-English questions and English contexts. Ablation studies show that our dataset generation method is relatively robust to noise from automatic word alignments.

1 Introduction

A common framing of question answering (QA) in NLP is as a reading comprehension task, where questions about a specific text are to be answered by a span from a given context. Developing strong QA systems thus advances progress towards developing systems which can read and reason about texts. While earlier work developed QA models and resources in English only (Rajpurkar et al., 2016; Kwiatkowski et al., 2019), recent work has sought to extend beyond English. Such datasets include TYDI QA (11 languages; Clark et al. 2020), MLQA (6 languages; Lewis et al. 2020), XQuAD (10 languages; Artetxe et al. 2020), and MKQA (26 languages; Longpre et al. 2021), and are are annotated with the help of native speakers of diverse languages. However, the high annotation cost required means they are limited to evaluation, and there is no data available for training.

These works therefore use several zero-shot approaches as baselines on their datasets. First, zero-shot transfer, involves fine-tuning a multilingual pre-trained language model (LM) on English QA data, then applying this model directly to multilingual QA. Another, translate-train, uses a machine translation system (MT) to translate English data into other languages, then train new models on the translated data. Third, translate-test instead uses an English QA model, at inference time translating other language QA to English, then back for the final evaluation.

Alternatively, recent work has shown promising results with synthetic data augmentation (Riabi et al., 2021; Shakeri et al., 2021; Agrawal et al., 2022). In this approach, a question generation (QG) model is trained to generate synthetic multilingual QA examples, which are used as training data for a downstream QA model.

In this work, we propose a methodology for synthetic data augmentation, motivated by the insight that generating cross-lingual QA need not be entirely zero-shot. Instead, indirect supervision can be taken from widely-available parallel datasets originally collected for machine translation. Our method termed PAXQA (Projecting annotations for cross-lingual (x) QA) extracts QA examples from parallel corpora. PAXQA decomposes the task into two stages: English question and answer generation, then machine translation of questions and answers informed by word alignments. Through

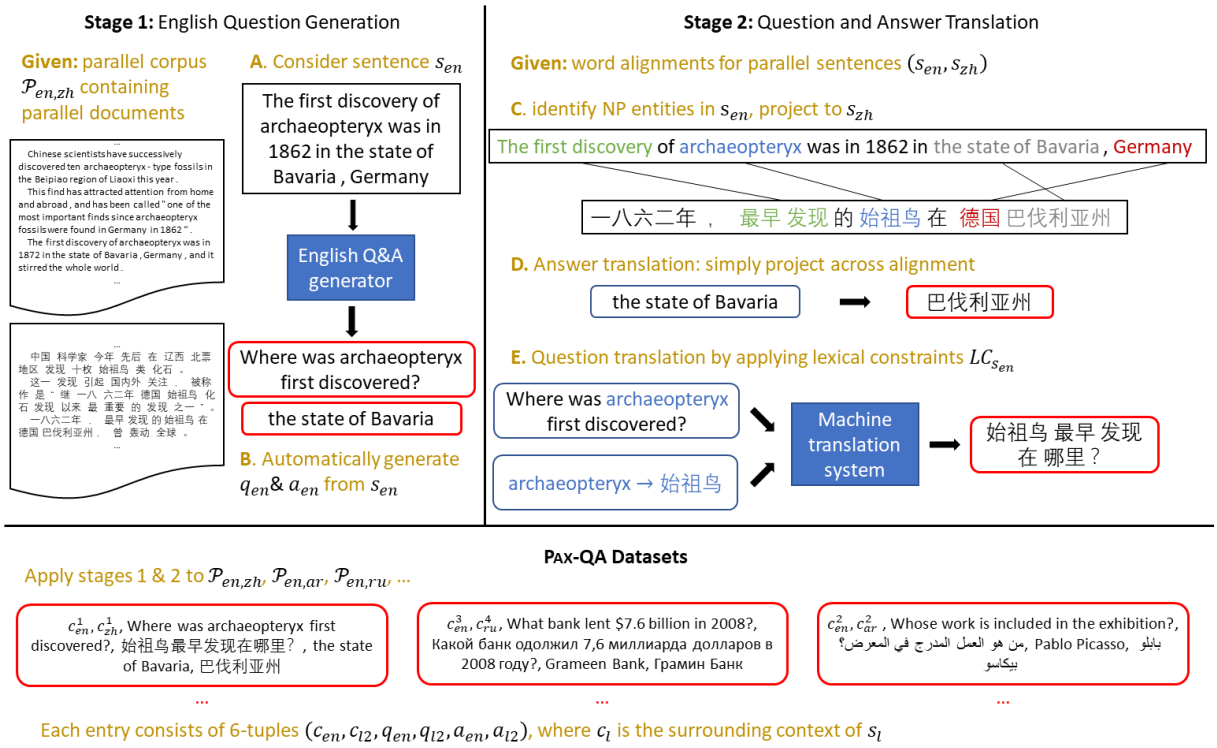

Figure 1: The PAXQA method generates a cross-lingual question-answering (QA) dataset given a word-aligned and parallel corpus. The two stages are English question generation (left), and question and answer translation (right). We run the pipeline on {ar-en}, {zh-en}, and {ru-en} datasets (bottom), resulting in a 661K cross-lingual QA examples – usable at training scale. Our *generation* pipeline proceeds similarly to prior works' *annotation* pipelines, but our method replaces all instances of human annotation with automated methods.

this decomposition, PAXQA serves as a framework in which the individual QG and MT systems can be updated with the latest developments.

We apply our methodology to generate large-scale cross-lingual QA datasets in 4 languages: Chinese, Arabic, Russian, and English. We validate the quality of our generations by showing both improvements for downstream QA tasks, and by performing a human evaluation task.

To facilitate follow-up work, we release our code and datasets.[1] Our four key contributions are:

- We introduce PAXQA, a method to generate cross-lingual question answering (QA) datasets at training scale. Our method, depicted in Figure 1, requires no new models to be trained, and instead decomposes cross-lingual QG into two automatic stages: 1) English question generation, and 2) word alignment-informed machine translation.
- To improve machine translation of questions, we propose a novel use of lexically-constrained machine translation. The lexical constraints are induced from the parallel sentences, and applied to the generated questions.
- We apply our method to generate cross-lingual QA datasets totaling 662K QA examples. Additionally, we ask human annotators to evaluate selected QA examples on several dimensions of quality; the resulting PAXQA validation and test sets total 1,724 QA examples.
- We use our generated datasets to fine-tune extractive cross-lingual QA models. Our models significantly outperform zero-shot baselines for the in-domain evaluations, and outperform prior data generation methods on benchmark datasets including MLQA. We perform ablations to show the robustness of PAXQA datasets created under various levels of noise.

2 Task Definition

In this work, we focus on *cross-lingual extractive question answering*. A cross-lingual extractive QA dataset consists of QA entries, each of which contains a context c_f, an answer a_f, and a question q_e (where f and e denote a source and a target language). The task is defined as follows: given q_e and

[1] link for review

c_f, a model must output an a_f which is extracted from c_f. Our goal is to both propose a method to synthetically generate such a dataset, and to train a model to solve the task.

Following prior works' broad definition of cross-lingual extractive QA (Lewis et al., 2020), it is possible that the two languages are the same ($e = f$). The only restriction is that in order to ensure the QA task is extractive, the context and answer must be in the same language; the question may or may not be in the same language.

In the literature, multilingual QA[2] most commonly refers the setting where QA is monolingual, but in multiple languages (Clark et al., 2020). While the dataset as a whole considers multiple languages $l_1^n = l_1, ...l_n.$, each entry on its own is monolingual – $(c, q, a)_{l_i}$, where $l_i \in l_1^n$. A cross-lingual QA dataset, in contrast, includes both monolingual and multilingual entries.

Notation We denote a language pair as {f-e}. A language pair covers 4 cross-lingual directions: f, e ; e, f ; f, f ; e, e. The fields are the languages of the context and question respectively. For example, f, e means a context in f, and a question in e.

3 Related Work

3.1 Cross-lingual QA Resources

Our work draws on two cross-lingual QA datasets to perform evaluation: MLQA (Lewis et al., 2020) and XORQA (Asai et al., 2021). **MLQA** is an extractive QA dataset. It is multi-way parallel across 7 languages, and consists of 46k examples in total. XORQA is an open-domain QA dataset. It covers 8 languages, which are each parallel to English, and consists of 40k examples. It also includes an extractive QA version, $\textbf{XORQA}_{\textbf{GoldP}}$.

3.2 Cross-lingual Question Generation

Cross-lingual question-only generation was explored by Kumar et al. (2019) and Chi et al. (2020). As answers are not provided, evaluation on QA tasks cannot be performed. These studies evaluate generated question quality both by human evaluations and by automated evaluations such as BLEU.

Concurrent to our work, QAMELEON (Agrawal et al., 2022) generates synthetic multilingual QA datasets using prompt-tuning of a pretrained large language model. They find that these models can generate good quality QA using only five QA examples per language. While they only evaluate on multilingual QA datasets, it is likely that their method can easily be adapted to cross-lingual QA.

Riabi et al. (2021) and Shakeri et al. (2021) are most related to our work, as they also consider cross-lingual question and answer generation (QA generation, or simply QG). Both approaches adopt the view that as QG is the dual of QA, one can flip existing QA datasets into QG datasets. A supervised model can then be trained to generate cross-lingual QA examples, which are used as synthetic training data for a downstream QA model. Riabi et al. (2021) train their QG model on both English and machine-translated SQuAD data, and then train a separate QA model. Shakeri et al. (2021) train a single model to perform both QA and QG. Their multi-task setup consists of a QG task using SQuAD examples, and a masked language modeling task on TYDI QA (without contexts).

The primary difference of our work versus the approaches of Riabi et al. (2021) and Shakeri et al. (2021) is that while those approaches train new models for cross-lingual QA generation, our work requires no additional models to be trained, instead introducing a methodology. We simply combine existing English QG systems and MT systems. This makes our approach simpler, and allows us to replace each component with state-of-the-art models as they are developed. Furthermore, a weakness of these prior approaches is that to train their models, they require supervised data for non-English QA. This means they cannot be extended to low-resource languages, where such resources do not exist. Still, parallel datasets between low-resource languages and English often exist, allowing our method to be applied.

3.3 Annotation Projection

Annotation projection is a time-tested technique which serves to transfer annotations from text in one language to parallel text in another language (Li, 2022). It relies on word alignments, which can be learned in an unsupervised manner. In low-resource scenarios, annotation projection has been shown to be successful in cross-lingual NLP tasks, from parsing (Hwa et al., 2005; Rasooli and Collins, 2017) to semantic role labeling (Aminian et al., 2019). In this work, we apply annotation projection as a way to translate spans for QA: either the answer spans or the entities found in questions.

[2]For here on, we use 'QA' to mean 'extractive QA'.

4 Data Generation Method

We now describe the PAXQA data generation methodology which was outlined in Figure 1. The goal is to generate synthetic cross-lingual QA datasets $\hat{\mathcal{D}}_{en,l}$, for each language of interest l ($\hat{}$ specifies that it is generated). Our method assumes the availability of a parallel corpus $\mathcal{P}_{en,l}$ between English (en) and each l. We denote the two sides of a parallel corpus as $\mathcal{P}_{en}$ and $\mathcal{P}_l$. We also assume that these parallel corpora are word-aligned, either by humans or by an word alignment tool. We now describe the two stages of our methodology.

4.1 Question and Answer Generation

In the first stage, we generate Q&A pairs from $\mathcal{P}_{en}$. For this purpose, we adopt the question generation (QG) model of Dugan et al. (2022). This model is fine-tuned for the three tasks of answer extraction, question generation and question answering, with the intuition that this improves individual task performance. We apply QG directly to each sentence s in $\mathcal{P}_{en}$. We then manually inspect some of the generated English Q&A pairs, and implement heuristic filters to remove low-quality generations. For example, we filter answers containing question marks; other are listed in Appendix B.

Using Paragraph Contexts Given that each question can be answered from the sentence s it was generated from, we make the task harder by setting the context c to the entire paragraph where s appears. This modification also aligns PAXQA entries to SQuAD-style paragraph contexts. We therefore choose parallel corpora which have paragraph annotations. Note that using paragraph contexts is an optional post-processing step, and not a necessary part of the PAXQA method. [3]

4.2 Question and Answer Translation

In the second stage, we extend $\hat{\mathcal{D}}_{en}$ to the cross-lingual setting by translating both the question and the answer. We propose to use a translation process that is informed by word alignments, and which differs for questions and answers.

To illustrate the process, let us consider a single generation from the prior stage, q_{en} and a_{en}. These are generated from s_{en}, which consists of tokens s^i_{en}; we are also given that s_{en} is parallel to s_l. A word alignment wa links tokens from s_{en} and s_l which are translations of each other.

4.2.1 Answer Translation

As a_{en} is extracted from s_{en}, it corresponds to a set of tokens $s^u_{en}, s^w_{en}, ...$ with indices $u, w, ...$ By applying the word alignment wa on these indices, we translate to l and obtain $a_l = s^x_l, s^y_l, ...$ Thus, answer translations come for "free" with the word alignments – no MT system is needed.

Of course, the quality of this translation process depends on the quality of the word alignment. Word alignment errors will propagate to the projected answers.[4] Furthermore, wa may be underspecified, in which case not all (or none) of the words in a_{en} can be projected. We discard those Q&A pairs where either a_{en} or a_l are blank.

4.2.2 Question Translation

The translation of q_{en} requires a machine translation (MT) system. In our work, we propose to use a lexically constrained MT system to that enforce lexical constraints within the text of questions. The rationale is that, since the task is already extractive with respect to answers, adding lexical constraints makes the questions more extractive as well. Furthermore, this can ensure the proper translation of more difficult terms, such as named entities, which are likely to be emphasized when evaluating reading comprehension. We perform question translation using three methods:

- For standard NMT, we use Transformer models (Vaswani et al., 2017) (**vanilla**). We train a model for each language pair with data from WMT (see Appendix A).
- We use the publicly available API for Google Translate (Wu et al., 2016) (**GT**). This is a strong translation system; however, it is not reproducible given its regular API updates. Also, the underlying mechanisms or training data are not specified.
- We also use a lexically constrained NMT system (**lex cons**), described below.

Lexically Constrained MT Lexically constrained MT adds to the model input a set of constraints LC which specify how specific source phrases should be translated into target phrases. In our work, we utilize the template-based MT

[3] As a looser restriction, we can instead choose parallel corpora with article-level annotations, in which case a 'paragraph' is simply $s + N$ surrounding sentences.

[4] Automated word aligners generally favor recall over precision. Therefore, alignment error are more likely to results in missing links, rather than incorrect ones.

model of Wang et al. (2022). The architecture is identical to a vanilla Transformer, but the method modifies the data format in order to incorporate the constraints as a template.

In prior work, constraints were sampled by choosing source phrases which were arbitrary sequences from s_l (Chen et al., 2021; Post and Vilar, 2018). Our setting differs in that we apply these lexical constraints to the generated abstractive question q_{en}, instead of the context s_{en} itself. Therefore, randomly sampling likely results in spans that do not appear in q_{en}. Using the insight that noun phrases (NPs) are more likely to be kept in q_{en}, we propose the key modification of sampling constraints by extracting NPs from s_{en}. We then keep only those constraints LC which appear in q_{en}.

An example is shown in Figure 1, Stage 2. Of the three NPs in s_{en}, only 'archaeopteryx' exists in q_{en}. So $LC = \{\text{archaeopteryx} \rightarrow \text{始祖鸟}\}$. By contrast, random sampling from might result a span '1862 in the' which does not appear in q_{en}.

We apply this NP constraint extraction process at inference-time, and otherwise retrain and reproduce the MT model of Wang et al. (2022) for our target languages. Compared to random sampling, our process allows for an average of three times more lexical constraints to be used per question.

5 Experimental Setup

Although the PAXQA method can be applied to any language l with English-l parallel data, in our experiments we address three languages: Chinese (zh), Arabic (ar), and Russian (ru). We hope that the diversity in scripts and language families will illustrate the wide applicability of our method.

5.1 Datasets Used

Our proposed cross-lingual QA generation method requires both parallel corpora and word alignments. We provide further details in Appendix A.

Parallel Corpora The machine translation community has made many parallel datasets publicly available.[5] We use the News-Commentary (NC) and GlobalVoices (GV) datasets which are multi-way parallel between many languages. We consider subsets which include English and our target languages. We also use the Arabic and Chinese GALE datasets from the LDC.[6]

[5]Many corpora are at https://opus.nlpl.eu
[6]https://catalog.ldc.upenn.edu

	PAXQA$_{\text{AWA}}$			PAXQA$_{\text{HWA}}$		
Lang	Train	Dev	Test	Train	Dev	Test
zh	72895	125	92	30963	104	190
ar	134912	193	87	51084	132	181
ru	370534	300	320	–	–	–
Total	578341	618	499	82047	236	371

Table 1: Number of generated cross-lingual QA entries per language l and per split. Entries are cross-lingual between l and English. PAXQA$_{\text{HWA}}$ is the dataset generated from the human word alignments, while PAXQA$_{\text{AWA}}$ is the dataset from automatic alignments.

Word Alignments The GALE datasets include word alignments annotated by humans. For NC and GV datasets, we obtain alignments with the `awesome-align` (Dou and Neubig, 2021) package. `awesome-align` induces alignments from a given multilingual LM. For zh, we use the provided fine-tuned checkpoint; for ru and ar, we use mBERT (Devlin et al., 2019).

Parallel dataset statistics, as well as the number of Q&A pairs generated from each dataset, are given in Appendix Table 5.

5.2 PAXQA Datasets

We run our data generation method on each of the parallel corpora to obtain cross-lingual QA entries.

We designate the dataset created by concatenating QA entries generated from human word aligned datasets as PAXQA$_{\text{HWA}}$; likewise, from automatic word aligned datasets as PAXQA$_{\text{AWA}}$. We further split into train/development/test sets; dev and test sets are created through human annotation filtering (further described in Section 7.1). As shown in Table 1, there are 580K cross-lingual QA entries for PAXQA$_{\text{AWA}}$, and 82K for PAXQA$_{\text{HWA}}$.

Each PAXQA entry is a 6-tuple $(c_f, c_e, q_f, q_e, a_f, a_e)$. Recall that a language pair {f-e} consists of 4 cross-lingual QA directions. So training a model on PAXQA$_{\text{HWA}}$, for example, uses $82\text{K} * 4 = 328\text{K}$ training instances.

6 Results

The PAXQA generation method creates synthetic QA data, we can then be used to fine-tune a cross-lingual extractive QA model. QA performance that beats prior work is our first way of validating the quality of the generations. We report results on three datasets: PAXQA$_{\text{HWA}}$ test, MLQA (Lewis et al., 2020), and XORQA$_{\text{GoldP}}$ (Asai et al., 2021).

The evaluation metric is the mean token F1 score, calculated with the official MLQA script.[7]

Prior work reports average results across all cross-lingual directions.[8] We instead group directions as follows: 'non-en q'uestion + en context; 'en q'uestion + non-en context; 'monolingual', and 'non-en (xling)' cross-lingual. We then analyze the results by group, which gives us a clearer picture of how the methods affect different directions.

QA Model We adopt XLM-R (Conneau et al., 2020) as our QA model, and initialize to the pretrained large checkpoint from the `transformers` library (Wolf et al., 2019). Following the advice of (Alberti et al., 2019), we fine-tune in two rounds: first on synthetic QA data, then on SquAD. We find that this improves results over shuffling real and synthetic data.

QA Data We use the PAXQA$_{\text{HWA}}$ and PAXQA$_{\text{AWA}}$ data splits. Recall that each PAXQA entry gives us 4 cross-lingual entries (f, e; e, f; f, f; e, e). For training, we also use SQuAD; for validation, we also use MLQA and XORQA$_{\text{GoldP}}$.

6.1 QA Models for Comparison

For transfer learning approaches, we consider only the **zero-shot** baseline, since the translate-test and translate-train baselines require significant computational overhead. Prior work compared these three baselines (Lewis et al., 2020; Longpre et al., 2021) and found that zero-shot and translate-train perform similarly, while translate-test performs poorly because it uses needs to translate inference data twice: the input in l to English, then the prediction in English to l.

We compare PAXQA-trained models to the zero-shot baseline results reported by MLQA. We also compare to Riabi et al. (2021), whose results are directly comparable to ours because the same underlying XLM-R model is used.

6.2 In-domain Results

Table 2 shows the results on the PAXQA$_{\text{HWA}}$ test sets. All 4 models are initialized to XLM-R, then further fine-tuned on SQuAD or on PAXQA$_{\text{HWA}}$. The scores of the three PAXQA$_{\text{HWA}}$-trained models are fairly close. This is likely because they use the same context and answers, and differ only in how English questions are translated. Still, we see that lex cons beats vanilla overall, and performs about the same as GT. This is notable because GT is a much stronger MT system than our bilingual Transformer-based models.

	non-en q		**en** q		**monolingual**			
q MT	en,zh	en,ar	ar,en	zh,en	ar,ar	zh,zh	en,en	avg
SQuAD	67.0	78.9	85.8	83.5	79.8	73.9	90.9	80.0
vanilla	**91.8**	90.6	87.7	88.4	87.8	82.2	92.7	88.7
lex cons	88.9	91.5	**90.4**	86.8	**90.2**	84.1	**93.9**	89.4
GT	88.0	**93.4**	89.7	**89.0**	88.3	**85.0**	93.3	**89.5**

Table 2: PAXQA$_{\text{HWA}}$ test F1 scores for XLM-R models trained on various datasets. Rows 2-4 are trained on only PAXQA$_{\text{HWA}}$, differing in the question translation method. 'e, f' indicates the context and answer are in e, while the question is in f.

The largest improvements with training on PAXQA data are for 'non-en q'. The lex cons model achieves +21.9 F1 for en,zh (88.9 > 67.0), and +12.6 F1 for en,ar (91.5 > 78.9). We also see significant improvements for 'monolingual' (+10 F1), and modest ones for 'en q' (+3-5 F1).

6.3 Generalization Results

We now evaluate how well PAXQA$_{\text{HWA}}$-trained models generalize to MLQA. These results can be considered out-of-domain, as MLQA was collected over Wikipedia, while PAXQA was generated from news articles. Results are shown in Table 3.

Our best overall model is lex cons + SQuAD (row 6), though the other PAXQA variants perform similarly (rows 5-7). Compared to the zero-shot baseline (row 2), our best model improves most significantly for 'non-en q': +16.4 F1 for en,zh (78.9 > 62.5); +11.9 F1 for en,ar (76.7 > 64.8). 'non-en xling' achieve +10 F1, while 'en q' and monolingual performance are similar to the baseline. Our method outperforms the prior best synthetic generation method (row 3) by 1.4 F1 overall.

We also observe that as expected, a model trained on PAXQA$_{\text{HWA}}$ alone (row 4) underperforms those with both PAXQA$_{\text{HWA}}$ and SQuAD; the same goes for PAXQA$_{\text{AWA}}$ (row 8 vs. row 9).

6.4 Results using Automatic Word Alignments

In this section, we report PAXQA$_{\text{AWA}}$ results.[9] In this setting, word alignments are noisy and include many alignment errors.

[7] https://github.com/facebookresearch/MLQA/blob/main/mlqa_evaluation_v1.py

[8] We report these averaged results in Appendix Table 8 Our best model achieves 71.1 F1 vs. 70.5 from prior work.

[9] We also trained on a concatenation of PAXQA$_{\text{HWA}}$ and PAXQA$_{\text{AWA}}$ data, but results were about the same.

	#	Method	Train Data	non-en q en,zh	non-en q en,ar	en q zh,en	en q ar,en	monolingual ar,ar	monolingual zh,zh	monolingual en,en	non-en xling ar,zh	non-en xling zh,ar	avg
baseline	1	zero-shot (Lewis et al., 2020)	SQuAD (S)	53.9	50.8	52.9	60.0	54.8	61.1	74.9	43.5	40.9	54.8
	2	zero-shot	SQuAD (S)	62.5	64.8	68.8	67.1	65.6	70.1	84.3	53.3	54.8	65.7
	3	Riabi et al. (2021)	synthetic + S	77.8	76.3	66.8	68.9	66.6	67.7	83.9	**64.9**	61.7	70.5
human WA	4	PAXQA$_{HWA}$	lex cons	72.6	70.2	64.6	61.1	60.1	65.1	77.9	57.2	58.3	65.2
	5	PAXQA$_{HWA}$	vanilla + S	77.9	75.9	69.8	68.8	**67.0**	70.7	84.6	64.1	63.7	71.4
	6	PAXQA$_{HWA}$	lex cons + S	**78.9**	**76.7**	**70.2**	**69.2**	**67.0**	**71.1**	**84.8**	64.3	**64.9**	**71.9**
	7	PAXQA$_{HWA}$	GT + S	77.7	75.0	69.3	68.3	66.4	70.2	84.1	63.2	63.2	70.8
auto WA	8	PAXQA$_{AWA}$	lex cons	69.5	67.8	61.7	60.5	59.6	63.1	77.0	55.8	56.0	63.4
	9	PAXQA$_{AWA}$	lex cons + S	77.3	74.6	69.8	69.0	67.0	70.3	84.5	63.4	62.5	70.9

Table 3: MLQA test F1 scores for models trained on various datasets. The model in row 1 is XLM, and in the other rows is XLM-R. The PAXQA rows are obtained by training on generated cross-lingual QA pairs from parallel datasets which are either human word-aligned (rows 3-7) or automatically word-aligned (rows 8-9). To translate questions from English, the systems are vanilla NMT, lexically constrained NMT, or Google Translate.

Method	Train Data	en,ar	en,ru
zero-shot	SQuAD (S)	57.4	71.2
PAXQA$_{HWA}$	lex cons	58.6	68.1
PAXQA$_{HWA}$	lex cons + S	**68.9**	71.1
PAXQA$_{AWA}$	lex cons + S	64.7	**72.3**

Table 4: XORQA$_{GoldP}$ test F1 scores for models trained on various datasets. All rows are based on a fine-tuned XLM-R model.

Results on MLQA are shown in Table 3. Comparing the best PAXQA$_{HWA}$ model (row 6) and the best PAXQA$_{AWA}$ model (row 9), we see performance only drops by 1.0 F1 across overall. Still, 'non-en q' and 'non-en xling' results for PAXQA$_{AWA}$ handily beat the baseline model (row 2), and monolingual F1 scores are similar.

We also report results for XORQA$_{GoldP}$ dev sets for en,ar and en,ru[10], as shown in Table 4. For en,ar PAXQA$_{HWA}$ achieves +11.5 F1 (68.9 > 57.4). For en,ru, PAXQA$_{AWA}$ achieves +1.1 (72.3 > 71.2); PAXQA$_{HWA}$ has not seen any en,ru synthetic data, so it performs the same as the baseline.

Discussion We find that the PAXQA$_{HWA}$-trained models perform the best, notably achieving a new state-of-the-art on MLQA over Riabi et al. (2021). The significant but relatively small improvements (1-2 F1) in downstream QA performance are expected, as the underlying approaches (training on synthetic data, then real data) and models (fine-tune from XLM-R) are very similar. Still, they likely reflect larger improvements in the primary QG task. As PAXQA$_{AWA}$-trained models perform only 1.0 F1 lower than PAXQA$_{HWA}$, our method is relatively robust to noise from automated alignments.[11] Recalling that our method does not require non-English QA data, these characteristics show that PAXQA is effective and extensible beyond the 4 languages considered here.

7 Analysis

7.1 Evaluating the Quality of Generations

We run a human annotation task to evaluate the quality of generations. At a high-level, we adapt the methodology of Dugan et al. (2022). We sample 2,921 QA entries from the PAXQA generations for the evaluation task. Of these, 1,724 (59.0%) were acceptable to human annotators. We then randomly split these entries into development and test sets. More details are provided in Appendix F.

7.2 Ablations

Prior work (Shakeri et al., 2021; Riabi et al., 2021) showed that a synthetic data training scheme allows for cross-lingual generalization. This means that even training with a single language pair improves cross-lingual QA performance for all directions, including unseen ones. From this, they hypothesize that multilingual models such as XLM-R already possess good multilingual internal representations, and this scheme allows for generalization to non-English QA. We verify this hypothesis by performing several ablations on our datasets.

Bilingual QA Models Instead of training a single model to perform cross-lingual QA for all pairs,

[10] XORQA does not release the test set answers, and ru,en and ar,en are not supported.

[11] For example, alignment error rate of {zh-en} is 13.4.

c_{en}: The first discovery of archaeopteryx was in 1862 in the state of Bavaria, Germany.
q_{en}: Where was archaeopteryx first discovered?
a_{en}: the state of Bavaria, a_{zh}: 巴伐利亚
q_{zh} **(vanilla)**: 最早的考古发现在哪里?
(en: Where was the earliest archaeological discovery?)
q_{zh} **(lex cons)**: 始祖鸟最早发现在哪里?
(en: Where was archaeopteryx first discovered?)
q_{zh} **(GT)**: 最早发现的最早的考古学是哪里?
(en: Where was the earliest discovery of the earliest archaeology?)

Figure 2: An example entry from PAXQA$_{\text{HWA}}$.

we can train bilingual models. We do so for the zh,en and ar,en pairs of PAXQA$_{\text{HWA}}$. Results are shown in Appendix Table 7. We see that the multilingual model performs quite similarly to the bilingual models. In fact, the model trained on only ar,en somewhat outperforms the zh,en for several zh directions (i.e. zero-shot), and vice versa for the model trained on only zh,en.

Extending to non-English Parallel Directions In our prior experiments, we generated English-centric QA entries. However, we can extend to non-English parallel directions by pivoting multi-way parallel dataset through English. We apply the following pivoting strategy to the News-Commentary parallel corpora. We first take the articles which are parallel between all 3 languages zh, ar, and en. The Q&A generation stage of PAXQA remains the same, since it operates on English. In the Q&A translation stage, we perform our alignment-informed translations from en to ar, from en to zh. We now have question, answers, and contexts in both languages, giving us an {ar-zh} cross-lingual extractive QA dataset.

Results for such a fine-tuned QA model (ar-zh pivot data + SQuAD + lex cons) are shown in Table 7. F1 scores are overall slightly lower than prior models. This is even the case for the ar,zh and zh,ar directions. A possible reason is because the pivoting strategy compounds errors from the noisy machine methods, since we apply automatic alignments twice, and perform MT twice. Still, this experiment provides more evidence for the cross-lingual generalization ability of LMs.

7.3 Case study: Comparing Question Translation Methods

In Section 6, showed that models using 3 different methods of question translation performed similarly. As a case study, consider the example shown in Figure 2. Because "archaeopteryx" is an uncommon word, the vanilla and GT systems fail to translate it properly; it is instead translated incorrectly to 考古 (archaeological) and 考古学 (archaeology). The lexically constrained system gets it correct, because it has the constraint given as input.

Despite the incorrect question translations, it is easy to see how any NMT system could derive the correct answer by simply noticing that the question asks for "where" (in either language), and could just return the location "the state of Bavaria". This is a known issue with reading comprehension-style questions (Kwiatkowski et al., 2019).

A Harder Task The case study and the relatively high QA results suggest that even a cross-lingual formulation of the extractive QA task is fairly easy. We identify *round-trip cross-lingual QA* as the immediate next step. For this task, given (c_f, q_e), a model must predict a_e.[12] While the answer can still be found in the context, it must now be translated back to the question's language (i.e., round-trip). This would be more useful to end-users who would like to be able to ask questions of multilingual documents, and receive answers they can understand. The PAXQA$_{\text{HWA}}$ and PAXQA$_{\text{AWA}}$ datasets can indeed be used for this new task. However, the modeling approaches covered here do not support it, and we leave such efforts to future work.

8 Conclusion

We presented PAXQA, a synthetic data generation method for cross-lingual QA which leverages indirect supervision from parallel datasets. We decompose the task into two stages: English QA generation, then QA translation informed by annotation projection. Unlike prior methods, PAXQA requires no training of new models, nor any non-English QA data to use for supervision. This means our method can even be applied to low-resource languages. We proposed the novel use of lexically-constrained MT to better translate questions, which assists in proper translation of uncommon entities. Finally, we showed that training on PAXQA data allows downstream models to significantly outperform zero-shot baselines, and achieve a new state-of-the-art on the MLQA benchmark. In order to facilitate future research in the field, we released our code and datasets.

[12] Asai et al. (2021) explores the related round-trip task for open-domain cross-lingual QA.

9 Limitations

The main limitation of our method is that it requires datasets which are parallel to English. However, because of the great efforts placed into collecting resources for machine translation, such datasets are relatively available. In the MT field, "low-resource" generally means 1M parallel sentences. This is ample enough data to train automatic word aligners between English and some language, needed by our method.

Because of resource constraints on our end, we only ran our method end-to-end for three languages. However, weave claimed that by decomposing cross-lingual QG into English QG and MT steps, our method allows for QA generation in low-resource languages. As an initial step, we are running the PAXQA pipeline on the FLoRes v1 (Guzmán et al., 2019) dataset, which covers Nepali and Sinhala. After the dataset is generated, we will investigate how we can evaluate the quality of generations for these languages not been studied by the QA community yet. While back-translation using NMT could be a first start, more likely this requires finding native human annotators.

Beyond the parallel dataset limitation, we acknowledge that the English-centric nature of our approach is not ideal. We inherit this problem from the general body of cross-lingual QA research. For example, almost all datasets collected require English fluent annotators – either to translate questions from English to their native language, or even to be able to read instructions written in English. Still, we highlight the need for future research to be fair to all languages. Our ultimate goal, as we discussed in Section 7.3, is to develop QA models that allow users to pose questions regarding documents in any language, and receive an answer back in their native language. Given that the bulk of the information available on the web is in English, such a system would allow for more equitable access to the world's information resources for all humans.

Another limitation is that our human evaluation task focused on the English side of cross-lingual QA entries. This is because our annotators were students in an American university, and therefore we did not expect them to be multilingual. We checked the quality of the translated answer through back-translation, but this is only a proxy. Furthermore, the non-English question remains unverified.

Ethical Considerations Before beginning to do annotations, the human annotators we recruited were given a set of instructions, and had the choice to participate or not, and to cease participation at any time. We believe that the extra credit for their final course grade was a fair incentive.

The synthetic data generation method we used can possibly generate misleading or even toxic information, depending on the contexts it is given. Some of our human annotators flagged certain generations for our review. The culprit was contexts which expressed someone's opinion; for example, an interview with a controversial politician. In such cases, the generated questions were from the perspective of that person. From the 3K annotations we did, we discarded any QA entries that were deemed unacceptable. However, we do not verify all 600K+ examples we release. We do apply some filtering steps to attempt to mitigate low-quality generations. Furthermore, we have only run our QA generation method on news datasets which are widely used and understood within the general community.

References

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2022. Qameleon: Multilingual qa with only 5 examples. *arXiv preprint arXiv:2211.08264*.

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.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2019. Cross-lingual transfer of semantic roles: From raw text to semantic roles. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 200–210, Gothenburg, Sweden. Association for Computational Linguistics.

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.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 547–564, Online. Association for Computational Linguistics.

Guanhua Chen, Yun Chen, and Victor O. K. Li. 2021. Lexically Constrained Neural Machine Translation with Explicit Alignment Guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12630–12638.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020. Cross-Lingual Natural Language Generation via Pre-Training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7570–7577.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, DaHyeon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. A feasibility study of answer-agnostic question generation for education. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1919–1926, Dublin, Ireland. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325.

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.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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.

Bryan Li. 2022. Word alignment in the era of deep learning: A tutorial. *arXiv preprint arXiv:2212.00138*.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324.

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.

Mohammad Sadegh Rasooli and Michael Collins. 2017. Cross-lingual syntactic transfer with limited resources. *Transactions of the Association for Computational Linguistics*, 5:279–293.

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.

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.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shuo Wang, Peng Li, Zhixing Tan, Zhaopeng Tu, Maosong Sun, and Yang Liu. 2022. A template-based method for constrained neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3665–3679. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

A Details on Datasets Used

The parallel corpora we use in this work come from three datasets: GALE, NewsCommentary, and GlobalVoices. The latter two come from OPUS (Tiedemann, 2012). Dataset statistics are given in Table 5.

GALE[13] are a collection of parallel news datasets available on LDC. These are word-aligned by trained human annotators. For Arabic-English, we use the following LDC numbers: LDC2013T10, LDC2013T14, LDC2014T03, LDC2014T05, LDC2014T08, LDC2014T10, LDC2014T14, LDC2014T19. For Chinese-English, we use: LDC2012T16, LDC2012T20, LDC2012T24, LDC2013T05, LDC2013T23, LDC2015T04, LDC2015T06.

GlobalVoices[14] is a parallel corpus of news articles in 46 languages. We use only the ar-en and ru-en subsets of the data. While standard (zhs) and traditional Chinese (zht) are part of this corpus, we do not use them because we have found that the zh-en sentence alignments are of very poor quality. As the sentence alignments for other GlobalVoices directions are near perfect, we suspect some preprocessing issue occured.

News-Commentary[15] is a parallel corpus of news commentaries in 15 languages. We use only the ar-en, zh-en, and ru-en subsets of the data. Note that both GlobalVoices and News-Commentary parallel between almost all languages considered. Our work only generates questions (originally) in English, which requires the parallel with English restriction. We leave future work to generate questions directly from multiple languages.

B Filtering Lower-Quality Generations

We implement heuristic filters to remove any generations that have the following properties:

1. The generation is a duplicate.
2. The question is of the form "What is the answer ...".
3. The answer contains a question mark.
4. The source sentence is less than 5 tokens, not including punctuation.
5. The answer (either a_{en} or a_l) consists of only punctuation.

[13] https://catalog.ldc.upenn.edu/LDC_NUMBER

[14] https://opus.nlpl.eu/GlobalVoices.php

[15] https://opus.nlpl.eu/News-Commentary.php

lang	dataset	# sents	# QA gen
zh	NewsComm	73623	73112
	GALE	31390	31257
ar	NewsComm	80119	80863
	GlobalVoices	58985	54329
	GALE	49568	51397
ru	NewsComm	203598	208190
	GlobalVoices	164895	162964

Table 5: Statistics for parallel corpora used in this work. Al corpora are parallel between English and the specified 'lang'. '# QA gen' is the number of question-answer pairs generated from each dataset using PAXQA. The bolded GALE dataset has human word alignments, while the others use automated alignments.

We note that most of these issues can be addressed by a higher quality question generation system. We leave this to future work, and note that PAXQAmethod is an orthogonal contribution to those developments.

C Modeling Details

We will release all code, dataset, and documentation in the final version of this paper (which will also include hyperparameters and other settings). For now, we provide links to the packages we used.

Our cross-lingual QA generation method decomposes the task to QG and then MT. We use the question and answer generation system of Dugan et al. (2022) [16] – without any additional fine-tuning. We use the lexically constrained MT system of Wang et al. (2022) [17]. To obtain word alignments, we use `awesome-align` [18].

Our QA models are developed on top of the `transformers` library. We modify the provided QA training scripts [19] for our specific needs.

D Additional Results

In-Domain Results Results for the PAXQA$_{\text{HWA}}$ test set for additional configurations are given in Table 6.

Generalization Results Results for the MLQA test set for additional configurations are given in Table 7. Table 8 reports the averaged F1 and EM scores across all MLQA directions. For our best model configuration (PAXQA$_{\text{HWA}}$ lex cons + S), Table 9 gives the individual EM scores, and Table 10 gives the individual F1 scores.

E Examples

Tables 11 and 12 show sample PAXQA$_{\text{HWA}}$ entries for Chinese and Arabic respectively. Recall the the contexts are drawn from the news domain. For the Chinese sample entry, Table 13 compares the question translation of the 3 different MT systems.

F QA Generation Evaluation Task

We sample 2921 QA entries from all QA generations for annotation. These QA entries are generated from randomly sampled articles. The human annotators are drawn from students enrolled in a graduate-level Artificial Intelligence course, at an American university. The 129 participants in total were rewarded with extra credit.Annotators are presented the context, the question, and the answer (all in English), and asked the following 3 yes/no questions:

(i). Does the question make sense outside of the immediate context?

(ii). Is the question relevant and/or interesting?

(iii). Is the answer to the question correct?

Because we did not specifically search for bilingual annotators, we evaluated only in English. As a proxy to evaluate answer translations, we propose to back-translate the aligned answers into English. We present this to annotators as an "Alternate Answer", and additionally ask:

(iv). Do "Answer" and "Alternate Answer" mean the same thing?

The annotation interface is shown in Figure 3. We collect 3 annotations per task, and assign the majority label as the gold label.

We evaluate inter-rater reliability using averaged pair-wise Cohen's kappa κ: (i) 0.18, (ii) 0.18, (iii) 0.41, (iv) 0.51. The κ scores for (i) and (ii) are especially low, which indicates that workers had very subjective understandings of interpretability

[16] https://huggingface.co/valhalla/t5-base-qa-qg-hl

[17] https://github.com/THUNLP-MT/Template-NMT

[18] https://github.com/neulab/awesome-align

[19] https://github.com/huggingface/transformers/blob/main/examples/pytorch/question-answering/run_qa.py

Method	Train Data	Train Lang	non-en q en,zh	non-en q en,ar	en q ar,en	en q zh,en	monolingual ar,ar	monolingual zh,zh	monolingual en,en	avg
PAXQA$_{\text{HWA}}$	lex cons	ar,zh	88.9	91.5	**90.4**	86.8	**90.2**	84.1	**93.9**	89.4
PAXQA$_{\text{HWA}}$	lex cons	zh	88.2	90.4	88.0	86.7	86.4	84.0	91.9	87.9
PAXQA$_{\text{AWA}}$	lex cons	ar,zh,ru	84.2	89.6	84.1	79.2	81.1	71.7	90.2	82.9

Table 6: PAXQA$_{\text{HWA}}$ test F1 scores for XLM-R models, under various additional configurations. Row 1 is the same as row 3 of Table 2.

Method	Train Data	Train Lang	non-en q en,zh	non-en q en,ar	en q zh,en	en q ar,en	monolingual ar,ar	monolingual zh,zh	monolingual en,en	non-en xling ar,zh	non-en xling zh,ar	avg
PAXQA$_{\text{HWA}}$	lex cons + S	ar,zh	**78.9**	76.7	70.2	**69.2**	**67.0**	**71.1**	**84.8**	**64.3**	**64.9**	71.9
PAXQA$_{\text{HWA}}$	S + lex cons	ar	78.0	**77.2**	70.5	66.3	65.0	70.5	84.6	60.9	63.6	70.7
PAXQA$_{\text{HWA}}$	S + lex cons	zh	**78.9**	73.7	68.7	68.3	66.7	69.2	84.2	63.9	62.4	70.7
PAXQA$_{\text{HWA}}$	S + lex cons + ar↔zh pivot	ar,zh	77.1	74.7	67.2	65.3	63.9	67.2	82.7	61.4	62.3	69.1

Table 7: MLQA test F1 scores for PAXQA$_{\text{HWA}}$ models, under variable additional configurations. Row 1 is the same as row 5 of Table 3. Row 4 additionally adds cross-lingual instances between ar and zh, generated through the pivoting strategy described in Section 7.2.

Method	F1	EM
SQuAD (S)	65.7	47.7
PAXQA$_{\text{HWA}}$ lex cons + S	71.1	52.9
Riabi et al. (2021)	70.6	–

Table 8: MLQA F1 and EM test scores, averaged across all 49 directions. Note that the PAXQA$_{\text{HWA}}$ is zero-shot with respect to 4 of the 7 languages covered by MLQA, while (Riabi et al., 2021)'s uses all 7 languages.

c/q	en	es	de	ar	hi	vi	zh
en	72.0	57.7	56.1	48.5	55.5	54.8	47.1
es	67.4	56.9	56.9	46.0	51.5	52.2	42.9
de	68.6	55.3	55.2	48.3	54.8	51.9	45.3
ar	63.2	51.3	51.8	46.2	48.1	50.7	43.6
hi	66.6	53.1	50.3	43.1	53.8	50.3	43.2
vi	65.4	55.6	52.4	45.4	51.4	54.3	45.7
zh	65.1	53.5	51.4	43.1	50.1	50.6	47.7

Table 9: MLQA EM test scores for each direction, using our best model (PAXQA$_{\text{HWA}}$ lex cons + SQuAD).

c/q	en	es	de	ar	hi	vi	zh
en	84.8	75.7	70.8	69.2	73.2	75.7	70.2
es	81.1	75.0	72.1	66.5	69.9	72.5	67.0
de	81.3	74.0	70.0	67.8	70.6	73.2	68.6
ar	76.7	68.9	65.7	67.0	65.5	71.2	64.9
hi	79.9	71.1	64.6	63.4	71.9	70.8	66.5
vi	78.4	71.7	66.1	65.5	68.7	75.2	68.4
zh	78.9	71.5	67.3	64.3	68.1	72.7	71.1

Table 10: MLQA F1 test scores for each direction, using our best model (PAXQA$_{\text{HWA}}$ lex cons + SQuAD).

and relevance. This is likely because we did not train workers, and merely provided them with the instructions. κ for (iii) and (iv) indicate moderate agreement.

We define 'high-quality' QA entries with the following criterion: either (i) or (ii) is 'Yes', (iii) is 'Yes' , and (iv) is 'Yes'. From the 2921 annotations, we filter to 1724 (59.0%) high-quality QA entries, which we then assign to the PAXQA$_{\text{HWA}}$ validation and test sets.

In other words, for the English QG model used in this work, 59.0% of generations were deemed acceptable by human annotators. As our methodology supports drop-in replacements, we suspect that using better QG models will improve QG quality, and likely downstream QA performance.

Field	Text
context_{en}	The scientists used a centrifuge from a nuclear weapon manufactured in the former-Soviet Union to obtain high purity silicon, then forged the obtained crystal into the most precise spherosome using hi-tech procedures based on a weight standard of "1kg". At the same time, they used x-ray crystal detector to measure the distance between the spherosome's silicon-28 atoms to determine if the spherosome undergoes obvious atomic changes under certain extreme conditions. In 1889, it was set at a standard one kilogram at the First General Conference of Weights and Measures.
context_{zh}	科学家们使用前苏联制造核武器的离心机来提取最高纯度的硅，并把得到的晶体通过高科技手段，按照“1 公斤” 的重量标准打造出这个最精准的圆球体。同时，利用X 射线晶体检测器来测量球体硅－28 原子之间的空间距离，确定在一些极端条件下该球体不发生明显的原子变化。在1889 年第一届国际计量大会上被定为1 千克的标准。
question_{en}	What did the scientists use to obtain high purity silicon?
question_{zh}	科学家们用什么来获得最高纯度的硅?
answer_{en}	a centrifuge
answer_{zh}	离心机

Table 11: Sample entry from the PAXQA$_{\text{HWA}}$ zh-en dataset. The non-English question is translated using lexically-constrained MT.

Field	Text
context_{en}	They cite as evidence the influx of thousands of tourists and visitors to the major international museums to see the best works of classical artists from by-gone ages. They also believe that a great artist can not engage in contemporary, modern and new schools, and be proficient in them, unless he is first proficient in classicism. They recall that Picasso himself, one of the most significant figures to break with classicism, was one of its most proficient exponents in his early career. This also applies to our senior sculpture, Wajih Nahlah, whose seventieth birthday we celebrated yesterday (along with Valentine's day). He himself is a major lover: of the brush, of diligent work, of sublime human and artistic beauty.
context_{ar}	و يستشهدون ب تدفق آلاف السياح و الزوار على متاحف العالم الكبرى ل مشاهدة روائع الكلاسيكيين من العصور الخوالي. و يرون اكثر: لا يمكن الفنان العظيم ان يخرج الى المدارس العصرية و الجديدة و الحديثة و يبرع في ها، الا اذا كان متمكناً من الكلاسيكية. و يذكرون ان بيكاسو نفس ه، و هو من اكبر من كسروا مع الكلاسيكية، كان من ابرع محترفي ها في مطالع ه الأولى. هذا الكلام ينطبق على كبير نا التشكيلي وجيه نحلة الذي احتفل نا امس ب الذات (مع عيد العشاق) ب عيد ميلاد ه السبعين، و هو نفس ه من كبار العاشقين: ريشةً و عملاً دؤوباً و جمالاً فنياً و انسانياً راقياً.
question_{en}	What ideology did Picasso break with in his early career?
question_{ar}	ما هي الإيديولوجية التي انفصل عنها بيكاسو في مطالع ه الأولى؟
answer_{en}	classicism
answer_{ar}	الكلاسيكية

Table 12: Sample entry from the PAXQA$_{\text{HWA}}$ ar-en dataset. The non-English question is translated using lexically-constrained MT.

System	Translated Question
Vanilla NMT	科学家们用什么来获得高纯度硅?
Lexically-constrained NMT	科学家们用什么来获得最高纯度的硅?
Google Translate	科学家用什么来获得高纯度硅?

Table 13: Translations from the different systems for the English question “What did the scientists use to obtain high purity silicon?”. The induced lexical constraints are ‘the scientists’ → ‘科学家们’ and ‘high purity silicon’ → ‘最高纯度的硅’, and are highlighted in each translation if they exist. NOTE: in this case, even though only the second translation satisfies all constraints, all 3 translations are grammatically and semantically correct.

Detailed Instructions (Click to expand)

Example Annotations (Click to expand)

The following **question** and **answer** have been generated from the given **context** sentence(s). Please read all three, then evaluate the quality by answering several yes/no statements. For more details, you may expand the buttons above. Please do not rush through the task.

Context: As John Maynard Keynes recognized, trade restrictions can protect or generate employment during economic recessions. But what may be desirable under extreme conditions for a single country can be highly detrimental to the world economy.

Question: What can protect or generate employment during economic recessions?

Answer: trade restrictions

1. Does the question make sense outside of the immediate context?

   ○ Yes ○ No

2. Is the question relevant and/or interesting?

   ○ Yes ○ No

3. Is the answer to the question correct?

   ○ Yes ○ No

4. Alternate Answer 1: Trade restrictions restricted trade

   Do "Answer" (trade restrictions) and "Alternate Answer 1" mean the same thing?

   ○ Yes ○ Possibly ○ No

5. Alternate Answer 2: Trade restriction

   Do "Answer" (trade restrictions) and "Alternate Answer 2" mean the same thing?

   ○ Yes ○ Possibly ○ No

Submit

Figure 3: Example QA generation evaluation task presented to human annotators. Note that the task focuses on evaluating the English side of the QA generations. The 'Alternate Answer' are the non-English answer spans back-translated to English; we use them as a proxy to evaluate the non-English answers. In this example, the correct answers would be 1. Yes; 2. Yes; 3. Yes; 4. Possibly; 5. Yes.