Question Answer Generation in Bengali: Mitigating the scarcity of QA datasets in a low-resource language

Md Shihab Shahriar Ahmad Al Fayad Chowdhury Md. Amimul Ehsan
Abu Raihan Kamal
Department of Computer Science and Engineering
Islamic University of Technology
{shihabshahriar, ahmadalfayad, amimulehsan, raihan.kamal}@iut-dhaka.edu

Abstract

The scarcity of comprehensive, high-quality Question-Answering (QA) datasets in low-resource languages has greatly limited the progress of research on QA for these languages. This has inspired research on Question-Answer Generation (QAG) which seeks to synthetically generate QA pairs and minimize the human effort required to compile labeled datasets. In this paper, we present the first QAG pipeline for the Bengali language, which consists of an answer span extraction model, a question generation model, and roundtrip consistency filtering to discard inconsistent QA pairs. To train our QAG pipeline, we translate SQuAD1.1 and SQuAD2.0 using the state-of-the-art NLLB machine translation model and accurately mark the answer spans using a novel embedding-based answer alignment algorithm to construct two Bengali QA datasets that we show are superior to the only two existing machine-translated datasets in terms of quality and quantity. We use our QAG pipeline to generate more than 170,000 QA pairs to build BanglaQA, a synthetic QA dataset from 16,000 Bengali news articles spanning 5 different news categories. We demonstrate the quality of BanglaQA by human evaluation on a variety of metrics. The best-performing model among several baselines on our dataset achieves an F1 score of 86.14 falling behind human performance of 95.72 F1. Our codebase and curated datasets are publicly available at https://github.com/shihabshahriar16/BengaliQAG.git.

1 Introduction

Pretrained language models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2020) etc. have shown performance comparable to human agents on the Natural Language Processing (NLP) task of Question Answering (QA). However, this performance has been recorded in the case of well-resourced languages that have extensive, publicly-available QA datasets to satisfy the incredible training requirements of these models. The scenario for low-resource languages is concerning as they have seen considerably less progress than their high-resource counterparts. This is primarily due to a scarcity of labeled data, which can be attributed to the massive amount of human effort and time required to create QA datasets. With a particular focus on the Bengali language for this work, we found mention of only two relevant Bengali QA datasets, namely Bengali-SQuAD (Tahsin Mayeesha et al., 2021) and SQuAD_Bn (Bhattacharjee et al., 2022a).

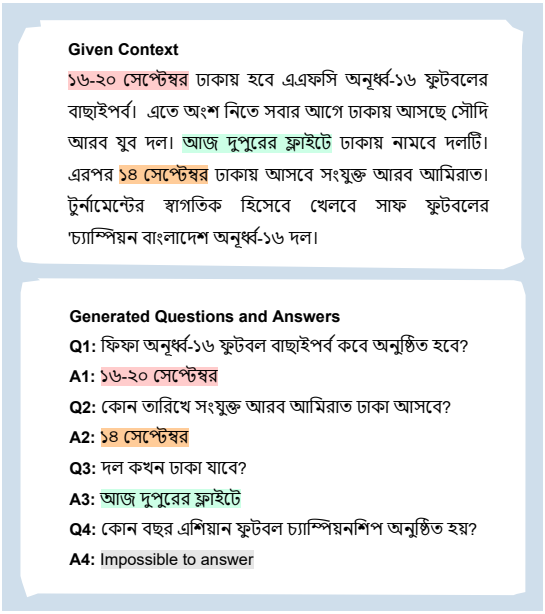

Figure 1: An example of synthetically generated QA pairs from a given context using our QAG pipeline. For more examples see Appendix A.

One approach that has been explored is machine-translated datasets from a high-resource source language to a low-resource target language. Both Bengali-SQuAD and SQuAD_Bn are examples of this method. Bengali-SQuAD is a Google

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 430–441
November 1–4, 2023.

Cloud Translation[1] of SQuAD2.0 (Rajpurkar et al., 2018) and SQuAD_Bn augments their translation of SQuAD2.0 with the Bengali subset of the popular TyDiQA (Clark et al., 2020) dataset. However, we note that such datasets often present data quality issues. Despite being economical in terms of time and cost, a major issue in this translation-based approach is that the translated answer does not represent the correct answer span in the context which results in discarding data samples or degrading the quality of the datasets.

Question Answer Generation (QAG) is an alternative approach proposed to tackle the problem of a scarcity of QA datasets. QAG is the task of generating QA pairs consistent with the information in a provided context and has garnered great interest from the NLP communities in both industry and academia (Zhao et al., 2018). Earlier QAG models employed regular Recurrent Neural Networks (RNNs) and their attention-augmented variants. However, the inability of RNNs to capture semantic information in long sequences has pushed work towards the use of transformer-based architectures (Vaswani et al., 2017). Alberti et al. (2019) and Chan and Fan (2019) have proven the effectiveness of these models in generating synthetic QA data that can supplement existing data to train more robust and accurate QA models.

In this work, we present a Bengali QAG pipeline that can generate synthetic datasets to mitigate the dearth of comprehensive QA datasets in Bengali. Most efforts in curating QA datasets in Bengali have so far been limited to translations of English datasets or have involved a laborious human annotation process. Our work is the first of its kind in the Bengali language to explore this area of research. The QAG pipeline consists of an answer span extraction model, a question generation model, and a roundtrip consistency filtering mechanism to produce QA pairs. To train the pipeline, we translate SQuAD1.1 (Rajpurkar et al., 2016) and SQuAD2.0 and generate two new translated QA datasets, namely SQuADBangla1.1 and SQuADBangla2.0. Witnessing its tremendous capabilities in machine translation as demonstrated by a 44% BLEU score improvement over the previous state-of-the-art model, we employ Meta AIs NLLB (NLLB Team et al., 2022) model to translate the SQuAD datasets in Bengali. We then apply a novel embedding-based answer alignment algorithm to accurately identify answer spans in the translated contexts, since we identified this as an issue in existing datasets.

Further, to demonstrate the effectiveness of our QAG pipeline, we introduce BanglaQA, the first synthetic Bengali QA dataset, comprising more than 170,000 QA pairs. We use the BARD dataset (Tanvir Alam and Mofijul Islam, 2018), a collection of scraped Bengali news articles spanning five categories, and generate both answerable and unanswerable QA pairs, following SQuAD2.0. We present an assessment of the quality of this dataset via human evaluation on five criteria and establish baseline performance scores of three different models on it.

The contributions of this paper can be summarized as:

- We present the first Bengali QAG pipeline to produce synthetic QA datasets.
- We introduce two new translated QA datasets, SQuADBangla1.1 and SQuADBangla2.0 which we show to be superior to existing Bengali QA datasets in terms of quality and quantity.
- We release BanglaQA, the first Bengali synthetic QA dataset, which also validates the effectiveness of our QAG pipeline.

2 Related Works

Explorations in the field of QA in the Bengali language began with building factoid-based QA systems. Banerjee et al. (2014) attempted to build the first Bengali factoid-based QA system, BFQA, which was an information retrieval system that classified questions, retrieved relevant sentences, ranked them, and extracted correct answers. Hoque et al. (2015) built BQAS, a bilingual question-answering system that could generate and answer factoid-based questions from English and Bengali documents. Islam and Nurul Huda (2019) also implemented a similar question-answering system but based it entirely on time-related questions. However, none of the work before Tahsin Mayeesha et al. (2021) employed deep learning techniques on SQuAD-like reading comprehension datasets in Bengali.

Question Generation (QG) is concerned with two questions - what to ask and how to ask. The first part, content selection, was tackled in the past

[1] https://cloud.google.com/translate

by applying semantic or syntactic parsing of text sequences to obtain intermediate symbolic representations. The second part involves question construction which takes these representations and converts them to natural language questions either in a transformation-based or a template-based approach. (Pan et al., 2019)

The current deep learning frameworks follow the sequence-to-sequence approach and employ transformer-based architectures to learn the content selection via the encoder and the question construction via the decoder. These QG models differ only in certain factors like answer encoding (for answer-aware question generation), question word generation, and paragraph-level contexts. Recent works have solved the problem of answer encoding by either treating the answers position as an input feature (Zhao et al., 2018), by encoding the answer with a separate RNN (Duan et al., 2017; Kim et al., 2018), or a mixture of both via transformer-based architectures (Lee et al., 2020; Alberti et al., 2019; Chan and Fan, 2019).

BERT models have been used effectively by Alberti et al. (2019) to generate synthetic QA pairs. The authors use three separate BERT models for the auxiliary tasks of answer extraction, question generation, and question answering. Coupled with roundtrip consistency which ensures that noisy context-question-answer tuples are removed, they show that QA models that are fully pretrained on QA datasets as well as synthetic QA pairs outperform those that are only fine-tuned on the QA datasets. Some works have also looked into different forms of encoding the answer as an input feature. Chan and Fan (2019) show that their BERT-HLSQG model, which highlights the answer span within the context with special tokens can outperform previously suggested RNN and LSTM-based models.

Lewis et al. (2021) uses a pipeline consisting of four components to generate QA pairs. For passage selection, the authors fine-tune a RoBERTa model on known QA datasets to identify information-rich contexts. Then a BERT-based model or an NER-based approach is used to extract plausible answer text spans. Subsequently, a BART model conditioned on the answer-annotated passage produces relevant questions. Finally, an existing QA model evaluates the question-answer compatibility to omit contradictory pairs which they coin as global filtering.

Drawing inspiration from these works in the English language, we leverage transformer-based architectures pretrained on Bengali corpora and build a QAG pipeline to overcome the problem of QA dataset scarcity. We demonstrate in later sections that the resulting synthetic QA data is comparable to human-annotated QA datasets and can be used to supplement existing QA datasets.

3 Methodology

In this section, we describe our process of generating QA pairs in Bengali. In section 3.1, we describe the process of translating the questions, answers, and contexts of a QA dataset separately and then aligning and correcting the answer in the translated dataset. In section 3.2, we provide an overview of our QAG pipeline which consists of an answer span extraction model, a QG model, and a QA model for ensuring round-trip consistency.

3.1 Translate QA dataset from a high-resource language

We denote the context as C, the question as Q and the answer as A. Before translating the context we tokenize the context into individual sentences $C = [c_1, c_2, \cdots, c_n]$ using the sentence tokenizer from the Natural Language Toolkit (NLTK) library (Bird et al., 2009). Both SQuAD1.1 and SQuAD2.0 contain the answer start position for every answer. We use this position to identify the context sentence which has the answer, denoting it as c_{ans}. Using the NLLB model, we independently translate C, Q and A and denote the translated context as C', the translated question as Q', and the translated answer as A'. We map c_{ans} to its corresponding sentence in C' and call it c'_{ans}.

3.2 Answer span alignment and correction

Following translation, we find the correct answer span in c'_{ans} using an alignment algorithm. We first tokenize c'_{ans} and the answer A' using the UToken[2] tokenizer. We choose this tokenizer over existing Bengali word tokenizers since it comes paired with a detokenizer that helps greatly with the reconstruction of the sentence after the alignment process. For each of n tokens in $c'_{ans} = [c'_{a1}, c'_{a2}, \cdots, c'_{an}]$ and m tokens in $A' = [a'_1, a'_2, \cdots, a'_m]$, we find the corresponding vector representations using fastText[3] word represen-

[2] https://github.com/uhermjakob/utoken

[3] https://fasttext.cc/

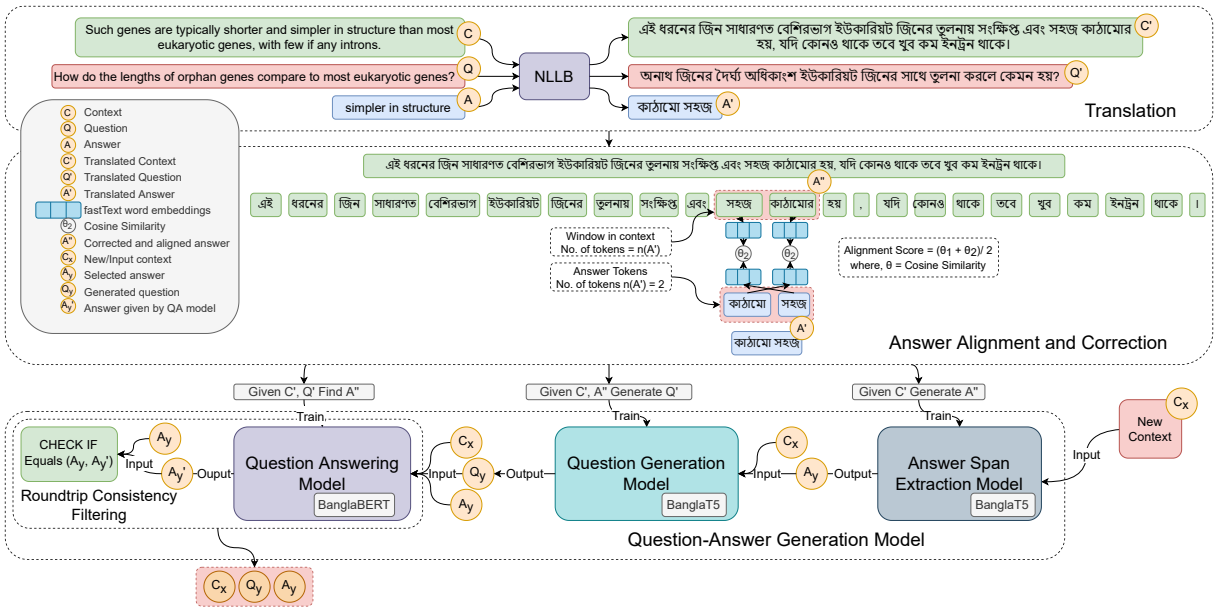

Figure 2: Conceptual diagram outlining the translation and answer alignment processes to generate the translated datasets, as well as the QAG pipeline consisting of the answer span extraction model, the question generation model, and the roundtrip consistency filtering mechanism.

tation model. In fastText word embeddings, a vector representation is associated with each character n-gram of a word. This allows us to identify similar words written differently in different contexts such as in **"সঙ্গীত শিল্পের"** and **"সঙ্গীত শিল্প"**. In each of these sentences the word **"শিল্প"** is used differently. For each window of m tokens in c'_{ans}, we compute the similarity using an alignment score, S, with the answer tokens using the following formula:

$$S = \frac{\sum_{k=i}^{i+m} \theta(c'_{ak}, a'_k)}{m}$$

where i is the starting position of the window and θ is the function for cosine similarity.

The window of m tokens in c'_{ans} which results in the maximum alignment score is selected as the correct answer span A''. The m tokens are then detokonized using the same tokenizer to find the answer span starting character index in c'_{ans}.

In some cases, the correct span in c'_{ans} has one or two more tokens than the number of tokens in A'. For example, the answer A, to a question in SQuAD2.0 is "99" which, translated to A', is **"৯৯ ডলার"**. However, we find a 3-token sequence **"৯৯ মার্কিন ডলার"** in c'_{ans} to be the most appropriate match. To account for these cases, we also run the algorithm for window sizes of $m+1$ and $m+2$.

We also find, sometimes, the correct span in c'_{ans} has a different ordering of tokens than A'. Such as one answer in SQuAD2.0 is translated as **"১২ সেপ্টেম্বর, ২০০৬"** whereas the correct span in c'_{ans} is found as **"২০০৬ সালের ১২ সেপ্টেম্বর"**. To alleviate this problem we calculate the alignment score for all permutations of A' for a window and the maximum among them is taken as the alignment score for that window.

If S_{max} is less than a specified $S_{threshold}$, we back-translate each of the tokens of A' and c'_{ans} separately to English. The back translation gives the tokens $A'_b = [a'_{b1}, a'_{b2}, \cdots, a'_{bm}]$ and $c'_b = [c'_{b1}, c'_{b2}, ...c'_{bn}]$ from A' and c'_{ans} respectively. An answer span is again selected using the same algorithm. If the new alignment score is greater than the previous alignment score by a specified δ, we select the new answer span tokens from c'_{ans}. A higher threshold and delta lead to a stricter alignment and hence more accurately translated QA pairs, but also generated more noise while a lower threshold and delta sacrificed some accuracy for reduced noise. Based on our experiments, we find that selecting a value of 0.6 for $S_{threshold}$ and 0.05 for δ provides a fair trade-off between accuracy and noise.

3.3 Question-answer generation with roundtrip consistency

The translated dataset, consisting of contexts denoted as C', questions denoted as Q' and correctly aligned answers denoted as A'', is used to train our proposed QAG pipeline. This pipeline

consists of three key elements described below:

Answer span extraction model: We formulate the problem of identifying possible answer spans for question-answer pairs as a token classification problem[4]. However, modifying our translated dataset to a token classification problem dataset would require careful tokenization and labeling of each token in the context of whether they can be possible answer spans. To avoid such rigorous modification and labeling we fine-tune BanglaT5 (Bhattacharjee et al., 2022b) in a conditional generation setting where the model receives the context sentence containing the answer c'_{ans} as the input text and the answer A'' as the target text. During inference, the model receives as input a context C_x and outputs an answer span A_y. The two are passed along to the question generation model.

Question generation model: The question generation model is used to generate questions based on the given context and an answer span from the context. For our synthetic datasets, we generate both answerable and unanswerable questions following the format of SQuAD2.0. For both, we use BanglaT5 in a conditional generation setting.

- For answerable questions, the model is fine-tuned to receive as input c'_{ans} as well as the answer A'', denoted as X_1, and output the question Q'. Since SQuAD1.1 has only answerable questions, we use our translation of SQuAD1.1 for fine-tuning in this case.

$$X_1 = [c'_{ans} < /sep > A'' < /s >]$$

- For unanswerable questions, the model is fine-tuned to receive as input c'_{ans} and "$impossible$" keyword in place of A'', denoted as X_2, and output the question Q'. During training, we select only unanswerable Q' from our translation of SQuAD2.0.

$$X_2 = [c'_{ans} < /sep > impossible < /s >]$$

During inference, the model outputs Q_y given C_x and A_y. In the case of unanswerable questions, A_y is replaced with "$impossible$".

[4] https://huggingface.co/tasks/token-classification

Roundtrip consistency filtering: In accordance with the work of Alberti et al. (2019), we adapt the roundtrip consistency filtering mechanism to discard QA pairs that are inconsistent. We fine-tune BanglaBert (Bhattacharjee et al., 2022a) on our translated datasets with the question-answering objective. During inference, this model receives C_x and Q_y from the output of the question generation model and identifies an answer span A'_y in C_x. We then compare A_y and A'_y and retain the QA pair if they are exactly similar to one another. In the case of unanswerable questions, if the QA model outputs an empty string and A_y is found to be "$impossible$", the QA pair is considered consistent.

4 Experimental Setup

4.1 Datasets

SQuAD1.1 and SQuAD2.0: We use translations of popular question-answering datasets, SQuAD1.1 and SQuAD2.0 for training our QAG pipeline. SQuAD1.1 consists of paragraphs from Wikipedia[5] and crowdsourced question-answer pairs. There are 536 paragraphs divided into 23,215 contexts and 107,785 question-answer pairs in the dataset. The answers are a span of tokens or words within the texts. Since the questions are crowdsourced there is a diverse range of questions in the datasets. SQuAD2.0 adds over 50,000 unanswerable questions to the SQuAD1.1 dataset. Since the test sets of the SQuAD are not public, we only use the translations of the train and validation sets to produce SQuADBangla1.1 and SQuADBangla2.0 after correcting the alignment of the answers and filtering out question-answer pairs with low alignment scores. For SQuADBangla1.1, we use the validation set of the original SQuAD1.1 as the test set, the first 400 paragraphs of the SQuAD1.1's train set as the train set, and the remaining 42 paragraphs of SQuAD1.1's train set as the validation set. We follow the same technique to produce the train, validation and test sets for SQuADBangla2.0.

Bengali-SQuAD: Tahsin Mayeesha et al. (2021) used the Google Cloud Translation API to translate 294 paragraphs from SQuAD2.0 to produce the translated dataset Bengali-SQuAD. The authors use random splitting to choose 235 paragraphs with 73,812 QA pairs as the training

[5] https://www.wikipedia.org/

set and the remaining 59 paragraphs with 17,607 QA pairs as the validation set. For the test set, they collected Bengali Wikipedia articles and made 300 QA pairs which they did not make publicly available. For our purpose of comparison, we use the validation set of Bengali-SQuAD as the test set and split the train set to use the first 200 articles as training data and the rest as validation data.

SQuAD_Bn: SQuAD_Bn was presented by Bhattacharjee et al. (2022a) combining translated SQuAD2.0 and the Bengali portion of TyDiQA as part of a natural language understanding benchmark in Bengali. The authors use the translations of both the train and validation sets of SQuAD2.0 as the train set of SQuAD_Bn and use the Bengali portion of TyDiQA as validation and test sets. The train set of SQuAD_Bn consists of 477 paragraphs with 118,117 QA pairs. The validation set has 1,221 paragraphs with only 2,502 QA pairs and the test set has 1,282 paragraphs with only 2,504 QA pairs.

BARD: Tanvir Alam and Mofijul Islam (2018) present BARD, a Bengali article classification dataset, in their work on the task of document classification in Bengali. BARD consists of around 376,226 articles collected from different Bengali news portals. The authors consider only the news articles that fall within five categories: state, international, economy, entertainment, and sports. We use a subset of the BARD dataset's articles to generate QA pairs for BanglaQA, our synthetic QA dataset.

4.2 Implementation Detail

We use the Hugging Face[6] implementation and pretrained checkpoints from the Hugging Face library for all required models except for fastText word vector model. For translation, we use the checkpoint "facebook/nllb-200-3.3B" with 3.3 billion parameters from the Hugging Face library. For fastText word embeddings, we use the pretrained word vector model for Bengali, trained on Common Crawl[7] and Wikipedia. For back-translation, we use the "csebuetnlp/banglat5_nmt_bn_en" model checkpoint. To train our answer span extraction model, we use the model checkpoint "csebuetnlp/banglat5" with 247 million parameters. We fine-tune the answer span extraction model for 3 epochs with a batch size of 8, a learning rate of 3e-5, max length of input text 128, and a max length of output text 30. This required around 3 GPU training hours. For our QA models, we use the pretrained model checkpoints "bert-base-multilingual-uncased", "xlm-roberta-base" and "csebuetnlp/banglabert" with 180, 270, and 110 million parameters respectively. We fine-tune 3 epochs with a batch size of 16 and a learning rate of 2e-5. This required around 2 GPU training hours. For each dataset, we fine-tune the QA models only once to reduce the carbon footprint. For QG, we use the model checkpoint "csebuetnlp/banglat5" which is fine-tuned for 3 epochs with a learning rate of 2e-4 and a batch size of 16. This takes around 2 to 3 GPU hours. The max input length for QG is taken to be 512 and the max output length is taken to be 64. We use an Nvidia GeForce RTX3090 GPU with 24 GB VRAM for all our experiments.

4.3 Evaluation Metrics

In accordance with prior literature, we use the EM and F1 scores to establish a benchmark of baseline scores on our synthetically generated QA dataset. We also quantitatively assess our translated datasets by fine-tuning QA models on them and evaluating them on other test sets.

To assess the quality of our synthetic dataset, we choose five different criteria as outlined below:

- **Grammatical accuracy:** We consider a QA pair to be grammatically accurate only if the question and the answer both had no grammatical errors.

- **Relevance to context:** If the question is based on the context and the answer can be derived from the context, the QA pair is considered relevant. We discard unanswerable questions for this criterion since they are impossible to answer from the information in the context.

- **Consistency of QA pairs:** We consider a QA pair to be consistent if the answer span actually answers the question. Unanswerable QA pairs are deemed to be consistent if there are no answers to them.

[6] https://huggingface.co
[7] https://commoncrawl.org

Train Datasets	Test Datasets			
	SQuADBangla1.1	SQuADBangla2.0	Bengali-SQuAD	SQuAD_Bn
		mBERT		
SQuADBangla1.1	54.57/70.65	27.42/35.18	13.30/25.18	56.07/63.17
SQuADBangla2.0	54.39/70.11	59.47/63.76	44.61/52.87	66.05/71.74
Bengali-SQuAD	7.74/47.84	46.05/54.01	**48.81/54.37**	53.15/61.71
SQuAD_Bn	46.79/64.68	60.55/65.73	42.89/51.37	67.05/70.02
		XLM-RoBERTa		
SQuADBangla1.1	58.18/73.73	27.52/35.24	10.86/23.21	45.25/52.62
SQuADBangla2.0	57.40/73.01	60.23/64.90	42.87/51.66	65.69/72.07
Bengali-SQuAD	7.96/49.98	43.43/53.42	**46.88/53.55**	53.71/62.45
SQuAD_Bn	47.95/66.09	60.67/66.26	43.04/51.50	67.75/73.13
		BanglaBERT		
SQuADBangla1.1	**62.69/78.09**	29.32/36.82	11.26/24.24	31.31/38.29
SQuADBangla2.0	**62.18/77.87**	**65.08/71.05**	42.10/53.15	**69.81/75.38**
Bengali-SQuAD	12.28/57.84	44.02/58.02	42.92/53.35	53.35/67.87
SQuAD_Bn	56.33/73.78	**67.53/74.47**	41.30/52.30	**70.69/76.79**

Table 1: Benchmark scores of different models fine-tuned and tested on different datasets. The scores emphasized in bold in every column are the top 2 EM/F1 scores for that particular dataset in our experiments.

- **Conciseness of answers:** An answer span is considered concise if it contained no words or characters beyond the actual answer to the question. Unanswerable questions are discarded for this criterion since they have no answer to assess.
- **Diversity of questions:** To quantify diversity, we opt for a binary mark of 1 or 0. We ask assessors at the end of each article whether the questions for that article are diverse in nature spanning different question types like "why", "where", "how", "who", "when" etc.

5 Experimental Results

5.1 Translated datasets

A comparison of different translated QA datasets along with SQuADBangla1.1 and SQuADBangla2.0 is shown in Table 1.

Of all three models assessed, we found BanglaBERT to significantly outperform the other two, XLM-RoBERTa (Conneau et al., 2020) and mBERT (Devlin et al., 2019). This is because XLM-RoBERTa and mBERT are both multilingual language models trained on huge corpora comprising multiple languages, whereas BanglaBERT is trained on specifically Bengali corpora.

Aligned with this, we find that fine-tuning BanglaBERT on SQuADBangla2.0 results in consistently good performances on all datasets. On SQuAD_Bn, this combination posts an EM of 69.81 and an F1 score of 75.38 and on SQuADBangla1.1, it scores 62.18 EM and 77.87 F1. Testing this combinations performance on SQuADBangla2.0 itself, we find an EM of 65.08 and an F1 score of 71.05. Fine-tuning on SQuADBangla2.0 yields consistently high performance across all datasets, establishing SQuADBangla2.0 as a robust and comprehensive Bengali QA dataset. We observe the lowest EM and F1 scores with Bengali-SQuAD and attribute it to discrepancies in the translation and answer span marking, as identified previously. The highest EM and F1 scores posted on Bengali-SQuAD are 48.81 and 54.37 respectively by mBERT fine-tuned on Bengali-SQuAD itself. Even so, fine-tuning mBERT and XLM-RoBERTa on SQuADBangla2.0 results in comparable performance at 44.61 EM and 52.87 F1 and 42.87 EM and 51.66 F1 respectively for each model. We also find that models fine-tuned on SQuADBangla1.1 do not perform well on other datasets. This is primarily because SQuADBangla1.1 does not consist of unanswerable questions.

5.2 Synthetic dataset: BanglaQA

In order to assess the performance of our QAG pipeline on native Bengali text, we use the pipeline on the BARD dataset, a collection of news articles written by native Bengali speakers scraped from trusted, popular online news portals, to generate our synthetic QA dataset, BanglaQA. The distribution of articles from each of the five categories to

Categories	Articles	QA Pairs	Unanswerable Questions
State	3090	29901	5910
Economy	2660	24964	4903
International	3530	38782	7778
Sports	3308	40731	8299
Entertainment	3409	43634	8941
Total	**15997**	**178012**	**35831**

Table 2: Distribution of news categories in the articles and QA data in BanglaQA.

use as contexts and the number of QA pairs under each category is shown in Table 2.

Table 3 shows the distribution of articles and QA pairs for our train, validation, and test sets. We retained 80% of the dataset for our training set resulting in 142,536 QA pairs across 12,797 articles. Of the remainder, 10% was allocated to the validation set, resulting in 17,861 QA pairs across 1,600 articles, and 10% to the test set, resulting in 17,615 QA pairs across 1,600 articles. BanglaQA has more than 170,000 QA pairs in total whereas the previously available datasets Bengali-SQuAD and SQuAD_Bn have roughly 90,000 QA pairs and 123,000 QA pairs respectively.

Set	Articles	QA Pairs	Unanswerable Questions
Train	12797	142536	28720
Validation	1600	17861	3576
Test	1600	17615	3535
Total	**15997**	**178012**	**35831**

Table 3: Statistics of BanglaQA train, validation, and test sets.

We extracted a random sampling of 2,100 QA pairs across 204 articles and asked a group of seven university students with a firm grasp of Bengali to assess them as per the criteria. Each student assessed 300 QA pairs. The summary of results is presented in Table 4. BanglaQA achieves a human-evaluated score of 98% in terms of grammatical accuracy. We attribute this to the syntactical accuracy of the translated SQuADBangla datasets that we used to train the QAG pipeline as well as the BARD dataset that we took the articles from. The QA pairs in BanglaQA are also mostly relevant to the context and consistent within themselves.

Acknowledging the general scarcity of human-annotated Bengali QA data, we show the use of BanglaQA as a standalone synthetic dataset. To that end, we provide a benchmark of baseline scores on BanglaQA. We show the performance of mBERT, XLM-RoBERTa, and BanglaBERT on BanglaQA in Table 5. Consistent with the fact that BanglaBERT is the only language model pre-trained on a solely Bengali corpus of the 3 models tested, it performs the best scoring 75.70 EM points and 86.14 F1. Furthermore, we examine the performance of models trained on BanglaQA on the test set of the SQuAD_Bn dataset. This particular test set is derived from the Bengali section of TyDiQA, which provides a human-annotated benchmark for evaluating model performance.

Criteria	Score
Grammatical accuracy	98%
Relevance to context	97%
Consistency of QA pairs	96%
Conciseness of answers	88%
Diversity of questions	64%

Table 4: Results of human evaluation of BanglaQA's quality on five different metrics.

Model	BanglaQA EM/F1	SQuAD_Bn EM/F1
mBERT	68.70/78.01	57.03/61.51
XLMRoBERTa	74.18/84.58	57.59/64.17
BanglaBERT	**75.70/86.14**	**57.74/65.41**

Table 5: Benchmark scores of different models trained on BanglaQA

6 Conclusion and Future Work

Prior research on QA for the Bengali language has been significantly hindered by a lack of large-scale Bengali QA datasets. Given the laborious nature of human annotation, the only solution explored so far has involved machine-translated versions of popular QA datasets. In this work, we propose an alternative approach through a QAG pipeline tailored for Bengali and demonstrate its effectiveness by generating the synthetic BanglaQA dataset. We also produce two empirically better translated datasets, SQuADBangla1.1 and SQuADBangla2.0, to train our QAG pipeline. Further, we assess the quality of BanglaQA by human evaluation on five different metrics and establish it as a benchmark Bengali QA dataset, reporting the baseline performance of three different QA models on it. The BanglaQA dataset should provide new opportunities to develop and evaluate Bengali QA systems, helping address the

shortage of training data. Future work on extending these QA generation techniques to additional tasks such as logical reasoning and multi-hop QA may further advance the capabilities of Bengali QA. We hope the BanglaQA dataset and the QAG framework presented here will inspire continued research into QA for other low-resource languages.

Limitations

Noting that this is the first work of its kind in the Bengali language, our work is not without its limitations. While the NLLB model achieves state-of-the-art English-to-Bengali performance, we acknowledge the possibility of residual errors in the translation and alignment of the translated SQuAD dataset. However, the results in Table 1 demonstrate that models trained on our SQuAD-Bangla datasets achieve strong performance when evaluated on other datasets. Furthermore, the QAG pipeline is given human-written Bengali text as input contexts and the QA pairs are synthetically generated from them without any further need for translation or alignment. Given the relatively high human evaluation scores for our BanglaQA dataset, we deduce these errors minimally impact the quality of the generated QA pairs.

Our method for answer span extraction in the QAG pipeline is not suited for multi-hop QA and deeper logical reasoning, which have garnered great interest recently, resulting in datasets like HotpotQA (Yang et al., 2018) and NarrativeQA (Kočiský et al., 2018). The answer spans extracted are from text sequences exactly as they are present in the contexts and the generated questions reflect this in their nature.

The embedding-based alignment algorithm that we used to identify the answer spans post-translation may not work as is for other languages because of syntactic and semantic differences. However, the principle should be easily adaptable to these languages as well.

Ethical Considerations

In compliance with the Copyright Act, 2008[8], Bangladesh, we are publicly releasing all the translated and synthetic datasets generated as a result of this work. There are also no concerns about copyright infringement issues since all of the datasets that we use are already publicly available for non-commercial research usage.

In appreciation of their efforts in assessing the quality of BanglaQA, the seven students we selected for human evaluation were given appropriate remunerations at standard rates.

BanglaQA being a synthetic dataset based on Bengali news articles may also be prone to negative bias. This is because news articles may often highlight negative incidents, political biases, and certain stereotypes. This is not a serious issue since it is very specific to the domain of news articles, which, by nature, revolve around such content. However, we can not guarantee that there will not be any serious biases from synthetic datasets generated by this work since this is heavily dependent on the choice of source contexts.

[8] http://bdlaws.minlaw.gov.bd/act-details-846.html

References

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

Somnath Banerjee, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. 2014. Bfqa: A bengali factoid question answering system. In *Text, Speech and Dialogue*, pages 217–224, Cham. Springer International Publishing.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022a. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, and Rifat Shahriyar. 2022b. Banglanlg: Benchmarks and resources for evaluating low-resource natural language generation in bangla. *CoRR*, abs/2205.11081.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162, Hong Kong, China. Association for Computational Linguistics.

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.

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.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Sanjidul Hoque, Mohammad Shamsul Arefin, and Mohammed Moshiul Hoque. 2015. Bqas: A bilingual question answering system. In *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, pages 586–591.

Samina Tasnia Islam and Mohammad Nurul Huda. 2019. Design and development of question answering system in bangla language from multiple documents. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–4.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2018. Improving neural question generation using answer separation. In *AAAI Conference on Artificial Intelligence*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. 2020. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224, Online. Association for Computational Linguistics.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019. Recent advances in neural question generation. *ArXiv*, abs/1905.08949.

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.

Tasmiah Tahsin Mayeesha, Abdullah Md Sarwar, and Rashedur M Rahman. 2021. Deep learning based question answering system in bengali. *Journal of Information and Telecommunication*, 5(2):145–178.

Md Tanvir Alam and Md Mofijul Islam. 2018. Bard: Bangla article classification using a new comprehensive dataset. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–5.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

Some examples of QA pairs generated using our QAG pipeline are shown in table 6. These QA pairs are generated using articles from BARD.

Context:	তাঁর পরিবর্তে 'মারদানি' ছবির অভিনেতা তাহির রাজ ভাসিনকে নেওয়া হবে বলেও শোনা যাচ্ছিল।
Question:	মারদানি'' ছবির অভিনেতা কে ছিলেন**?**
Answer:	তাহির রাজ ভাসিনকে
Context:	গত কার্যদিবসে মোট লেনদেনের পরিমাণ ছিল ৩৪ কোটি ৫৬ লাখ টাকা।
Question:	গত কার্যদিবসে মোট লেনদেনের পরিমাণ কত ছিল**?**
Answer:	৩৪ কোটি ৫৬ লাখ টাকা
Context:	ফিলিপাইনের রাজধানী ম্যানিলা থেকে এ ঘোষণা দেওয়া হয়েছে।
Question:	ফিলিপাইনের রাজধানী কি**?**
Answer:	ম্যানিলা
Context:	তখন ভোক্তাকে নিজের নাম, ঠিকানা ও ইউনিক কোডটি উল্লেখ করে তিনটি প্রশ্নের উত্তর দিতে হবে।
Question:	ভোক্তাকে তার নাম, ঠিকানা এবং ইউনিক কোড উল্লেখ করে কি করতে হবে**?**
Answer:	তিনটি প্রশ্নের উত্তর দিতে হবে
Context:	বৈদ্যুতিক শর্টসার্কিট থেকে এ অগ্নিকাণ্ডের সূত্রপাত হয়েছে বলে নিশ্চিত হওয়া গেছে।
Question:	কোন কারণে এই আগুনের সূত্রপাত হয়েছিল বলে জানা গেছে**?**
Answer:	বৈদ্যুতিক শর্টসার্কিট
Context:	ছোটদের মাসিক সাময়িকী কিশোর আলোর জন্মবার্ষিকী উপলক্ষে ওই মেলার আয়োজন করা হয়েছে।
Question:	কোন পত্রিকার জন্মদিন উপলক্ষে এই মেলা অনুষ্ঠিত হয়**?**
Answer:	কিশোর আলোর
Context:	রাজধানীর তেজগাঁওয়ে গতকাল বুধবার বিএসটিআইয়ের প্রধান কার্যালয়ে এ চারটি প্রতিষ্ঠানের প্রতিনিধির কাছে সনদ হস্তান্তর করেন প্রতিষ্ঠানটির মহাপরিচালক ইকরামুল হক।
Question:	বিএসটিআই এর প্রধান কে ছিলেন**?**
Answer:	ইকরামুল হক
Context:	এঁদের মধ্যে মুত্তিয়া মুরালিধরন ও শেন ওয়ার্নের উইকেটসংখ্যা হাজারের ওপর।
Question:	কোন দুইজন খেলোয়াড় ১০০০ এর বেশি উইকেট নিয়েছেন**?**
Answer:	মুত্তিয়া মুরালিধরন ও শেন ওয়ার্নের
Context:	আর কিউবার প্রতিনিধিদলটির নেতৃত্বে দেশটির পররাষ্ট্র মন্ত্রণালয়ের যুক্তরাষ্ট্রবিষয়ক পরিচালক হোসেফিনা ভিদাল।
Question:	কিউবার পররাষ্ট্র মন্ত্রণালয়ের যুক্তরাষ্ট্র বিষয়ক পরিচালক কে ছিলেন**?**
Answer:	হোসেফিনা ভিদাল
Context:	কেননা, বাংলাদেশের যত রপ্তানি হয়, এর ৫০ শতাংশের মতো ইউরোপীয় ইউনিয়নে হয়।
Question:	কত শতাংশ রপ্তানি ইউরোপীয় ইউনিয়নে হয়**?**
Answer:	৫০ শতাংশের
Context:	এ সময় তাঁর সঙ্গে ছিলেন জেলা প্রশাসক মো. সেলিম রেজা, বোরহানউদ্দিন উপজেলা পরিষদের চেয়ারম্যান মোহাব্বাতজান চৌধুরী, পশ্চিমাঞ্চলীয় বিদ্যু বিতরণ কোম্পানির ব্যবস্থাপনা পরিচালক সরোয়ার হোসেন প্রমুখ।
Question:	জেলা প্রশাসক কে ছিলেন**?**
Answer:	মো. সেলিম রেজা
Context:	ব্রাজিলের ব্রুনো সোরেসের সঙ্গে জুটি বেঁধে ভারতীয় টেনিস তারকা জিতেছেন প্রথমবারের মতো ইউএস ওপেনের একটা শিরোপা।
Question:	কে ভারতীয় টেনিস তারকাকে ইউএস ওপেনের শিরোপা জিততে সাহায্য করেছে**?**
Answer:	ব্রুনো সোরেসের

Table 6: Examples from BanglaQA. Only the portion of the context containing the answer span is shown for each example.