MLQA: Evaluating Cross-lingual Extractive Question Answering

Patrick Lewis[*][†]   **Barlas Oğuz**[*]   **Ruty Rinott**[*]   **Sebastian Riedel**[*][†]   **Holger Schwenk**[*]

[*]Facebook AI Research   [†]University College London

{plewis,barlaso,ruty,sriedel,schwenk}@fb.com

Abstract

Question answering (QA) models have shown rapid progress enabled by the availability of large, high-quality benchmark datasets. Such annotated datasets are difficult and costly to collect, and rarely exist in languages other than English, making building QA systems that work well in other languages challenging. In order to develop such systems, it is crucial to invest in high quality multilingual *evaluation* benchmarks to measure progress. We present MLQA, a multi-way aligned extractive QA evaluation benchmark intended to spur research in this area.[1] MLQA contains QA instances in 7 languages, *English, Arabic, German, Spanish, Hindi, Vietnamese* and *Simplified Chinese*. MLQA has over 12K instances in English and 5K in each other language, with each instance parallel between 4 languages on average. We evaluate state-of-the-art cross-lingual models and machine-translation-based baselines on MLQA. In all cases, transfer results are significantly behind training-language performance.

1 Introduction

Question answering (QA) is a central and highly popular area in NLP, with an abundance of datasets available to tackle the problem from various angles, including extractive QA, cloze-completion, and open-domain QA (Richardson, 2013; Rajpurkar et al., 2016; Chen et al., 2017; Kwiatkowski et al., 2019). The field has made rapid advances in recent years, even exceeding human performance in some settings (Devlin et al., 2019; Alberti et al., 2019).

Despite such popularity, QA datasets in languages other than English remain scarce, even for relatively high-resource languages (Asai et al., 2018), as collecting such datasets at sufficient scale and quality is difficult and costly. There

are two reasons why this lack of data prevents internationalization of QA systems. First, we cannot *measure* progress on multilingual QA without relevant benchmark data. Second, we cannot easily *train* end-to-end QA models on the task, and arguably most recent successes in QA have been in fully supervised settings. Given recent progress in cross-lingual tasks such as document classification (Lewis et al., 2004; Klementiev et al., 2012; Schwenk and Li, 2018), semantic role labelling (Akbik et al., 2015) and NLI (Conneau et al., 2018), we argue that while multilingual QA training data might be useful but not strictly necessary, multilingual evaluation data is a must-have.

Recognising this need, several cross-lingual datasets have recently been assembled (Asai et al., 2018; Liu et al., 2019a). However, these generally cover only a small number of languages, combine data from different authors and annotation protocols, lack parallel instances, or explore less practically-useful QA domains or tasks (see Section 3). Highly parallel data is particularly attractive, as it enables fairer comparison across languages, requires fewer source language annotations, and allows for additional evaluation setups at no extra annotation cost. A purpose-built evaluation benchmark dataset covering a range of diverse languages, and following the popular extractive QA paradigm on a practically-useful domain would be a powerful testbed for cross-lingual QA models.

With this work, we present such a benchmark, MLQA, and hope that it serves as an accelerator for multilingual QA in the way datasets such as SQuAD (Rajpurkar et al., 2016) have done for its monolingual counterpart. MLQA is a multi-way parallel extractive QA evaluation benchmark in seven languages: *English, Arabic, German, Vietnamese, Spanish, Simplified Chinese* and *Hindi*. To construct MLQA, we first automatically identify sentences from Wikipedia articles which have the same or similar meaning in multiple languages. We

---

[1]MLQA is publicly available at https://github.com/facebookresearch/mlqa

extract the paragraphs that contain such sentences, then crowd-source questions on the English paragraphs, making sure the answer is in the aligned sentence. This makes it possible to answer the question in all languages in the vast majority of cases.[2] The generated questions are then translated to all target languages by professional translators, and answer spans are annotated in the aligned contexts for the target languages.

The resulting corpus has between 5,000 and 6,000 instances in each language, and more than 12,000 in English. Each instance has an aligned equivalent in multiple other languages (always including English), the majority being 4-way aligned. Combined, there are over 46,000 QA annotations.

We define two tasks to assess performance on MLQA. The first, cross-lingual transfer (XLT), requires models trained in one language (in our case English) to transfer to test data in a different language. The second, generalised cross-lingual transfer (G-XLT) requires models to answer questions where the question and context language is *different*, e.g. questions in Hindi and contexts in Arabic, a setting possible because MLQA is highly parallel.

We provide baselines using state-of-the-art cross-lingual techniques. We develop machine translation baselines which map answer spans based on the attention matrices from a translation model, and use multilingual BERT (Devlin et al., 2019) and XLM (Lample and Conneau, 2019) as zero-shot approaches. We use English for our training language and adopt SQuAD as a training dataset. We find that zero-shot XLM transfers best, but all models lag well behind training-language performance.

In summary, we make the following contributions: i) We develop a novel annotation pipeline to construct large multilingual, highly-parallel extractive QA datasets ii) We release MLQA, a 7-language evaluation dataset for cross-lingual QA iii) We define two cross-lingual QA tasks, including a novel generalised cross-lingual QA task iv) We provide baselines using state-of-the-art techniques, and demonstrate significant room for improvement.

2   The MLQA corpus

First, we state our desired properties for a cross-lingual QA evaluation dataset. We note that whilst some existing datasets exhibit these properties,

none exhibit them all in combination (see section 3). We then describe our annotation protocol, which seeks to fulfil these desiderata.

Parallel   The dataset should consist of instances that are parallel across many languages. First, this makes comparison of QA performance as a function of transfer language fairer. Second, additional evaluation setups become possible, as questions in one language can be applied to documents in another. Finally, annotation cost is also reduced as more instances can be shared between languages.

Natural Documents   Building a parallel QA dataset in many languages requires access to parallel documents in those languages. Manually translating documents at sufficient scale entails huge translator workloads, and could result in unnatural documents. Exploiting existing naturally-parallel documents is advantageous, providing high-quality documents without requiring manual translation.

Diverse Languages   A primary goal of cross-lingual research is to develop systems that work well in many languages. The dataset should enable quantitative performance comparison across languages with different linguistic resources, language families and scripts.

Extractive QA   Cross-lingual understanding benchmarks are typically based on classification (Conneau et al., 2018). Extracting spans in different languages represents a different language understanding challenge. Whilst there are extractive QA datasets in a number of languages (see Section 3), most were created at different times by different authors with different annotation setups, making cross-language analysis challenging.

Textual Domain   We require a naturally highly language-parallel textual domain. Also, it is desirable to select a textual domain that matches existing extractive QA training resources, in order to isolate the change in performance due to language transfer.

To satisfy these desiderata, we identified the method described below and illustrated in Figure 1. Wikipedia represents a convenient textual domain, as its size and multi-linguality enables collection of data in many diverse languages at scale. It has been used to build many existing QA training resources, allowing us to leverage these to train QA models, without needing to build our own training dataset. We choose English as our source language as it has the largest Wikipedia, and to easily source crowd
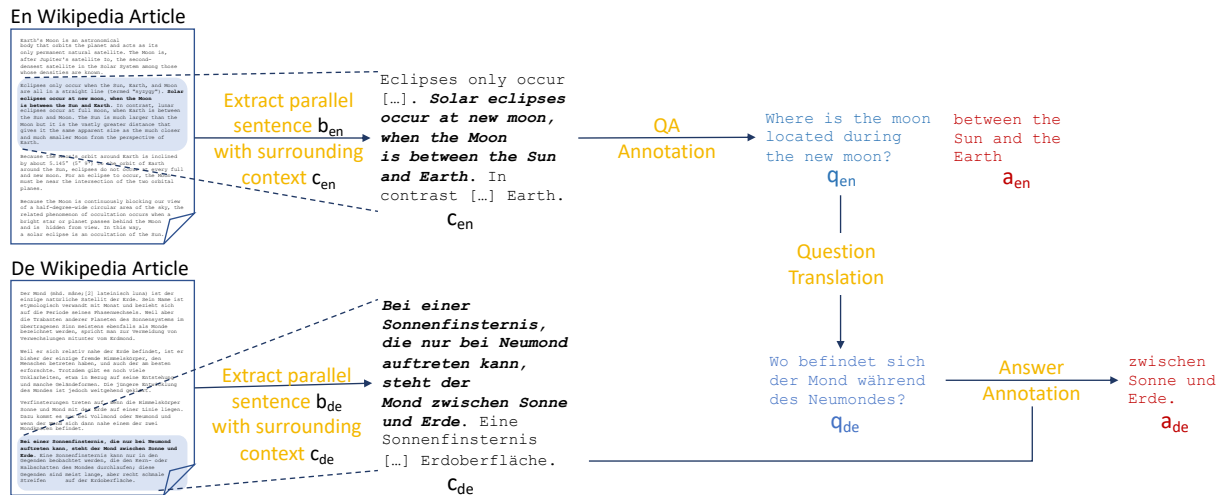
---

[2]The automatically aligned sentences occasionally differ in a named entity or information content, or some questions may not make sense without the surrounding context. In these rare cases, there may be no answer for some languages.

Figure 1: MLQA annotation pipeline. Only one target language is shown for clarity. Left: We first identify N-way parallel sentences $b_{en}, b_1 \ldots b_{N-1}$ in Wikipedia articles on the same topic, and extract the paragraphs that contain them, $c_{en}, c_1 \ldots c_{N-1}$. Middle: Workers formulate questions q_{en} from c_{en} for which answer a_{en} is a span within b_{en}. Right: English questions q_{en} are then translated by professional translators into all languages q_i and the answer a_i is annotated in the target language context c_i such that a_i is a span within b_i.

workers. We choose six other languages which represent a broad range of linguistic phenomena and have sufficiently large Wikipedia. Our annotation pipeline consists of three main steps:

Step 1) We automatically extract paragraphs which contain a parallel sentence from articles on the same topic in each language (left of Figure 1).

Step 2) We employ crowd-workers to annotate questions and answer spans on the English paragraphs (centre of Figure 1). Annotators must choose answer spans within the parallel source sentence. This allows annotation of questions in the source language with high probability of being answerable in the target languages, even if the rest of the context paragraphs are different.

Step 3) We employ professional translators to translate the questions and to annotate answer spans in the target language (right of Figure 1).

The following sections describe each step in the data collection pipeline in more detail.

2.1 Parallel Sentence Mining

Parallel Sentence mining allows us to leverage naturally-written documents and avoid translation, which would be expensive and result in potentially unnatural documents. In order for questions to be answerable in every target language, we use contexts containing an N-way parallel sentence. Our approach is similar to WikiMatrix (Schwenk et al., 2019) which extracts parallel sentences for many language pairs in Wikipedia, but we limit the search

de	es	ar	zh	vi	hi
5.4M	1.1M	83.7k	24.1K	9.2k	1340

Table 1: Incremental alignment with English to obtain 7-way aligned sentences.

for parallel sentences to documents on the same topic only, and aim for N-way parallel sentences.

To detect parallel sentences we use the LASER toolkit,[3] which achieves state-of-the-art performance in mining parallel sentences (Artetxe and Schwenk, 2019). LASER uses multilingual sentence embeddings and a distance or margin criterion in the embeddings space to detect parallel sentences. The reader is referred to Artetxe and Schwenk (2018) and Artetxe and Schwenk (2019) for a detailed description. See Appendix A.6 for further details and statistics on the number of parallel sentences mined for all language pairs.

We first independently align all languages with English, then intersect these sets of parallel sentences, forming sets of N-way parallel sentences. As shown in Table 1, starting with 5.4M parallel English/German sentences, the number of N-way parallel sentences quickly decreases as more languages are added. We also found that 7-way parallel sentences lack linguistic diversity, and often appear in the first sentence or paragraph of articles.

As a compromise between language-parallelism

---

[3] https://github.com/facebookresearch/LASER

and both the number and diversity of parallel sentences, we use sentences that are 4-way parallel. This yields 385,396 parallel sentences (see Appendix A.6) which were sub-sampled to ensure parallel sentences were evenly distributed in paragraphs. We ensure that each language combination is equally represented, so that each language has many QA instances in common with every other language. Except for any rejected instances later in the pipeline, each QA instance will be parallel between English and three target languages.

2.2 English QA Annotation

We use Amazon Mechanical Turk to annotate English QA instances, broadly following the methodology of Rajpurkar et al. (2016). We present workers with an English aligned sentence, b_{en} along with the paragraph that contains it c_{en}. Workers formulate a question q_{en} and highlight the shortest answer span a_{en} that answers it. a_{en} must be be a subspan of b_{en} to ensure q_{en} will be answerable in the target languages. We include a "No Question Possible" button when no sensible question could be asked. Screenshots of the annotation interface can be found in Appendix A.1. The first 15 questions from each worker are manually checked, after which the worker is contacted with feedback, or their work is auto-approved.

Once the questions and answers have been annotated, we run another task to re-annotate English answers. Here, workers are presented with q_{en} and c_{en}, and requested to generate an a'_{en} or to indicate that q_{en} is not answerable. Two additional answer span annotations are collected for each question. The additional answer annotations enable us to calculate an inter-annotator agreement (IAA) score. We calculate the mean token F1 score between the three answer annotations, giving an IAA score of 82%, comparable to the SQuAD v1.1 development set, where this IAA measure is 84%.

Rather than provide all three answer annotations as gold answers, we select a single representative reference answer. In 88% of cases, either two or three of the answers exactly matched, so the majority answer is selected. In the remaining cases, the answer with highest F1 overlap with the other two is chosen. This results both in an accurate answer span, and ensures the English results are comparable to those in the target languages, where only one answer is annotated per question.

We discard instances where annotators marked the question as unanswerable as well as instances where over 50% of the question appeared as a subsequence of the aligned sentence, as these are too easy or of low quality. Finally, we reject questions where the IAA score was very low (< 0.3) removing a small number of low quality instances. To verify we were not discarding challenging but high quality examples in this step, a manual analysis of discarded questions was performed. Of these discarded questions, 38% were poorly specified, 24% did not make sense/had no answer, 30% had poor answers, and only 8% were high quality challenging questions.

2.3 Target Language QA Annotation

We use the One Hour Translation platform to source professional translators to translate the questions from English to the six target languages, and to find answers in the target contexts. We present each translator with the English question q_{en}, English answer a_{en}, and the context c_x (containing aligned sentence b_x) in target language x. The translators are only shown the aligned sentence and the sentence on each side (where these exist). This increases the chance of the question being answerable, as in some cases the aligned sentences are not perfectly parallel, without requiring workers to read the entire context c_x. By providing the English answer we try to minimize cultural and personal differences in the amount of detail in the answer.

We sample 2% of the translated questions for additional review by language experts. Translators that did not meet the quality standards were removed from the translator pool, and their translations were reallocated. By comparing the distribution of answer lengths relative to the context to the English distribution, some cases were found where some annotators selected very long answers, especially for Chinese. We clarified the instructions with these specific annotators, and send such cases for re-annotation. We discard instances in target languages where annotators indicate there is no answer in that language. This means some instances are not 4-way parallel. "No Answer" annotations occurred for 6.6%-21.9% of instances (Vietnamese and German, respectively). We release the "No Answer" data separately as an additional resource, but do not consider it in our experiments or analysis.

2.4 The Resulting MLQA corpus

Contexts, questions and answer spans for all the languages are then brought together to create the

fold	en	de	es	ar	zh	vi	hi
dev	1148	512	500	517	504	511	507
test	11590	4517	5253	5335	5137	5495	4918

Table 2: Number of instances per language in MLQA.

	de	es	ar	zh	vi	hi
de	5029					
es	1972	5753				
ar	1856	2139	5852			
zh	1811	2108	2100	5641		
vi	1857	2207	2210	2127	6006	
hi	1593	1910	2017	2124	2124	5425

Table 3: Number of parallel instances between target language pairs (all instances are parallel with English).

final corpus. MLQA consists of 12,738 extractive QA instances in English and between 5,029 and 6,006 instances in the target languages. 9,019 instances are 4-way parallel, 2,930 are 3-way parallel and 789 2-way parallel. Representative examples are shown in Figure 2. MLQA is split into development and test splits, with statistics in Tables 2, 3 and 4. To investigate the distribution of topics in MLQA, a random sample of 500 articles were manually analysed. Articles cover a broad range of topics across different cultures, world regions and disciplines. 23% are about people, 19% on physical places, 13% on cultural topics, 12% on science/engineering, 9% on organisations, 6% on events and 18% on other topics. Further statistics are given in Appendix A.2.

	en	de	es	ar	zh	vi	hi
# Articles	5530	2806	2762	2627	2673	2682	2255
# Contexts	10894	4509	5215	5085	4989	5246	4524
# Instances	12738	5029	5753	5852	5641	6006	5425

Table 4: Number of Wikipedia articles with a context in MLQA.

3 Related Work

Monolingual QA Data There is a great variety of English QA data, popularized by MCTest (Richardson, 2013), CNN/Daily Mail (Hermann et al., 2015) CBT (Hill et al., 2016), and Wik-iQA (Yang et al., 2015) amongst others. Large span-based datasets such as SQuAD (Rajpurkar et al., 2016, 2018), TriviaQA (Joshi et al., 2017), NewsQA (Trischler et al., 2017), and Natural Questions (Kwiatkowski et al., 2019) have seen extractive QA become a dominant paradigm. However,

large, high-quality datasets in other languages are relatively rare. There are several Chinese datasets, such as DUReader (He et al., 2018), CMRC (Cui et al., 2019b) and DRCD (Shao et al., 2018). More recently, there have been efforts to build corpora in a wider array of languages, such as Korean (Lim et al., 2019) and Arabic (Mozannar et al., 2019).

Cross-lingual QA Modelling Cross-lingual QA as a discipline has been explored in QA for RDF data for a number of years, such as the QALD-3 and 5 tracks (Cimiano et al., 2013; Unger et al., 2015), with more recent work from Zimina et al. (2018). Lee et al. (2018) explore an approach to use English QA data from SQuAD to improve QA performance in Korean using an in-language seed dataset. Kumar et al. (2019) study question generation by leveraging English questions to generate better Hindi questions, and Lee and Lee (2019) and Cui et al. (2019a) develop modelling approaches to improve performance on Chinese QA tasks using English resources. Lee et al. (2019) and Hsu et al. (2019) explore modelling approaches for zero-shot transfer and Singh et al. (2019) explore how training with cross-lingual data regularizes QA models.

Cross-lingual QA Data Gupta et al. (2018) release a parallel QA dataset in English and Hindi, Hardalov et al. (2019) investigate QA transfer from English to Bulgarian, Liu et al. (2019b) release a cloze QA dataset in Chinese and English, and Jing et al. (2019) released BiPar, built using parallel paragraphs from novels in English and Chinese. These datasets have a similar spirit to MLQA, but are limited to two languages. Asai et al. (2018) investigate extractive QA on a manually-translated set of 327 SQuAD instances in Japanese and French, and develop a phrase-alignment modelling technique, showing improvements over back-translation. Like us, they build multi-way parallel extractive QA data, but MLQA has many more instances, covers more languages and does not require manual document translation. Liu et al. (2019a) explore cross-lingual open-domain QA with a dataset built from Wikipedia "Did you know?" questions, covering nine languages. Unlike MLQA, it is distantly supervised, the dataset size varies by language, instances are not parallel, and answer distributions vary by language, making quantitative comparisons across languages challenging. Finally, in contemporaneous work, Artetxe et al. (2019) release XQuAD, a dataset of

Figure 2: (a) MLQA example parallel for En-De-Ar-Vi. (b) MLQA example parallel for En-Es-Zh-Hi. Answers shown as highlighted spans in contexts. Contexts shortened for clarity with "[...]".

1190 SQuAD instances from 240 paragraphs manually translated into 10 languages. As shown in Table 4, MLQA covers 7 languages, but contains more data per language – over 5k QA pairs from 5̃k paragraphs per language. MLQA also uses real Wikipedia contexts rather than manual translation.

Aggregated Cross-lingual Benchmarks Recently, following the widespread adoption of projects such as GLUE (Wang et al., 2019), there have been efforts to compile a suite of high quality multilingual tasks as a unified benchmark system. Two such projects, XGLUE (Liang et al., 2020) and XTREME (Hu et al., 2020) incorporate MLQA as part of their aggregated benchmark.

4   Cross-lingual QA Experiments

We introduce two tasks to assess cross-lingual QA performance with MLQA. The first, *cross-lingual transfer* (XLT), requires training a model with (c_x, q_x, a_x) training data in language x, in our case English. Development data in language x is used for tuning. At test time, the model must extract answer a_y in language y given context c_y and question q_y. The second task, *generalized cross-lingual transfer* (G-XLT), is trained in the same way, but at test time the model must extract a_z from c_z in language z given q_y in language y. This evaluation setup is possible because MLQA is highly parallel,

allowing us to swap q_z for q_y for parallel instances without changing the question's meaning.

As MLQA only has development and test data, we adopt SQuAD v1.1 as training data. We use MLQA-en as development data, and focus on zero-shot evaluation, where no training or development data is available in target languages. Models were trained with the SQuAD-v1 training method from Devlin et al. (2019) and implemented in Pytext (Aly et al., 2018). We establish a number of baselines to assess current cross-lingual QA capabilities:

Translate-Train We translate instances from the SQuAD training set into the target language using machine-translation.[4] Before translating, we enclose answers in quotes, as in Lee et al. (2018). This makes it easy to extract answers from translated contexts, and encourages the translation model to map answers into single spans. We discard instances where this fails (∼5%). This corpus is then used to train a model in the target language.

Translate-Test The context and question in the target language is translated into English at test time. We use our best English model to produce an answer span in the translated paragraph. For all languages other than Hindi,[5] we use attention

---

[4]We use Facebook's production translation models.

[5]Alignments were unavailable for Hindi-English due to production model limitations. Instead we translate English

scores, a_{ij}, from the translation model to map the answer back to the original language. Rather than aligning spans by attention argmax, as by Asai et al. (2018), we identify the span in the original context which maximizes F1 score with the English span:

$$RC = \sum_{i \in S_e, j \in S_o} a_{ij} \Big/ \sum_{i \in S_e} a_{i*}$$
$$PR = \sum_{i \in S_e, j \in S_o} a_{ij} \Big/ \sum_{j \in S_o} a_{*j}$$
$$F1 = 2 * RC * PR \big/ RC + PR \qquad (1)$$
$$\text{answer} = \arg\max_{S_o} F1(S_o)$$

where S_e and S_o are the English and original spans respectively, $a_{i*} = \sum_j a_{ij}$ and $a_{*j} = \sum_i a_{*j}$.

Cross-lingual Representation Models We produce zero-shot transfer results from multilingual BERT (cased, 104 languages) (Devlin et al., 2019) and XLM (MLM + TLM, 15 languages) (Lample and Conneau, 2019). Models are trained with the SQuAD training set and evaluated directly on the MLQA test set in the target language. Model selection is also constrained to be strictly zero-shot, using only English development data to pick hyper-parameters. As a result, we end up with a single model that we test for all 7 languages.

4.1 Evaluation Metrics for Multilingual QA

Most extractive QA tasks use Exact Match (EM) and mean token F1 score as performance metrics. The widely-used SQuAD evaluation also performs the following answer-preprocessing operations: i) lowercasing, ii) stripping (ASCII) punctuation iii) stripping (English) articles and iv) whitespace tokenisation. We introduce the following modifications for fairer multilingual evaluation: Instead of stripping ASCII punctuation, we strip all unicode characters with a punctuation *General_Category*.[6] When a language has stand-alone articles (English, Spanish, German and Vietnamese) we strip them. We use whitespace tokenization for all MLQA languages other than Chinese, where we use the mixed segmentation method from Cui et al. (2019b).

5 Results

5.1 XLT Results

Table 5 shows the results on the XLT task. XLM performs best overall, transferring best in Span-

answers using another round of translation. Back-translated answers may not map back to spans in the original context, so this Translate-Test performs poorly.

[6]http://www.unicode.org/reports/tr44/tr44-4.html#General_Category_Values

Figure 3: F1 score stratified by English wh* word, relative to overall F1 score for XLM

ish, German and Arabic, and competitively with translate-train+M-BERT for Vietnamese and Chinese. XLM is however, weaker in English. Even for XLM, there is a 39.8% drop in mean EM score (20.9% F1) over the English BERT-large baseline, showing significant room for improvement. All models generally struggle on Arabic and Hindi.

A manual analysis of cases where XLM failed to exactly match the gold answer was carried out for all languages. 39% of these errors were completely wrong answers, 5% were annotation errors and 7% were acceptable answers with no overlap with the gold answer. The remaining 49% come from answers that partially overlap with the gold span. The variation of errors across languages was small.

To see how performance varies by question type, we compute XLM F1 scores stratified by common English wh-words. Figure 3 shows that "When" questions are the easiest for all languages, and "Where" questions seem challenging in most target languages. Further details are in Appendix A.3.

To explore whether questions that were difficult for the model in English were also challenging in the target languages, we split MLQA into two subsets on whether the XLM model got an English F1 score of zero. Figure 4 shows that transfer performance is better when the model answers well in English, but is far from zero when the English answer is wrong, suggesting some questions may be easier to answer in some languages than others.

5.2 G-XLT Results

Table 6 shows results for XLM on the G-XLT task.[7] For questions in a given language, the model performs best when the context language matches the question, except for Hindi and Arabic. For con-

[7]Additional results may be found in Appendix A.4

F1 / EM	en	es	de	ar	hi	vi	zh
BERT-Large	**80.2 / 67.4**	-	-	-	-	-	-
Multilingual-BERT	77.7 / 65.2	64.3 / 46.6	57.9 / 44.3	45.7 / 29.8	43.8 / 29.7	57.1 / 38.6	57.5 / 37.3
XLM	74.9 / 62.4	**68.0 / 49.8**	**62.2 / 47.6**	**54.8 / 36.3**	48.8 / 27.3	61.4 / 41.8	61.1 / **39.6**
Translate test, BERT-L	-	65.4 / 44.0	57.9 / 41.8	33.6 / 20.4	23.8 / 18.9*	58.2 / 33.2	44.2 / 20.3
Translate train, M-BERT	-	53.9 / 37.4	62.0 / 47.5	51.8 / 33.2	**55.0 / 40.0**	**62.0 / 43.1**	**61.4** / 39.5
Translate train, XLM	-	65.2 / 47.8	61.4 / 46.7	54.0 / 34.4	50.7 / 33.4	59.3 / 39.4	59.8 / 37.9

Table 5: F1 score and Exact Match on the MLQA test set for the cross-lingual transfer task (XLT)
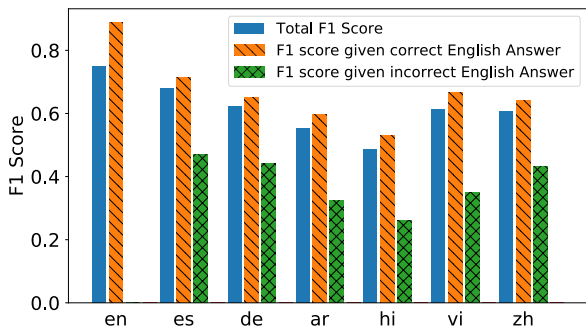

Figure 4: XLM F1 score stratified by English difficulty

texts in a given language, English questions tend to perform best, apart from Chinese and Vietnamese.

c/q	en	es	de	ar	hi	vi	zh
en	74.9	65.0	58.5	50.8	43.6	55.7	53.9
es	69.5	68.0	61.7	54.0	49.5	58.1	56.5
de	70.6	67.7	62.2	57.4	49.9	60.1	57.3
ar	60.0	57.8	54.9	54.8	42.4	50.5	43.5
hi	59.6	56.3	50.5	44.4	48.8	48.9	40.2
vi	60.2	59.6	53.2	48.7	40.5	61.4	48.5
zh	52.9	55.8	50.0	40.9	35.4	46.5	61.1

Table 6: F1 Score for XLM for G-XLT. Columns show question language, rows show context language.

5.3 English Results on SQuAD 1 and MLQA

The MLQA-en results in Table 5 are lower than reported results on SQuAD v1.1 in the literature for equivalent models. However, once SQuAD scores are adjusted to reflect only having one answer annotation (picked using the same method used to pick MLQA answers), the discrepancy drops to 5.8% on average (see Table 7). MLQA-en contexts are on average 28% longer than SQuAD's, and MLQA covers a much wider set of articles than SQuAD. Minor differences in preprocessing and answer lengths may also contribute (MLQA-en answers are slightly longer, 3.1 tokens vs 2.9 on average). Question type distributions are very similar in both datasets (Figure 7 in Appendix A)

Model	SQuAD	SQuAD*	MLQA-en
BERT-Large	91.0 / 80.8	84.8 / 72.9	80.2 / 67.4
M-BERT	88.5 / 81.2	83.0 / 71.1	77.7 / 65.1
XLM	87.6 / 80.5	82.1 / 69.7	74.9 / 62.4

Table 7: English performance comparisons to SQuAD using our models. * uses a single answer annotation.

6 Discussion

It is worth discussing the quality of context paragraphs in MLQA. Our parallel sentence mining approach can source independently-written documents in different languages, but, in practice, articles are often translated from English to the target languages by volunteers. Thus our method sometimes acts as an efficient mechanism of sourcing existing human translations, rather than sourcing independently-written content on the same topic. The use of machine translation is strongly discouraged by the Wikipedia community,[8] but from examining edit histories of articles in MLQA, machine translation is occasionally used as an article seed, before being edited and added to by human authors.

Our annotation method restricts answers to come from specified sentences. Despite being provided several sentences of context, some annotators may be tempted to only read the parallel sentence and write questions which only require a single sentence of context to answer. However, single sentence context questions are a known issue in SQuAD annotation in general (Sugawara et al., 2018) suggesting our method would not result in less challenging questions, supported by scores on MLQA-en being similar to SQuAD (section 5.3).

MLQA is partitioned into development and test splits. As MLQA is parallel, this means there is development data for every language. Since MLQA will be freely available, this was done to reduce the risk of test data over-fitting in future, and to estab-

lish standard splits. However, in our experiments, we only make use of the English development data and study strict zero-shot settings. Other evaluation setups could be envisioned, e.g. by exploiting the target language development sets for hyperparameter optimisation or fine-tuning, which could be fruitful for higher transfer performance, but we leave such "few-shot" experiments as future work. Other potential areas to explore involve training datasets other than English, such as CMRC (Cui et al., 2018), or using unsupervised QA techniques to assist transfer (Lewis et al., 2019).

Finally, a large body of work suggests QA models are over-reliant on word-matching between question and context (Jia and Liang, 2017; Gan and Ng, 2019). G-XLT represents an interesting testbed, as simple symbolic matching is less straightforward when questions and contexts use different languages. However, the performance drop from XLT is relatively small (8.2 mean F1), suggesting word-matching in cross-lingual models is more nuanced and robust than it may initially appear.

7 Conclusion

We have introduced MLQA, a highly-parallel multilingual QA benchmark in seven languages. We developed several baselines on two cross-lingual understanding tasks on MLQA with state-of-the-art methods, and demonstrate significant room for improvement. We hope that MLQA will help to catalyse work in cross-lingual QA to close the gap between training and testing language performance.

Acknowledgements

References

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating High Quality Proposition Banks for Multilingual Semantic Role Labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407, Beijing, China. Association for Computational Linguistics.

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.

Ahmed Aly, Kushal Lakhotia, Shicong Zhao, Mrinal Mohit, Barlas Oguz, Abhinav Arora, Sonal Gupta, Christopher Dewan, Stef Nelson-Lindall, and Rushin Shah. 2018. Pytext: A seamless path from nlp research to production. *arXiv preprint arXiv:1812.08729*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the Cross-lingual Transferability of Monolingual Representations. *arXiv:1910.11856 [cs]*. ArXiv: 1910.11856.

Mikel Artetxe and Holger Schwenk. 2018. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *arXiv:1812.10464 [cs]*. ArXiv: 1812.10464.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual Extractive Reading Comprehension by Runtime Machine Translation. *arXiv:1809.03275 [cs]*. ArXiv: 1809.03275.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Philipp Cimiano, Vanessa López, Christina Unger, Elena Cabrio, Axel-Cyrille Ngonga Ngomo, and Sebastian Walter. 2013. Multilingual Question Answering over Linked Data (QALD-3): Lab Overview. In *CLEF*.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. *arXiv:1809.05053 [cs]*. ArXiv: 1809.05053.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2019a. Cross-Lingual Machine Reading Comprehension. 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 1586–1595, Hong Kong, China. Association for Computational Linguistics.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019b. A Span-Extraction Dataset for Chinese Machine Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics.

Yiming Cui, Ting Liu, Li Xiao, Zhipeng Chen, Wentao Ma, Wanxiang Che, Shijin Wang, and Guoping Hu. 2018. A Span-Extraction Dataset for Chinese Machine Reading Comprehension. *arXiv:1810.07366 [cs]*. ArXiv: 1810.07366.

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.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the Robustness of Question Answering Systems to Question Paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. MMQA: A Multi-domain Multi-lingual Question-Answering Framework for English and Hindi. In *LREC*.

Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019. Beyond English-only Reading Comprehension: Experiments in Zero-Shot Multilingual Transfer for Bulgarian. *arXiv:1908.01519 [cs]*. ArXiv: 1908.01519.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Tsung-Yuan Hsu, Chi-Liang Liu, and Hung-yi Lee. 2019. Zero-shot Reading Comprehension by Cross-lingual Transfer Learning with Multi-lingual Language Representation Model. 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 5935–5942, Hong Kong, China. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *ArXiv*, abs/2003.11080.

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Yimin Jing, Deyi Xiong, and Zhen Yan. 2019. BiPaR: A Bilingual Parallel Dataset for Multilingual and Cross-lingual Reading Comprehension on Novels. 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 2452–2462, Hong Kong, China. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing Crosslingual Distributed Representations of Words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. 2019. Cross-Lingual Training for Automatic Question Generation. *arXiv:1906.02525 [cs]*. ArXiv: 1906.02525.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Red-field, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *arXiv:1901.07291 [cs]*. ArXiv: 1901.07291.

Chia-Hsuan Lee and Hung-Yi Lee. 2019. Cross-Lingual Transfer Learning for Question Answering. *arXiv:1907.06042 [cs]*. ArXiv: 1907.06042.

Kyungjae Lee, Sunghyun Park, Hojae Han, Jinyoung Yeo, Seung-won Hwang, and Juho Lee. 2019. Learning with Limited Data for Multilingual Reading Comprehension. 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 2833–2843, Hong Kong, China. Association for Computational Linguistics.

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).

David D. Lewis, Yiming yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *jmlr*, 5:361–397.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised Question Answering by Cloze Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Bruce Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Rangan Majumder, and Ming Zhou. 2020. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *ArXiv*, abs/2004.01401.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1.0: Korean qa dataset for machine reading comprehension. *arXiv:1909.07005v2 [cs.CL]*.

Jiahua Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2019a. XQA: A Cross-lingual Open-domain Question Answering Dataset. In *Proceedings of ACL 2019*.

Pengyuan Liu, Yuning Deng, Chenghao Zhu, and Han Hu. 2019b. XCMRC: Evaluating Cross-lingual Machine Reading Comprehension. *arXiv:1908.05416 [cs]*. ArXiv: 1908.05416.

Hussein Mozannar, Karl El Hajal, Elie Maamary, and Hazem Hajj. 2019. Neural Arabic Question Answering. *arXiv:1906.05394 [cs]*. ArXiv: 1906.05394.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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.

Matthew Richardson. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. 2018. DRCD: a Chinese Machine Reading Comprehension Dataset. *arXiv:1806.00920 [cs]*. ArXiv: 1806.00920.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. XLDA: Cross-Lingual Data Augmentation for Natural Language Inference and Question Answering. *arXiv:1905.11471 [cs]*. ArXiv: 1905.11471.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What Makes Reading Comprehension Questions Easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

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.

Christina Unger, Corina Forescu, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. 2015. Question Answering over Linked Data (QALD-5). In *CLEF*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Elizaveta Zimina, Jyrki Nummenmaa, Kalervo Jarvelin, Jaakko Peltonen, and Kostas Stefanidis. 2018. MuG-QA: Multilingual Grammatical Question Answering for RDF Data. *2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 57–61.
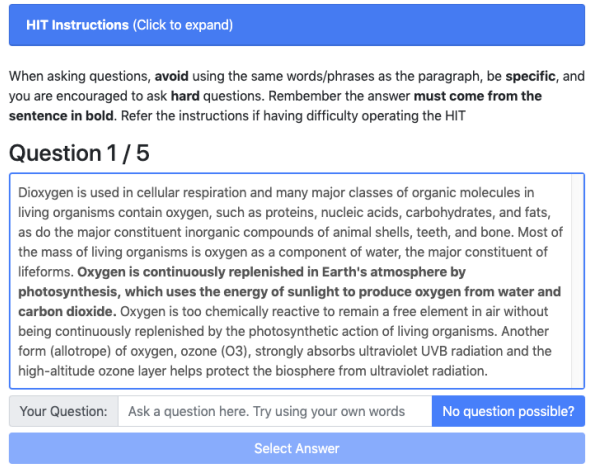
Figure 5: English QA annotation interface screenshot

	en	de	es	ar	zh*	vi	hi
Context	157.5	102.2	103.4	116.8	222.9	195.1	141.5
Question	8.4	7.7	8.6	7.6	14.3	10.6	9.3
Answer	3.1	3.2	4.1	3.4	8.2	4.5	3.6

Table 8: Mean Sequence lengths (tokens) in MLQA. *calculated with mixed segmentation (section 4.1)

A Appendices

A.1 Annotation Interface

Figure 5 shows a screenshot of the annotation interface. Workers are asked to write a question in the box, and highlight an answer using the mouse in the sentence that is in bold. There are a number of data input validation features to assist workers, as well as detailed instructions in a drop-down window, which are shown in Figure 6

A.2 Additional MLQA Statistics

Figure 7 shows the distribution of wh words in questions in both MLQA-en and SQuAD v.1.1. The distributions are very similar, suggesting training on SQuAD data is an appropriate training dataset choice.

Table 4 shows the number of Wikipedia articles that feature at least one of their paragraphs as a context paragraph in MLQA, along with the number of unique context paragraphs in MLQA. There are 1.9 context paragraphs from each article on average. This is in contrast to SQuAD, which instead features a small number of curated articles, but more densely annotated, with 43 context paragraphs per article on average. Thus, MLQA covers a much broader range of topics than SQuAD.

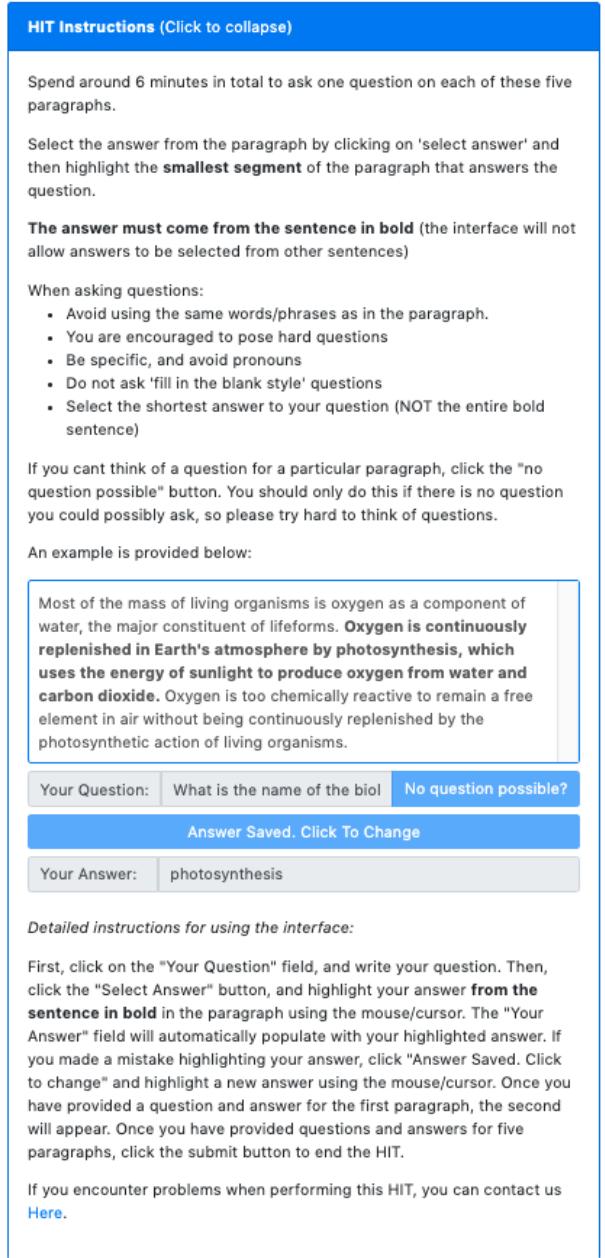Table 8 shows statistics about the lengths of con-

Figure 6: English annotation instructions screenshot

texts, questions and answers in MLQA. Vietnamese has the longest contexts on average and German are shortest, but all languages have a substantial tail of long contexts. Other than Chinese, answers are on average 3 to 4 tokens.

A.3 QA Performance stratified by question and answer types

To examine how performance varies across languages for different types of questions, we stratify MLQA with three criteria — By English Wh-word, by answer Named-Entity type and by English Question Difficulty
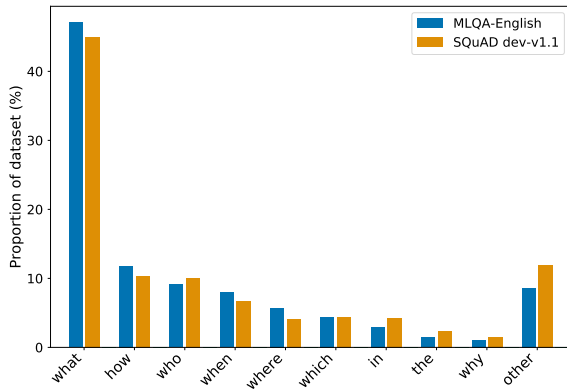
Figure 7: Question type distribution (by "wh" word) in MLQA-en and SQuAD V1.1. The distributions are strikingly similar
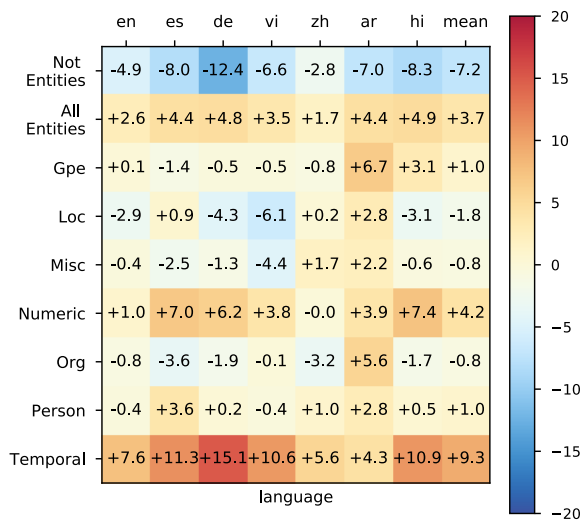

Figure 8: F1 score stratified by named entity types in answer spans, relative to overall F1 score for XLM

By wh-word: First, we split by the English Wh* word in the question. This resulting change in F1 score compared to the overall F1 score is shown in Figure 3, and discussed briefly in the main text. The English wh* word provides a clue as to the type of answer the questioner is expecting, and thus acts as a way of classifying QA instances into types. We chose the 5 most common wh* words in the dataset for this analysis. We see that "when" questions are consistently easier than average across the languages, but the pattern is less clear for other question types. "Who" questions also seem easier than average, except for Hindi, where the performance is quite low for these questions. "How"-type questions (such as "how much", "how many" or "how long") are also more challenging to answer than average in English compared to the other languages. "Where" questions also seem challenging

for Spanish, German, Chinese and Hindi, but this is not true for Arabic or Vietnamese.

By Named-Entity type We create subsets of MLQA by detecting which English named entities are contained in the answer span. To achieve this, we run Named Entity Recognition using SPaCy (Honnibal and Montani, 2017), and detect where named entity spans overlap with answer spans. The F1 scores for different answer types relative to overall F1 score are shown for various Named Entity types in Figure 8. There are some clear trends: Answer spans that contain named entities are easier to answer than those that do not (the first two rows) for all the languages, but the difference is most pronounced for German. Secondly, "Temporal" answer types (DATE and TIME entity labels) are consistently easier than average for all languages, consistent with the high scores for "when" questions in the previous section. Again, this result is most pronounced in German, but is also very strong for Spanish, Hindi, and Vietnamese. Arabic also performs well for ORG, GPE and LOC answer types, unlike most of the other languages. Numeric questions (CARDINAL, ORDINAL, PERCENT, QUANTITY and MONEY entity labels) also seem relatively easy for the model in most languages.

By English Question Difficulty Here, we split MLQA into two subsets, according to whether the XLM model got the question completely wrong (no word overlap with the correct answer). We then evaluated the mean F1 score for each language on the two subsets, with the results shown in Figure 4. We see that questions that are "easy" in English also seem to be easier in the target languages, but the drop in performance for the "hard" subset is not as dramatic as one might expect. This suggests that not all questions that are hard in English in MLQA are hard in the target languages. This could be due to the grammar and morphology of different languages leading to questions being easier or more difficult to answer, but an another factor is that context documents can be shorter in target languages for questions the model struggled to answer correctly in English, effectively making them easier. Manual inspection suggests that whilst context documents are often shorter for when the model is correct in the target language, this effect is not sufficient to explain the difference in performance.

A.4 Additional G-XLT results

Table 6 in the main text shows for XLM on the G-XLT task, and Table 9 for Multilingual-BERT respectively. XLM outperforms M-BERT for most language pairs, with a mean G-XLT performance of 53.4 F1 compared to 47.2 F1 (mean of off-diagonal elements of Tables 6 and 9). Multilingual BERT exhibits more of a preference for English than XLM for G-XLT, and exhibits a bigger performance drop going from XLT to G-XLT (10.5 mean drop in F1 compared to 8.2).

c/q	en	es	de	ar	hi	vi	zh
en	77.7	64.4	62.7	45.7	40.1	52.2	54.2
es	67.4	64.3	58.5	44.1	38.1	48.2	51.1
de	62.8	57.4	57.9	38.8	35.5	44.7	46.3
ar	51.2	45.3	46.4	45.6	32.1	37.3	40.0
hi	51.8	43.2	46.2	36.9	43.8	38.4	40.5
vi	61.4	52.1	51.4	34.4	35.1	57.1	47.1
zh	58.0	49.1	49.6	40.5	36.0	44.6	57.5

Table 9: F1 Score for M-BERT for G-XLT. Columns show question language, rows show context language.

A.5 Additional preprocessing Details

OpenCC (https://github.com/BYVoid/OpenCC) is used to convert all Chinese contexts to Simplified Chinese, as wikipedia dumps generally consist of a mixture of simplified and traditional Chinese text.

A.6 Further details on Parallel Sentence mining

Table 10 shows the number of mined parallel sentences found in each language, as function of how many languages the sentences are parallel between. As the number of languages that a parallel sentence is shared between increases, the number of such sentences decreases. When we look for 7-way aligned examples, we only find 1340 sentences from the entirety of the 7 Wikipedia. Additionally, most of these sentences are the first sentence of the article, or are uninteresting. However, if we choose 4-way parallel sentences, there are plenty of sentences to choose from. We sample evenly from each combination of English and 3 of the 6 target languages. This ensures that we have an even distribution over all the target languages, as well as ensuring we have even numbers of instances that will be parallel between target language combinations.

N-way	en	de	es	ar	zh	vi	hi
2	12219436	3925542	4957438	1047977	1174359	904037	210083
3	2143675	1157009	1532811	427609	603938	482488	83495
4	385396	249022	319902	148348	223513	181353	34050
5	73918	56756	67383	44684	58814	54884	13151
6	12333	11171	11935	11081	11485	11507	4486
7	1340	1340	1340	1340	1340	1340	1340

Table 10: Number of mined parallel sentences as a function of how many languages the sentences are parallel between