Cross-Lingual Open-Domain Question Answering with Answer Sentence Generation

Benjamin Muller[1]*, **Luca Soldaini**[2]†, **Rik Koncel-Kedziorski**[3], **Eric Lind**[3], **Alessandro Moschitti**[3]
[1]Inria, Paris, France
[2]Allen Istitute for AI
[3]Amazon Alexa AI
benjamin.muller@inria.fr, lucas@allenai.org,
{rikdz,ericlind,amosch}@amazon.com

Abstract

Open-Domain Generative Question Answering has achieved impressive performance in English by combining document-level retrieval with answer generation. These approaches, which we refer to as GENQA, can generate complete sentences, effectively answering both factoid and non-factoid questions. In this paper, we extend GENQA to the multilingual and cross-lingual settings. For this purpose, we first introduce GEN-TYDIQA, an extension of the TyDiQA dataset with well-formed and complete answers for Arabic, Bengali, English, Japanese, and Russian. Based on GEN-TYDIQA, we design a cross-lingual generative model that produces full-sentence answers by exploiting passages written in multiple languages, including languages different from the question. Our cross-lingual generative system outperforms answer sentence selection baselines for all 5 languages and monolingual generative pipelines for three out of five languages studied.

1 Introduction

Improving coverage of the world's languages is essential for retrieval-based Question Answering (QA) systems to provide a better experience for non-English speaking users. One promising direction for improving coverage is multilingual, multi-source, open-domain QA. Multilingual QA systems include diverse viewpoints by leveraging answers from multiple linguistic communities. Further, they can improve accuracy, as all facets necessary to answer a question are often unequally distributed across languages on the Internet (Valentim et al., 2021).

With the advance of large-scale language models, multilingual modeling has made impressive progress at performing complex NLP tasks without requiring explicitly translated data. Building on pre-trained language models (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021; Liu et al., 2020), it is now possible to train models that accurately process textual data in multiple languages (Kondratyuk and Straka, 2019) and perform cross-lingual transfer (Pires et al., 2019) using annotated data in one language to process another language.

At the same time, answer generation-based approaches have been shown to be effective for many English QA tasks, including Machine Reading (MR) (Izacard and Grave, 2021; Lewis et al., 2020c), question-based summarization (Iida et al., 2019; Goodwin et al., 2020; Deng et al., 2020), and, most relevant to this work, answer generation for retrieval-based QA (Hsu et al., 2021) — that we refer to as GENQA.

Compared to generative MR models, GENQA approaches are trained to produce complete and expressive sentences that are easier to understand than extracted snippets (Choi et al., 2021). Most importantly, they are trained to generate entire sentences, allowing them to answer both factoid or non-factoid questions, e.g., asking for descriptions, explanation, or procedures.

In this paper, we study and propose a simple technique for open-domain QA in a cross-lingual setting. Following Hsu et al. (2021) (and as illustrated in Figure 1), we work with a pipeline made of 3 main modules. First, a document retriever that retrieves relevant documents given a question; second, an answer sentence selection (AS2) model (Garg et al., 2020; Vu and Moschitti, 2021) that ranks the sentences from the retrieved documents based on how likely they are to include the answer; and third, a generative model that generates a full sentence to answer the question given the sentence candidates.

Our contribution focuses on the generative model. We introduce CROSSGENQA. CROSSGENQA can generate full-sentence answers using sentence candidates written in multiple languages

* Work conducted during internship at Amazon Alexa.
† Work conducted while employed at Amazon Alexa.

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 337–353
November 20–23, 2022. ©2022 Association for Computational Linguistics

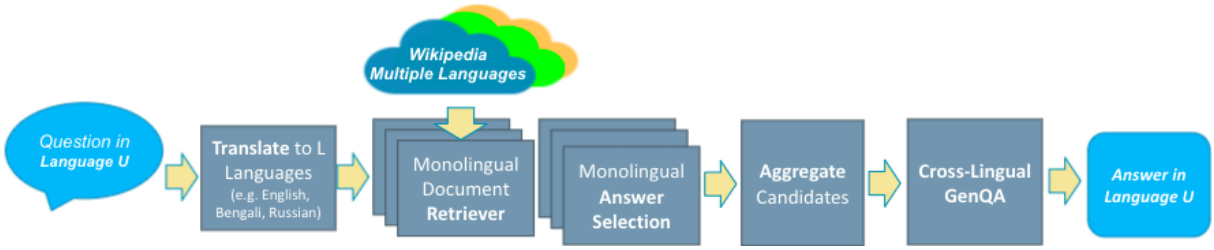

Figure 1: Illustration of our proposed Cross-Lingual, Retrieval-based GENQA pipeline.

including languages different from the question and English.

Given the scarcity of annotated corpora for GENQA, especially in languages different from English, we introduce the GEN-TYDIQA dataset. GEN-TYDIQA is an extension of TyDiQA, a dataset for typologically diverse languages in which questions are answered with passages and short spans extracted from Wikipedia (Clark et al., 2020). Our GEN-TYDIQA includes human-generated, fluent, self-contained answers in Arabic, Bengali, English, Russian and Japanese, making it a valuable resource for evaluating multilingual generative QA systems. We found human-generated answers to be essential in evaluating GENQA: compared to the standard approach of providing reference documents, they dramatically speed-up annotations and improve inter-annotator agreement.

Our evaluation shows that our CROSSGENQA system outperforms AS2 ranking models, and matches or exceeds similar monolingual pipelines.

In summary, our contribution is three-fold:

(i) We introduce GEN-TYDIQA[1], an evaluation dataset that contains natural-sounding answers in Arabic, Bengali, English, Russian and Japanese, to foster the development of multilingual GENQA systems.

(ii) We confirm and extend the results of Hsu et al. (2021) by showing that monolingual generative QA (MONOGENQA) outperforms extractive QA systems in Arabic, Bengali, English and Russian.

(iii) We demonstrate that CROSSGENQA outperforms all our QA systems for Arabic, Russian, and Japanese, answering questions using information from multiple languages.

[1]We make GEN-TYDIQA available at the following URL: s3://alexa-wqa-public/datasets/cross-genqa/

2 Related Work

Multilingual Datasets for QA Researchers have introduced several datasets for QA in multiple languages. Unlike our GEN-TYDIQA, to the best of our knowledge, they are designed exclusively for extractive QA. Artetxe et al. (2019) extended the English machine reading SQuAD dataset (Rajpurkar et al., 2016) by translating the test set to 11 languages. Similarly, Lewis et al. (2020a) collected new question and answer pairs for 7 languages following the SQuAD format. Recently, Longpre et al. (2020) released MKQA, which includes question and answer pairs (predominantly Yes/No answers and entities) for 26 languages. Clark et al. (2020) released TyDiQA, a dataset for extractive QA in 11 typologically diverse languages. Riabi et al. (2020) and Shakeri et al. (2021) have explored the use of techniques to synthetically generate data for extractive question answering using cross-lingual transfer.

Generating Fluent Answers for QA The Generation of fluent and complete-sentence answers is still in its infancy, as most generative models for QA are used for extractive QA (e.g., (Guu et al., 2020; Lewis et al., 2020b; Asai et al., 2021a,b). Approaches to ensure response fluency have been explored in the context of dialogue systems (Baheti et al., 2020; Ni et al., 2021), but remain nevertheless understudied in the context of QA. Providing natural sounding answers is a task of particular interest to provide a better experience for users of voice assistants. One resource for this task is the MS-MARCO dataset (Nguyen et al., 2016). It includes 182,669 question and answer pairs with human-written well-formed answers. However, it only contains samples in English.

Our GEN-TYDIQA extends TyDiQA (Clark et al., 2020) adding natural human-generated answers for Arabic, Bengali, English, Japanese, and Russian. To the best of our knowledge, it is the first

work that provides well-formed, natural-sounding answers for non-English languages.

Multilingual Extractive QA Designing QA models for languages different from English is challenging due to the limited number of resources and the limited size of those datasets. For this reason, many studies leverage transfer learning across languages by designing systems that can make use of annotated data in one language to model another language. For instance, Clark et al. (2020) showed that concatenating the training data from multiple languages improves the performance of a model on all the target languages for extractive QA. In the Open-Retrieval QA setting, multilingual modeling can be used to answer questions in one language using information retrieved from other languages. Da San Martino et al. (2017) showed how cross-language tree kernels can be used to rank English answer candidates for Arabic questions. Montero et al. (2020) designed a cross-lingual question similarity technique to map a question in one language to a question in English for which an answer has already been found. Asai et al. (2021a) showed that extracting relevant passages from English Wikipedia can deliver better answers than relying only on the Wikipedia corpora of the question language. Vu and Moschitti (2021) showed how machine translated question-answer pairs can be used to train a multilingual QA model; in their study, they leveraged English data to train an English and German AS2 model.

Finally, Asai et al. (2021c) introduced CORA and reached state-of-the-art performance on open-retrieval span-prediction question answering across 26 languages. While related to our endeavor, it is significantly different in several key aspects. First, unlike CROSSGENQA, CORA does not produce full, complete sentences; rather, it predicts spans of text that might contain a factoid answer. Second, it mainly relies on sentence candidates that are written in English and in the question language; by contrast, in our work we choose to translate the questions into a variety of languages, allowing us to use monolingual retrieval pipelines to retrieve candidate sentences in diverse languages. We show that this form of cross-lingual GENQA outperforms monolingual GENQA in a majority of the languages studied.

Answer Sentence Selection (AS2) The AS2 task originated in the TREC QA Track (Voorhees, 2001); more recently, it was revived by Wang et al. (2007). Neural AS2 models have also been explored (Wang and Jiang, 2017; Garg et al., 2020). AS2 models receive as input a question and a (potentially large) set of candidate answers; they are trained to estimate, for each candidate, its likelihood to be a correct answer for the given question.

Several approaches for monolingual AS2 have been proposed in recent years. Severyn and Moschitti (2015) used CNNs to learn and score question and answer representations, while others proposed alignment networks (Shen et al., 2017; Tran et al., 2018; Tay et al., 2018). Compare-and-aggregate architectures have also been extensively studied (Wang and Jiang, 2017; Bian et al., 2017; Yoon et al., 2019). Tayyar Madabushi et al. (2018) exploited fine-grained question classification to further improve answer selection. Garg et al. (2020) achieved state-of-the-art results by fine-tuning transformer-based models on a large QA dataset first, and then adapting to smaller AS2 dataset. Matsubara et al. (2020) showed how, similar in spirit to GENQA, multiple heterogeneous systems for AS2 can be be combined to improve a question answer pipeline.

3 The GEN-TYDIQA Dataset

To more efficiently evaluate our multilingual generative pipeline (lower cost and higher speed), we built GEN-TYDIQA, an evaluation dataset for answer-generation-based QA in Arabic, Bengali, English, Japanese, and Russian. This extends the TyDiQA (Clark et al., 2020) dataset.

TyDiQA is a QA dataset that includes questions for 11 typologically diverse languages. Each entry is composed of a human-generated question and a single Wikipedia document providing relevant information. For a large subset of its questions, TyDiQA also contains a human-annotated passage extracted from the Wikipedia document, as well as a short span of text that answers the question. We extend the TyDiQA validation set[2] by collecting human-generated answers based on the provided questions and passages using Amazon Mechanical Turk[3] (cf. Appendix C.1 for hiring criteria and rewards). Collecting human-generated answers is crucial for properly evaluating GENQA models, as we will show in section 5.4. We use a two-stage data collection process:

[2]The TyDiQA test set is not publicly available.
[3]https://requester.mturk.com

(EN) Question: What do pallid sturgeons eat? **TyDiQA Span**: – **GEN-TYDIQA Answer**: Pallid sturgeons eat various species of insects and fish depending on the seasons.
(RU) Question: Когда закончилась Английская революция? *When did the English Revolution end?* **TyDiQA Span**: 1645 **GEN-TYDIQA Answer**: Английская революция, известная также как Английская гражданская вой закончилась в 1645, когда Кромвель создал «Армию нового образца», одержавшую решающую победу в сражении при Нэйсби *The English Revolution, also known as the English Civil War; ended in 1645, when Cromwell created the "Army of the new model", which won a decisive victory at the Battle of Naysby.*
(JA) Question: ストーンズリバーの戦いによる戦死者は何人 *How many were the deaths from the Battle of Stones River?* **TyDiQA Span**: 23,515名 *23,515 people* **GEN-TYDIQA Answer**: ストーンズリバーの戦いで23,515人が川で殺されました。 *23,515 people were killed in the river in the Battle of Stones River.*

Table 1: GEN-TYDIQA question and answer samples.

(1) Answer Generation We show each turker a question and its corresponding passage, and ask them to write an answer that meets the following three properties: (*i*) The answer must be **factually correct and aligned** with the information provided in the passage. If a passage is not sufficient to answer a question, turkers will respond "no answer". (*ii*) The answer must be a **complete and grammatically correct** sentence, or at most a few sentences. (*iii*) The answer should be **self-contained**; that is, it should be understandable without reading the question or the passage. Based on this condition, "yes" or "no" are not acceptable answers.

(2) Answer Validation We show each question alongside its corresponding passage and the human-generated answer from Step (1) to five turkers. We ask them to label if the collected answer meets the three properties listed above: correctness, completeness, and self-containedness. We aggregate labels and keep only answers that received at least 3/5 positive judgements for each property. Table 1 contains some examples of the data collected.

Data Statistics We report the number of GEN-TYDIQA collected human-generated natural answers in table 2, and our coverage of the TyDiQA dataset. We do not reach 100% coverage due to our highly selective validation stage: we only accept answers that receive 3/5 votes for each property, a process that guarantees a high-quality dataset.

Lang. (iso)	#Answers	Avg. Length (utf-8)	%TyDiQA
Arabic (AR)	859	152.5	75.7
Bengali (BN)	89	177.2	63.6
English (EN)	593	64.0	79.5
Japanese (JA)	550	112.0	62.1
Russian (RU)	595	277.9	52.6

Table 2: Statistics on GEN-TYDIQA Answers

4 Multilingual GenQA Systems

Our goal is to build a QA system that, given a question in a target language, retrieves the top-k most relevant passages from text sources in multiple languages, and generates an answer in the target language from these passages (even if the passages are in a different language from the question).

4.1 Task Definition and System Architecture

We first describe the AS2 and GENQA tasks in a language-independent monolingual setting, and then generalize to the cross-lingual setting.

In the monolingual setting for a language L_i, an AS2 system takes as input a question q and a possibly large set of candidate answers C_{L_i} (e.g. all sentences from Wikipedia in the language L_i), ranks each candidate answer given q, and returns the top-ranking candidate $c_m \in C_{L_i}$. A GENQA system uses the top k AS2-ranked answers in C_{L_i} to synthesize a machine-generated answer g in language L_i.

The cross-lingual GENQA task extends this setup as follows: Consider a set of languages $\{L_1, \ldots, L_r\}$. Given a question q in language L_i, let $M = \cup_{j=1}^{r} C_{L_j}$ be the set of relevant candidate sentence answers for q in any language. A cross-lingual GENQA system uses the top k ranked answers in M — regardless of language — to generate an answer g in L_i.

Our architecture, illustrated in Figure 1, consists of the following components: (*i*) question translation[4] from L_i to produce queries q_{L_j} in each language L_j, (*ii*) a document retriever for each L_j to get C_{L_j}, (*iii*) a monolingual AS2 model for each language, which sorts the candidates in C_{L_j} in terms of probability to be correct given q_{L_j}, where C_{L_j} is created by splitting the retrieved documents into sentences, (*iv*) an aggregator component, which builds a multilingual candidate set M using the top k candidates for each language, and

[4]We used Amazon's AWS Translate service, https://aws.amazon.com/translate/service. We validate the quality of AWS Translate on the languages we study in the Appendix section A.3.

(v) a cross-lingual answer generation model, which generates g from M.

We now present in more details each component of our system.

4.2 Multilingual Passage Retrieval

To obtain candidates for our multilingual pipeline, we used Wikipedia snapshots collected in May 2021. We processed each snapshot using WikiExtractor (Attardi, 2015), and create monolingual indices using PyTerrier (Macdonald and Tonellotto, 2020). During retrieval, we first translate queries in each language using AWS Translate. We validate the good quality of this system for all our languages in table 9 in the Appendix. We then use BM25 (Robertson et al., 1995) to score documents. We choose BM25 because, as shown by Thakur et al. (2021), it is competitive with DPR-based models (Karpukhin et al., 2020) and it outperforms DPR across a great diversity of domains.

Evaluation We evaluate the different retrievers independently: for each question, we compare the exact match of the title of the retrieved document with the gold document's title provided by TyDiQA. We compute the Hit@N at the document level, i.e., the percentage of questions having a correct document in the top-N predicted documents. In our experiments, we retrieve the top-100 documents from Wikipedia to feed them to the AS2 model.

4.3 AS2 models for different languages

We build AS2 models by fine-tuning the multilingual masked-language model XLM-R (Conneau et al., 2020) into multiple languages, using question/sentence pairs, which we created with the TyDiQA dataset. We followed the procedure by Garg et al. (2020) performed on the NQ dataset (Kwiatkowski et al., 2019) to build the ASNQ dataset for English. For each $\langle$question, Wikipedia document, span$\rangle$ triplet from the TyDiQA dataset, we use the span to identify positive and negative sentence candidates in the Wikipedia document. We first segment each document at the sentence level using the `spacy` library[5]. We define positive examples to be the sentences that contain the span provided by the TyDiQA dataset, and negative examples to be all other sentences from the same Wikipedia document. We report statistics on AS2-TyDiQA in the Appendix in table 11. For more details, we refer the reader to Garg et al. (2020).

Model We fine-tune XLM-R extended with a binary classification layer on the AS2-TyDiQA dataset described above. At test time, we rank the candidates using the model output probability. Preliminary experiments confirmed the results of Clark et al. (2020) regarding machine reading models on TyDiQA : the best performance is obtained when concatenating the datasets from all languages.

4.4 Multilingual Answer Generation Models

We extended the work of Hsu et al. (2021) on monolingual GENQA modeling. For each question, this model takes the top-5 candidates ranked by AS2 as input. For CROSS-LINGUAL GENQA, we build a set of multiligual candidates M with two methods: (i) TOP 2 / LANG., which selects the top 2 candidates for each language and concatenates them (in total $2 \times 5 = 10$); and (ii) TOP 10, which selects the 10 candidates associated with the highest scores regardless of their language.

Model We used the pre-trained multilingual T5 language model (MT5) by Xue et al. (2021). This is an encoder-decoder transformer-based model (Vaswani et al., 2017) pre-trained with a span-masking objective on a large amount of web-based data from 101 languages (we use the base version). We fine-tuned MT5 following (Hsu et al., 2021): for each sample, we give the model the question concatenated with the candidates M as input and a natural answer as the generated output. GENQA models are trained on MS-MARCO (Nguyen et al., 2016)[6], which includes 182,669 examples of $\langle$question, 10 candidate passages, natural answer$\rangle$ instances in English. When the language of the question (and answer) is not English or when we use candidates in multiple languages, we translate the training samples with Amazon's AWS Translate service and fine-tune the model on the translated data. For instance, to design a GENQA model answering questions in Arabic using input passages in Arabic, English, and Bengali, we fine-tune the model with questions and gold standard answers translated from English to Arabic, and input candidates in English, Arabic, and Bengali, where the latter two are translated from the MS-MARCO English passages.

[5] https://spacy.io/

[6] Using the train split of the NLGEN(v2.1) version.

Evaluation As pointed out by Chen et al. (2019), automatically evaluating generation-based QA systems is challenging. We experimented with BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004), two standard metrics traditionally used for evaluating generation-based systems, but found that they do not correlate with human judgment. For completeness, we report them in the Appendix D.2 along with a detailed comparison with human judgment. Thus, we rely on human evaluation through Amazon Mechanical Turk[7]: we ask three turkers to vote on whether the generated answer is correct, and report the $\frac{\sum PositiveVotes}{\sum TotalVotes}$ as system Accuracy.

5 Experiments

Multilinguality and the different components of our system pipeline raise interesting research questions. Our experimental setup is defined by the combinations of our target set of languages with respect to questions, candidates, and answers. We experiment with GENQA in the monolingual (one model per language) and multilingual (one model for several languages) settings, where the question and candidates in the same language are used to generate an answer. Then we experiment with a cross-lingual GENQA model that is fed candidates in multiple languages. Despite being an apparent more complex task, we find that in many cases, the cross-lingual model outperform all other settings.

5.1 Setup

We approach multilingual generation-based question answering in three ways:

MONOLINGUAL GENQA (MONOGENQA) The candidate language is the same as the question. For each language (Arabic, Bengali, English, Japanese and Russian), we monolingually fine-tune MT5, and report the performance of each GENQA model on the GEN-TYDIQA dataset (Tab. 5).

Our contribution is to show that this approach, first introduced by Hsu et al. (2021) for English, delivers similar performance for other languages.

MULTILINGUAL GENQA (MULTIGENQA) We train one MT5 for all five languages by concatenating their training and validation sets. This single model can answer questions in multiple languages, but it requires that answer candidates be in the same language as the question. We report the performance of this MULTIGENQA model in table 5.

For this set of experiments, we show that a single multilingual GENQA model can compete with a collection of monolingual models.

Model	CANDIDATES	Accuracy
MONOGENQA	EN	**77.9**
CROSSGENQA	DE	70.5
CROSSGENQA	DE ES FR IT	68.8
CROSSGENQA	AR JA KO	31.4
Clozed-Book	NONE	21.0

Table 3: Impact of the candidate language set on CROSS-LINGUAL GENQA in English on MS-MARCO. The language set is controlled with machine translation.

CROSS-LINGUAL GENQA (CROSSGENQA) We use candidates in multiple languages (Arabic, Bengali, Russian, English, Arabic) to answer a question in a target language. We retrieve and rerank sentence candidates in each language, aggregate candidates across all the languages, and finally generate answers (in the same language as the question). We report the performance on the GEN-TYDIQA dataset (table 5).

These experiments aim to determine whether our generative QA model can make use of information retrieved from multiple languages and outperform the baseline methods.

Manual Evaluation We stress the fact that all the results derived in the following experiments were manually evaluated with Amazon Mechanical Turk. In total, we run 34 tasks (system evaluations), requiring around 60k Hits, for a total manual evaluation of 20k QA pairs (times 3 turkers).

5.2 Feasibility Study

To explore whether a model fed with candidates written in languages different from the question can still capture relevant information to answer the question, we conduct a feasibility study using the MS-MARCO dataset with English as our target language and machine translated candidates.

For each question, we translate the top 5 candidate passages to different languages and provide these translated candidates as input to the model. We experiment with three translation settings: all candidates translated to German (DE); each candidate translated to a random choice of German, Spanish, French or Italian (DE-ES-FR-IT); translated to Arabic, Japanese or Korean (AR-JA-KO). We compare all these CROSS-LINGUAL GENQA models with a Clozed-Book QA Model (Roberts

[7] We describe in C.1 how we choose and reward turkers.

Language	BLEU	ROUGE	Accuracy
MONOLINGUAL GENQA			
AR	24.8 / 17.2	47.6 / 38.8	77.1 / 68.4
BN	27.4 / 21.7	48.6 / 43.0	82.0 / 67.4
EN	31.5 / 23.0	54.4 / 46.4	68.5 / 43.6
JA	24.5 / 19.4	50.2 / 45.0	72.3 / 64.3
RU	10.2 / 6.4	30.2 / 23.4	82.6 / 61.3
MULTILINGUAL GENQA			
AR	24.3 / 17.4	47.9 / 39.0	74.9 / 72.7
BN	27.3 / 23.7	47.8 / 44.9	84.3 / 76.5
EN	30.8 / 21.8	54.5 / 46.2	65.3 / 37.4
JA	23.9 / 19.1	50.0 / 45.5	76.8 / 65.5
RU	10.6 / 6.4	31.0 / 23.2	76.6 / 66.7

Table 4: Performance of our GENQA models fine-tuned on MSMARCO and evaluated on GENTYDIQA using Gold-Passage from TyDiQA/Ranked Candidates from Wikipedia.

et al., 2020) for which no candidates are fed into the model.

Results We report the performance in table 3. All CROSS-LINGUAL GENQA models outperform significantly the Clozed-book approach. This means that even when the candidates are in languages different from the question, the model is able to extract relevant information to answer the question. We observe this even when the candidates are in languages distant from the question language (e.g., Arabic, Japanese, Korean).

5.3 GEN-TYDIQA Experiments

This section reports experiments of the full GENQA pipeline tested on the GEN-TYDIQA dataset with candidates retrieved from Wikipedia. For each question, we retrieve documents with a BM25-based retriever, rank relevant candidates using the AS2 model, and feed them to the GENQA models. We note that we cannot compare the model performance across languages: as pointed out in (Clark et al., 2020) regarding TyDiQA.

MONOGENQA Performance We measure the impact of the retrieval and AS2 errors by computing the ideal GENQA performance, when fed with gold candidates (TyDiQA gold passage). We report the results in table 4. We evaluate the performance of the GENQA models, also comparing it to AS2 models on the GEN-TYDIQA dataset of each language. We report the results in table 5 (cf. MONOGENQA). The first row shows the document retrieval performance in terms of Hit@100 for the different languages considered in our work. We note comparable results among all languages, where Arabic reaches the highest accuracy, 70.7, and Japanese the lowest, 57.0. The latter may be due to the complexity of indexing ideogram-based languages. However, a more direct explanation is the fact that retrieval accuracy strongly depends on the complexity of queries (questions), which varies across languages for GEN-TYDIQA. Similarly to Clark et al. (2020), we find that queries in English and Japanese are more complex to answer compared to other languages.

Model	AR	BN	EN	JA	RU
RETRIEVER (Hit@100 doc.)	70.7	66.3	66.9	57.0	67.8
AS2	68.0	58.0	39.0	70.4	60.8
MONOGENQA	68.4	67.4	**43.6**	64.3	61.3
MULTIGENQA	72.7	**76.5**	37.4	65.5	66.7
CROSSGENQA TOP 10	72.0	25.3	31.0	70.3	**74.3**
CROSSGENQA TOP. 2 / LANG.	**73.2**	18.5	29.3	**71.6**	**74.7**

Table 5: Hit@100 doc. of the retriever and Accuracy of GENQA models on GEN-TYDIQA. All CROSS-GENQA experiments use candidates aggregated from all the languages (AR, BN, EN, JA, RU).

Regarding answering generation results, rows 2 and 3 for English confirm Hsu et al. (2021)'s findings: GENQA outperforms significantly AS2 by 4.6% (43.6 vs. 39.0). We also note a substantial improvement for Bengali (+9.4%, 67.4 to 58.0). In contrast, Arabic and Russian show similar accuracy between GENQA and AS2 models. Finally, AS2 seems rather more accurate than GENQA for Japanese (70.4 vs 64.3). Results reported by Xue et al. (2021) show MT5 to be relatively worse for Japanese than all other languages we consider in many downstream tasks, so the regression seen here might be rooted in similar issues.

MULTIGENQA Performance We compare the performance of the MONOLINGUAL GENQA models (one model per language) to the performance of the MULTILINGUAL GENQA model fine-tuned after concatenating the training datasets from all the languages. We report the performance in table 5 (cf. MULTIGENQA): multilingual fine-tuning improves the performance over monolingual fine-tuning for all languages except English. This shows that models benefit from training on samples from different languages. For Bengali, we observe an improvement of around 9% in accuracy. This result has a strong practical consequence: at test time, we do not need one GENQA model per language, we can rely on a single multilingual model trained on the concatenation of datasets from multiple languages (except for English, where we find that the monolingual model is more accurate). This result generalizes what has been shown for extractive QA (Clark et al., 2020) to the GENQA task.

Model	Candidates	Accuracy
MONOGENQA	EN	57.8
CROSSGENQA	JA	60.3
CROSSGENQA	AR-BN-EN-JA-RU TOP 10	56.9
CROSSGENQA	AR-BN-EN-JA-RU TOP 2 / LANG	**63.8**

Table 6: GENQA scores in English on Japanese-culture-specific questions extracted from TyDiQA. CANDIDATES defines the language set of the input candidates.

CROSSGENQA Performance Our last and most important contribution is in table 5, which reports the performance of a GENQA model trained and evaluated with candidates in multiple languages. This model can answer a user question in one language (e.g., Japanese) by using information retrieved from many languages, e.g., Arabic, Bengali, English, Japanese, and Russian). For Arabic, Japanese, and Russian, we observe that CROSS-LINGUAL GENQA outperforms other approaches by a large margin, e.g., for Russian, 13.8% (74.6-60.8) better than AS2, and an 8% percent improvement over MULTIGENQA.

For Bengali, the model fails at generate good quality answers (CROSSGENQA models reach at best 25.3% in accuracy compared to the 76.9% reached by the MULTIGENQA model). We hypothesize that this is the consequence of a poor translation quality of the question from Bengali to other languages such as English, Arabic, or Japanese, which leads to poor candidate retrieval and selection, ultimately resulting in inaccurate generation.

Finally, we compare the two candidate aggregation strategies used for CROSS-LINGUAL GENQA: TOP 2 / LANG. and TOP 10 (see section 4.4). We observe that the aggregation strategy impacts moderately the downstream performance. For English, Arabic, Japanese and Russian the gap between the two methods is at most 2 points in accuracy. We leave the refinement of candidate selection in the multilingual setting for future work.

5.4 Analysis

Examples Table 7 shows the output of AS2, MULTILINGUAL GENQA, and CROSS-LINGUAL GENQA models to questions in Russian and Bengali. For Bengali, the GENQA models provide a correct and fluent answer while the AS2 model does not. For Russian, only the CROSS-LINGUAL GENQA model is able to answer correctly the question. This because AS2 does not rank the right information in the top k, while CROSS-LINGUAL GENQA can find the right information in another language in the multi-language candidate set.

Question: জাস্টিন ড্রিউ বিবারের জন্ম কবে হয় ?
When was Justin Drew Bieber born?
AS2 Prediction:
ম্যাথু লরেন্স হেইডেন, এএম (; জন্ম: ২৯ অক্টোবর, ১৯৭১) কুইন্সল্যান্ডের কিঙ্গারয় এলাকায় জন্মগ্রহণকারী সাবেক অস্ট্রেলীয় ক্রিকেটার হিসেবে সমগ্র ক্রিকেট বিশ্বে পরিচিত ব্যক্তিত্বরেন
Matthew Lawrence Hayden, AM (born October 29, 1971) is a former Australian cricketer born in Kingroy, Queensland.
MULTIGENQA Prediction:
জাস্টিন ড্রু বিবার ১৯৯৪ সালের ১ মার্চ জন্মগ্রহণ করেন।
Justin Drew Bieber was born on March 1, 1994.
CROSSGENQA Prediction
জাস্টিন ড্রু বিবার ১৯৯৪ সালের ১ মার্চ জন্মগ্রহণ করেন।
Justin Drew Bieber was born on March 1, 1994.

Question: トゥールのグレゴリウスはいつ生まれた？
When was Gregory of Tours born?
AS2 Prediction: グレゴリウス14世（Gregorius XIV,1535年2月11日 - 1591年10月16日）はローマ教皇（在位：1590年 - 1591年）。 *Pope Gregory XIV (February 11, 1535 – October 16, 1591) is the Pope of Rome (reigned: 1590 – 1591).*
MULTIGENQA Prediction:トゥールのグレゴリウスは、1535年2月11日に生まれた。 *Gregory of Tours was born on February 11, 1535.*
CROSSGENQA Predictionトゥールのグレゴリウスは538年頃11月30日に生まれた。 *Gregory of Tours was born on November 30, 538.*

Table 7: Example of predicted answers to questions in Bengali and Japanese. **Blue** indicates correct predictions while **Red** incorrect ones. Translations are intended for the reader and are not part of the predictions.

Error Propagation We observe (table 4) that the GENQA models are highly impacted by the retriever and AS2 quality. For example, English GENQA performance drops of 27.9 (65.3-37.4) points in Accuracy. This suggests that large improvement could be achieved by improving the document retriever and/or AS2 modules.

Culture-Specific Questions in English One striking result across our experiments is the lower performance of CROSS-LINGUAL GENQA model than GENQA model on English. We hypothesize that English questions from the GEN-TYDIQA dataset are more easily answered using information retrieved from English compared to other languages because those questions are centered on

Eval mode	Strong agreement	Perfect agreement	Fleiss' kappa
No Reference	55.00 %	16.43 %	0.1387
With Reference	85.36 %	55.25 %	0.5071

Table 8: Comparison between providing a reference answer and not for evaluating MONOGENQA predictions (EN). Providing a reference increases agreement.

cultures specific to English-speaking countries.

To verify our hypothesis, we re-run the same set of experiments, using culture-specific Japanese questions rather than English queries. To do so, we (i) took the Japanese questions set from GEN-TYDIQA, (ii) manually translated it in English, (iii) manually select 116 questions that are centered on Japanese culture, and (iv) run the same GENQA pipeline on those questions. The results reported in table 6 show that CROSSGENQA outperforms MONOGENQA, suggesting that the former improves also the English setting if the question set is culturally not centered on English, i.e., it requires answers that cannot be found in English.

Use of Reference Answer in Model Evaluation We found the use of human-generated reference answers to be crucial to ensure a consisted annotation of each model. A comparison between annotation with and without reference answer is provided in table 8. When using a reference, we found annotators to be dramatically more consistent, achieving a Fleiss' Kappa (Fleiss, 1971) of 0.5017; when providing no reference answer, the inter-annotation agreement dropped to 0.1387. This trend is reflected in the number of questions with strong (4+ annotators agree) and perfect agreement.

6 Limits

Our system requires translating the questions. We also use the standard BM25 approach. Even though it was shown to be more robust compared to dense retriever (Thakur et al., 2021; Rosa et al., 2022), using a cross-lingual retriever (Li et al., 2021) could improve performance and save the cost of translating the question. This has been explored by Asai et al. (2021c) but their retriever mainly retrieves passages in English and the question language which may lead to English-centric answers. Another limit is the fact that our system is not designed to handle questions that are not answerable. In the future, we may want to integrate a no-answer setting to avoid unwanted answer.

7 Conclusion

We study retrieval-based Question Answering systems using answer generation in a multilingual context. We proposed (i) GEN-TYDIQA, a new multilingual QA dataset that includes natural and complete answers for Arabic, Bengali, English, Japanese, and Russian; based on this dataset (ii) the first multilingual and cross-lingual GENQA retrieval-based systems. The latter can accurately answer questions in one language using information from multiple languages, outperforming answer sentence selection baseline for all languages and monolingual pipeline for Arabic, Russian, and Japanese.

References

Abdulwhab Alkharashi and Joemon Jose. 2018. Vandalism on collaborative web communities: An exploration of editorial behaviour in wikipedia. In *Proceedings of the 5th Spanish Conference on Information Retrieval*, pages 1–4.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021a. 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.

Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. 2021b. One question answering model for many languages with cross-lingual dense passage retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pages 7547–7560. Curran Associates, Inc.

Akari Asai, Xinyan Yu, Jungo Kasai, and Hanna Hajishirzi. 2021c. One question answering model for many languages with cross-lingual dense passage retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pages 7547–7560. Curran Associates, Inc.

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

Ashutosh Baheti, Alan Ritter, and Kevin Small. 2020. Fluent response generation for conversational question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 191–207, Online. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with

dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1987–1990, New York, NY, USA. Association for Computing Machinery.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

A. Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *ACL*.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

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, Edouard 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, Online. Association for Computational Linguistics.

Giovanni Da San Martino, Salvatore Romeo, Alberto Barroón-Cedeño, Shafiq Joty, Lluís Maàrquez, Alessandro Moschitti, and Preslav Nakov. 2017. Cross-language question re-ranking. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1145–1148, New York, NY, USA. Association for Computing Machinery.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7651–7658. AAAI Press.

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.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

William Falcon and Kyunghyun Cho. 2020. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7780–7788.

Travis Goodwin, Max Savery, and Dina Demner-Fushman. 2020. Towards Zero-Shot Conditional Summarization with Adaptive Multi-Task Fine-Tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3215–3226, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjan Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.

Chao-Chun Hsu, Eric Lind, Luca Soldaini, and Alessandro Moschitti. 2021. Answer generation for retrieval-based question answering systems. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4276–4282, Online. Association for Computational Linguistics.

Ryu Iida, Canasai Kruengkrai, Ryo Ishida, Kentaro Torisawa, Jong-Hoon Oh, and Julien Kloetzer. 2019. Exploiting background knowledge in compact answer generation for why-questions. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 142–151. AAAI Press.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

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

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

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. *ArXiv*, abs/2005.11401.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020c. Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv preprint arXiv:2005.11401*.

Yulong Li, Martin Franz, Md Arafat Sultan, Bhavani Iyer, Young-Suk Lee, and Avirup Sil. 2021. Learning cross-lingual ir from an english retriever. *ArXiv*, abs/2112.08185.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Shayne Longpre, Yi Lu, and Joachim Daiber. 2020. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering.

Craig Macdonald and Nicola Tonellotto. 2020. Declarative experimentation ininformation retrieval using pyterrier. In *Proceedings of ICTIR 2020*.

Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. 2020. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1577–1580.

Ivan Montero, Shayne Longpre, Ni Lao, Andrew J. Frank, and Christopher DuBois. 2020. Pivot through english: Reliably answering multilingual questions without document retrieval. *CoRR*, abs/2012.14094.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, V. Ananth Krishna Adiga, and E. Cambria. 2021. Recent advances in deep learning based dialogue systems: A systematic survey. *ArXiv*, abs/2105.04387.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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.

Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2020. Synthetic data augmentation for zero-shot cross-lingual question answering. *CoRR*, abs/2010.12643.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Guilherme Moraes Rosa, Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Marzieh Fadaee, Roberto de Alencar Lotufo, and Rodrigo Nogueira. 2022. No parameter left behind: How distillation and model size affect zero-shot retrieval.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 373–382, New York, NY, USA. Association for Computing Machinery.

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.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-cast attention networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, page 2299–2308, New York, NY, USA. Association for Computing Machinery.

Harish Tayyar Madabushi, Mark Lee, and John Barnden. 2018. Integrating question classification and deep learning for improved answer selection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3283–3294, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Quan Hung Tran, Tuan Lai, Gholamreza Haffari, Ingrid Zukerman, Trung Bui, and Hung Bui. 2018. The context-dependent additive recurrent neural net. 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 1274–1283, New Orleans, Louisiana. Association for Computational Linguistics.

Rodolfo Vieira Valentim, Giovanni Comarela, Souneil Park, and Diego Sáez-Trumper. 2021. Tracking knowledge propagation across wikipedia languages. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1):1046–1052.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ellen M. Voorhees. 2001. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378.

Thuy Vu and Alessandro Moschitti. 2021. Multilingual answer sentence reranking via automatically translated data.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection. *CoRR*, abs/1905.12897.

A Discussion

A.1 Machine Translation of the Questions and BM25 Retriever Engines

Our work introduces CROSS-LINGUAL GENQA, a system that can answer questions — with complete sentence answers — in multiple languages using candidates in multiple languages, possibly distinct from the question. They were many possible design choices to achieve such a goal. We chose to rely on automatically translating the questions before retrieving relevant documents in several languages using multiple (monolingual) BM25 retrievers. We could have chosen to use the recently released multilingual Dense passage Retrieval (mDPR) (Asai et al., 2021b). We decided not to for the two following reasons. First, as shown by Thakur et al. (2021), BM25 is a very reasonable design choice for a retriever engine, that outperforms other approaches in many settings (including dense retrievers). Second, as seen in (Asai et al., 2021b), multilingual dense retrievers usually retrieve passages in the same language as the question or English. This means that mDPR is highly biased toward the English language. In our work, by combining translation and monolingual retrievers, we can control the language set that we use for answer generation. We leave for future work the refinement of mDPR to enable for more diversity in the retrieved passage languages and to integrate it in our pipeline.

A.2 Machine Translation Errors

At test time, our system applies Machine Translation to the question to formulate queries in different languages and retrieve candidates for these languages using the BM25 retrieval engine. To our knowledge this is the best approach to generate queries in different languages, as MT systems are very powerful tools, trained on millions of data points and, thanks to Transformer model, they take the entire question context into account (other cross-query formulations can be applied but they will be probably less accurate and multilingual DPR is an excellent research line but not as much assessed as BM25 as effective and general approach). Clearly MT errors can impact the quality of our candidates. However, if a question is badly translated the retrieved content will be inconsistent with the candidates retrieved for the question in the original language (and also inconsistent with candidates retrieved using questions translated in other languages). Our joint modeling through large generation-based Transformers can recover from these random errors. For example, for 3 languages out of 5, we show that the Cross-GenQA pipelines that use MT for the question outperform monolingual pipelines (MONOGENQA and MULTIGENQA). This shows that translation errors are recovered by our approach.

	ar	bn	en	ja	ru
ar		**25.9**/16.1	**40.8**/25.5	**26.1**/16.0	**27.3**/17.8
bn	**22.8**/10.7		**32.8**/22.9	**23.5**/16.5	**21.8**/14.7
en	**39.5**/17.9	**32.7**/23.0		**34.2**/22.8	**36.6**/27.1
ja	**21.0**/10.3	**22.6**/16.0	**28.0**/19.4		**21.4**/15.3
ru	**25.9**/13.5	**24.9**/18.1	**37.3**/27.5	**26.4**/20.3	

Table 9: Performance measured with spBLEU of AWS translate compared to a Many-to-Many (M2M) Multilingual Transformer Model (reported in (Goyal et al., 2022)) on the FLORES devtest dataset (Goyal et al., 2022). Cell(i,j) reports the score of AWS/M2M from language i to language j. AWS translate outperforms the M2M model for all language pairs.

A.3 AWS-Translation for Machine Translation

For translating the questions automatically, we use AWS Translate. AWS Translate is a machine translation API that competes and outperforms in some cases other available translation APIs[8]. We compare the performance of a strong baseline on the FLORES dataset in table 9. We find that AWS translate outperforms the baseline for all the language pairs we work with. We leave for future work the study of the impact of different machine translation systems on our CROSS-LINGUAL GENQA models.

B Ethics Statement

B.1 Potential Harms of GENQA

All our GENQA are fine-tuned from a large pre-trained language model, MT5 (Xue et al., 2021). In general, large language models have been shown to have a potential to amplify societal biases (Bender et al., 2021), and might leak information about the datasets they were trained on (Carlini et al., 2021). In particular, the Colossal Cleaned Crawled Corpus (C4) and its multilingual counterpart (MC4) that were used to train MT5 have been shown to

[8] cf. https://aws.amazon.com/blogs/machine-learning/amazon-translate-ranked-as-1-machine-translation-provider-by-intento/

disproportionately under-represent content about minority individuals (Dodge et al., 2021).

In its use as a retrieval-based question answering system, GENQA also can also cause harm due to (*i*) the use of candidate sentences that are extracted from web documents, and (*ii*) model hallucinations that are produced during decoding. In this work, (*i*) is mitigated by only relying on content from Wikipedia, which, while not immune to vandalism (Alkharashi and Jose, 2018), is of much higher quality of unvetted web data. Regarding the risk of model hallucinations, this work does not attempt to directly mitigate any potential issue through modeling; rather, we always show annotators reference answer so that hallucination that result in factually incorrect answers can be properly caught during evaluation.

B.2 GEN-TYDIQA Copyright

Our GEN-TYDIQA dataset is based on the TyDiQA dataset questions (Clark et al., 2020). TyDiQA is released under the Apache 2.0 License which allows modification and redistribution of the derived dataset. Upon acceptance of this paper, we will release GEN-TYDIQA and honor the terms of this license.

GEN-TYDIQA answers were collected using Amazon Mechanical Turk. No geolocation filters or any personal information were used to hire turkers. Additionally, GEN-TYDIQA questions treat scientific or cultural topics that can be answered objectively using Wikipedia. For these reasons, the collected answers cannot be used to identify their authors. Finally, to ensure the complete anonymity of the turkers, we will not release the turkers id along with the collected answers.

B.3 Energy Consumption of Training

All our experiments are based on the MT5 base model. We run all our fine-tuning and evaluation runs using 8 Tesla P100 GPUs[9], which have a peak energy consumption of 300W each. Fine-tuning our CROSS-LINGUAL GENQA models on MS-MARCO (Nguyen et al., 2016) takes about 24 hours.

[9] https://www.nvidia.com/en-us/data-center/tesla-p100/

C Reproducibility

C.1 Mechanical-Turk Settings

In this paper, we rely on Amazon Mechanical Turk for two distinct uses.

On the one hand, we use it to build the GEN-TYDIQA dataset. For data collection, we request 1 turker per question to generate an answer. For the GEN-TYDIQA data validation, we request 5 turkers to select only answers that are correct, aligned with the provided passage, self-contained and complete.

On the other hand, we use Amazon Mechanical Turk to estimate the answer accuracy of our models. To do so, for each question, we provide the GEN-TYDIQA reference and ask 3 turkers to vote on whether the generated answer is correct or not.

For those two uses, we use the following Amazon Mechanical Turk filters to hire turkers.

- We hire turkers that received at least a 95% HIT[10] approval rate.
- We request turkers that have performed at least 500 approved HITs.
- When possible, we use the "*master turker*" filter[11] provided by Amazon Mechanical Turk. We find that this filter can only be used for English. For other languages, this filter leads to a too-small turker pool making it unusable in practice.

On Mechanical turk, the reward unit for workers is the HIT. In our case, a HIT is the annotation/validation of a single question. We make sure that each turker is paid at least an average of 15 USD/hour. To estimate the fair HIT reward, we first run each step with 100 samples ourselves in order to estimate the average time required per task. For data collection, we set the HIT reward to 0.50 USD based on an estimation of 0.5 HIT/min. For data validation, we set it to 0.15 USD based on an estimation of 1.6 HIT/min. For model evaluation,

[10] A HIT, as defined in Amazon Mechanical Turk, is a *Human Intelligent Task*. In our case, a HIT consists in generating, validating, or accepting an answer to a single question.

[11] As stated on the Amazon Mechanical Turk website, "Amazon Mechanical Turk has built technology which analyzes Worker performance, identifies high performing Workers, and monitors their performance over time. Workers who have demonstrated excellence across a wide range of tasks are awarded the Masters Qualification. Masters must continue to pass our statistical monitoring to retain the Amazon Mechanical Turk Masters Qualification."

Parameter	Value	Bounds
Effective Batch Size	128	[1, 8192]
Optimizer	Adam	-
Learning Rate	5e-4	[1e-6,1e-3]
Gradient Clipping value	1.0	-
Epochs (best of)	10	[1, 30]
Max Sequence Length Input	524	[1, 1024]
Max Sequence Length Output	100	[1, 1024]

Table 10: Optimization Hyperparameter to fin-tune MT5 for the GENQA task. For each hyper-parameter, we indicate the value used as well as the parameter lower and upper bounds when applicable.

Language	# Candidates	% Positive Candidates
AR	1,163,407 / 100,066	1.30 / 1.46
EN	688,240 / 197,606	0.56 / 0.49
BN	334,522 / 23892	0.76 / 0.74
JA	827,628 / 214,524	0.47 / 0.47
RU	1,910,388 / 245,326	0.34 / 0.48

Table 11: AS2-TyDiQA dataset extracted from the TyDiQA dataset. We report Train/Dev set following the TyDiQA split. We note that each question have at least one positive candidate

we set the HIT reward to 0.10 USD based on an estimation of 2.5 HIT/min.

C.2 Model Optimization

All the GENQA experiments we present in this paper are based on fine-tuning MT5 base (Xue et al., 2021). Models are implemented in PyTorch (Paszke et al., 2019), and leverage `transformers` (Wolf et al., 2020) and `pytorch-lightning` (Falcon and Cho, 2020). For fine-tuning, we concatenate the question and the candidate sentences, input it to the model and train it to generate the answer. Across all our runs, we use the hyperparameters reported in table 10.

D Analysis

D.1 Gold vs. Retrieved Candidates

We report in table 4 the performance of the MONOGENQA and MULTIGENQA models when we feed them gold passages (using TyDiQA passage) and compare them with the performance of the same models fed with the retrieved candidates. We discuss those results in section 5.4.

D.2 Human Evaluation vs. BLEU and ROUGE-L

For comparison with previous and future work, we report the BLEU score (computed with SacreBLEU (Post, 2018)) and the F-score of the ROUGE-L metric (Lin, 2004) along with the human evaluation accuracy in table 14.

LANGUAGE	W. BLEU	W. ROUGE
AR	9.5	24.5
BN	21.2	5.3
EN	11.7	23.5
RU	5.9	16.8

Table 12: Spearman Rank Correlation (%) of human estimated Accuracy with BLEU and the ROUGE-L F score. We run this analysis at the sentence level on the MULTILINGUAL GENQA predictions.

LANGUAGE	W. BLEU	W. ROUGE
AR	30.0	30.0
BN	-50.0	-50.0
EN	40.0	40.0
JA	-90.0	-60.0
RU	-87.2	100.0

Table 13: Spearman Rank Correlation (%) of human estimated Accuracy with the BLEU score and the ROUGE-L F score at the model level across our 5 models (AS2, MONOGENQA, MULTIGENQA, CROSSGENQA (x2))

As seen in previous work discussing the automatic evaluation of QA systems by Chaganty et al. (2018) and Chen et al. (2019), we observe that for many cases, BLEU and ROUGE-L do not correlate with human evaluation. In table 12, we take the predictions of our MULTIGENQA model across all the languages and compute the Spearman rank correlation at the sentence level of the human estimated accuracy with BLEU and ROUGE-L. We find that this correlation is at most 25%. This suggests that those two metrics are not able to discriminate between correct predictions and incorrect ones.

Additionally, we report the Spearman rank correlation between the Accuracy and BLEU or ROUGE across all our 5 models in table 13. We find that neither BLEU nor ROUGE-L correlates strongly with human accuracy across all the languages. This means that those metrics are not able to rank the quality of a model in agreement with human judgment. Those results lead us to focus our analysis and to take our conclusions only on human evaluated accuracy. We leave for future work the development of an automatic evaluation method for multilingual GENQA.

MODEL	QUESTION	CANDIDATES	BLEU	ROUGE	Accuracy
AS2	AR	AR	5.9	20.6	68.0
MONOGENQA	AR	AR	17.2	38.8	68.4
MULTIGENQA	AR	AR	17.4	39.0	72.7
CROSSGENQA	AR	AR-BN-EN-JA-RU TOP 10	15.3	36.5	72.0
CROSSGENQA	AR	AR-BN-EN-JA-RU TOP 2 PER LANG.	14.7	36.3	**73.2**
AS2	BN	BN	3.8	16.6	58.0
MONOGENQA	BN	BN	21.7	43.0	67.4
MULTIGENQA	BN	BN	23.7	44.9	**76.5**
CROSSGENQA	BN	AR-BN-EN-JA-RU TOP 10	35.2	56.5	25.3
CROSSGENQA	BN	AR-BN-EN-JA-RU TOP 2 PER LANG.	33.5	54.8	18.5
AS2	EN	EN	5.6	20.0	39.0
MONOGENQA	EN	EN	23.0	46.4	**43.6**
MULTIGENQA	EN	EN	21.8	46.2	37.4
CROSSGENQA	EN	AR-BN-EN-JA-RU TOP 10	21.0	45.5	31.0
CROSSGENQA	EN	AR-BN-EN-JA-RU TOP 2 PER LANG.	20.2	44.8	29.3
AS2	JA	JA	6.7	22.4	70.4
MONOGENQA	JA	JA	19.4	45.0	64.3
MULTIGENQA	JA	JA	19.1	45.5	65.5
CROSSGENQA	JA	AR-BN-EN-JA-RU TOP 10	17.6	42.2	70.3
CROSSGENQA	JA	AR-BN-EN-JA-RU TOP 2 PER LANG.	16.6	43.0	**71.6**
AS2	RU	RU	7.4	13.3	60.8
MONOGENQA	RU	RU	6.4	23.4	61.3
MULTIGENQA	RU	RU	6.4	23.2	66.7
CROSSGENQA	RU	AR-BN-EN-JA-RU TOP 10	4.2	21.0	**74.3**
CROSSGENQA	RU	AR-BN-EN-JA-RU TOP 2 PER LANG.	5.3	22.8	**74.7**

Table 14: Performance of GENQA models on GEN-TYDIQA based on retrieved and reranked candidates. QUESTION indicates the language of the question and the answer while CANDIDATES indicates the language set of the retrieved candidate sentences.