A Semi-Automated Approach to Generate Bangla Dataset for Question-Answering and Query-Based Text Summarization

Mueeze Al Mushabbir and **Refaat Mohammad Alamgir** and **Ahmed Azaz Humdoon** and **Dr. Hasan Mahmud** and **Dr. Kamrul Hasan**
Department of Computer Science and Engineering
Islamic University of Technology
{almushabbir, refaatalamgir, azazhuoon, hasan, hasank}@iut-dhaka.edu

Abstract

With the vast amount of information available on the Internet, finding answers to questions is as important as ever in today's day and age. In Natural Language Processing Research, Question Answering (QA) and Query-based Text Summarization (QBSUM) are there to tackle this challenge. However, most of the work being done neglects low resource languages such as Bangla, resulting in the small number of quality datasets available in the literature. Therefore to address this research gap, in this work, we propose a semi-automated methodology for generating a Bangla dataset with Natural Questions for three tasks - Question Answering (QA), Query-based Single Document Text Summarization (SD-QBSUM) and Query-based Multi-Document Text Summarization (MD-QBSUM). We then provide baselines for this dataset on those tasks and also compare our dataset with existing ones on various metrics.

1 Introduction

In the advent and progress in the domain Natural Language Processing (NLP) and Deep Learning (DL) around the world, tasks like Question-Answering (QA) and Query-based Summarization (QBSUM) have gained a lot of traction and popularity in recent years. However, a closer look in the domain shows that majority of the work has mostly been done only in English language and not in low-resource languages like Bangla, which is the very reason that inspired us to take up on the research work in the domain of QA and QBSUM in Bangla.

In order to address this research gap, we target 3 dataests - Question Answering (QA), Single Document Query Based Summarization (SD-QBSUM), and Multi-Document Query Based Summarization (MD-QBSUM). These 3 datasets will be in Bangla, which has only few and low quality datasets for QA, and no dataset at all for SD-QBSUM or for MD-QBSUM. The quesries or questions for all of these datasets are the same, and are Bangla translated queries from Google Natural Question (NQ) (Kwiatkowski et al., 2019). These are Natural Questions, which means these are the questions that geneal people actually queried in the Google search engine, as opposed to the Synthetic Questions which are generated by machines or people working solely for the purpose of creating question in order to create a dataset. Our approach also includes a Semi-Automatic method where we have a Human-in-the-loop (HITL) and a Semi-Supervised Annotation approach (as shown in the Figure1.

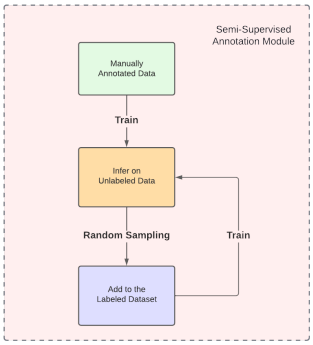


Figure 1: Semi-Supervised Annotation Module

In short, our contributions are:

- We propose a single pipeline to create 3 datasets to tackle 3 tasks - Question Answering, Single-Document Query Based Summarization (SD-QBSUM) and Multi-Document Query Based Summarization (MD-QBSUM)
- We include, in order to create the datasets, a Semi-Automatic approach including Human-in-the-loop (HITL) and Semi-Supervised Annotation
- We make Natural Questions available in Bangla
- We present Baselines for the datasets

The rest of the paper consists of our literature review in section 2, our proposed methodology in section 3, data preparation in section 4, our experi-

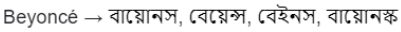


Figure 2: Inconsistent spellings

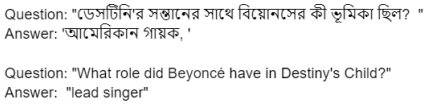


Figure 3: Wrong answer

mental setup in section 6, and baseline and result analysis in section 7.

2 Literature Review

In our literature review, we present 3 works that are related to our work. The first one being Bengli SQuAD, which translated SQuAD 2.0 dataset into Bengali, but only for question answering. We also present another related work TyDiQA, which has multilingual QA dataset including Bangla. We review these 2 datasets for QA task in Bangla. Additionally, since there are no work done in terms of Single and Multi-document Query Based Summarization in Bangla, we do not have any related work specifically for these tasks in Bangla. However, we present a related paper AQUAMUSE that proposes a method to turn a Single Document Query Bassed Summarization to a Multi-Document Query Based Summarization.

2.1 Bengali SQuAD

The key contribution in this paper (Tahsin Mayeesha et al., 2021) is creating a QA dataset in Bangla and this was done by machine-translating the most well known QA dataset that exists in the literature, i.e. SQuAD 2.0 (Rajpurkar et al., 2018). Despite the valiant effort that went into creating this, we have identified some issues in their dataset which we hope to address in our work. Most of the problems were related to their translation. We have observed that the answers sometimes fail to recognize named entities and even when they do, spellings are not always kept consistent across the whole dataset as shown in Figure 2.

Unnatural and semantically incoherent translations such as the translated question in Figure 3 were also seen. In the same figure, we can also notice that the target label was wrongly annotated. And lastly, in other cases, some answers were completely missed and therefore those were empty while their English counterparts were not. Most importantly, they have not made their full dataset public which made it difficult to do an extensive analysis as only some samples were publicly available.

2.2 TyDi QA

TyDi QA is a dataset for question answering task. It includes data in 11 languages that are, according to them, "typologically" different. (Clark et al., 2020) They propose their dataset for a couple of tasks - Primary and Secondary. (Clark et al., 2020) In primary task, the core task is to identify a passage or a minimum span of words that includes the answer, or return no answer if there are not any. Conversely, the secondary task, also known as Gold Passage task, guarantees that there are answers and the task is to find them. [1]

2.3 AQUAMUSE

The paper AQUAMUSE proposes a pipeline for automatically generating datasets for both abstractive and extractive multi-document QBSUM. (Kulkarni et al., 2020) Long answers are taken from the NQ dataset and are considered to be the summaries. These summaries are then matched with embedded document sentences collected from a cleaned version of Common Crawl known as the Colossal Clean Crawled Corpus. (Raffel et al., 2019) The matching is done slightly differently for the two types of query-based text summarization, with having semantic relevance for the abstractive summaries and in-place substitution for the extractive summaries.

Even though this pipeline is well laid out, one will run into a brick wall if they are to apply the same technique for Bangla. One of the key reasons for this is that NQ does not have Bangla dataset and the corresponding Wikipedia articles for the NQ dataset may not have the same content. Secondly, annotation is done in a discriminative fashion (which means that summaries already exist as long answers and documents are checked with these summaries to determine the semantic relevance) rather than in a generative mode (where summaries are written either by experts or by machines). Furthermore, the long answers from Wikipedia are kept as the overall summarized answer, which means that the answer only reflects

[1] https://github.com/google-research-datasets/tydiqa

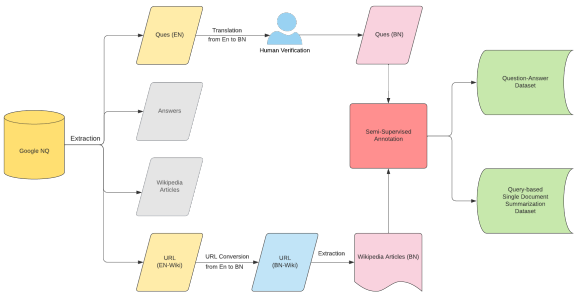


Figure 4: Proposed Methodology - Single Document QA + QBSUM

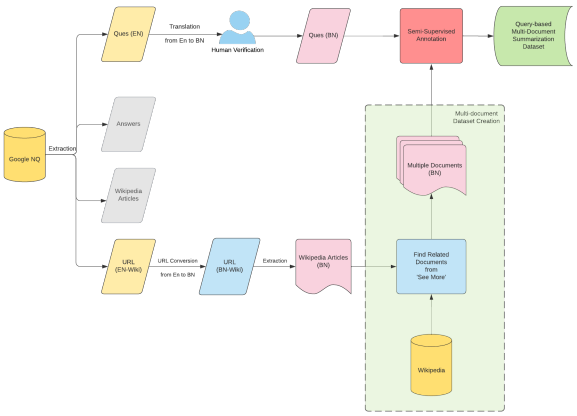


Figure 5: Proposed Methodology - Multi-Document QBSUM

Wikipedia content and not other multiple relevant documents.

3 Proposed Methodology

Our idea is a single process to generate datasets for three different tasks - QA, QBSUM-SD, and QBSUM-MD. The whole process can further be thought of as two sub-process as explained in the following sections.

3.1 Single Document summarization

As illustrated in Figure 4 , we first extract some elements from the Google NQ Dataset - Questions (in English), Answers, Wikipedia Articles and their URLs. Among these elements, only the questions and URLs are needed in the next step so we ignore the rest. The questions will be translated to Bangla and this will be a HITL(Human-in-the-loop) process, i.e. the translations will be done by machine and supervised by a human. On the other side, we retrieve the corresponding Bangla Wikipedia articles if they exist. Finally, we will use these Wikipedia articles and provide them to the Semi-Supervised Module where there will be labelling performed automatically and the first set of true labelling will be annotated by human annotators who will mark the short answers (which will be necessary for Question Answering dataset) and long answers (which will be used to create the Query-based Text Summarization Single Document dataset).

3.2 Multi-Document summarization

The method for generating dataset for QBSUM for Multi-Document is quite similar to that for QA and QBSUM-SD with a few key differences as shown in Figure 5. In the previous sub-process where the corresponding Bangla Wikipedia articles are retrieved, those documents are matched with different documents from Common Crawl and Banglapedia, usin g cosine similarity. Thus, we will now have multiple documents which will be the context for QBSUM-MD. On the other side, we will also have the questions translated with HITL as before. Finally, these questions and multiple documents will be provided to the Semi-Supervised Module for annotation. In this module, there will be annotation done automatically based on the first set of true labelling that is done by human annotators, who will find the answers to these questions, thereby completing the creation of the dataset.

3.3 Semi-Supervised Annotation Module

As mentioned before, we adopt a semi-supervised approach to annotate the data. this way there is portion of the data that is manually annotated by human annotators. And the rest of the annotation comes from training the model and inferring on the unlabeled data.

4 Dataset

Our Dataset creation is followed as shown in the figures 4 and 5.

4.1 Data Extraction from Google Natural QUestion

In Google Natural Question (NQ), it consists of English data. The attributes of each data are: URL, Question, Context, Answer, Answer Tokens/Bytes. Of the 2 types of dataset in NQ, we collect data from the non-simplified version.

4.2 Bangla Wikipedia Page Scrape and Cleaning

From the data samples collected from NQ, we go through each of them, and check if there exists any URL for Bangla Wikipedia page of the same topic from the corresponding URL of the English Wikipedia page. If the URL is valid, we collect the text contents using Wikipedia Python library. The tables in the wikipedia pages pose a challenge since cleaning and parsing the table data are very complex and difficult. So, we used Python Pandas library. It reads the table html and generates a DataFrame. From there, we can gather the texts and add to the scraped and cleaned data.

4.3 English Question Translation

The data we collected from NQ contains questions in English language. We need to translate them to Bangla. So we use the Google Translation API, and translate all the questions to Bangla. Afterwards, to verify the data, we manually verify them by humans - this is one point where we adopted the HITL approach.

4.4 Annotation

From the wikipedia bangla pages scraped and the questions translated to bangla, we now apply Semi-Supervised Annotation Module here. First, there is a human annotator who annotates some data manually. These are considered as gold samples because the human annotator annotation process is considered rarely noisy.

After human annotators annotated manually, the labeled data are used in the automatic portion of the Semi-Supervised Annotation Module, where a model is trained, inferred on unseen data, and randomly appended a portion of the predictions to the annotated data.

4.5 Multi-Document Corpus Collection

Our initial approach was to find similar documents based on similarity measures like shown in AQUAMUSE. However, we find that the encoder used in the AQUAMUSE - Universal Sentence Encoder, does not perform well in Bangla Language, and the number of documents contained in the C4 dataset is over 7 million. Finding similar documents in this vast pool of documents is not resourcefully feasible. So, we adopt to a different approach. In the newer approach, as shown in the figure 5, we extract URLs of the articles related to the given article from the bottom of the page, where there is a section 'related articles'. The article itself notes to other articles that are related to this one. Thus, we create a data sample with set of related documents .

5 Experimental Setup

To train our models we used GPU from google colab with specifications of

- GPU: K80 or T4 (12GB VRAM).
- RAM: 16GB.
- STORAGE: 80GB.

For baselines and experiments, we chose several models as following:

QA	SD-QBSUM & MD-QBSUM
mT5	xlm-r-100langs-bert-base-nli-mean-tokens
XLM-RoBERTa	Paraphrase-xlm-r-multilingual-v1
mBERT	paraphrase-multilingual-mpnet-base-v2

Table 1: Models Chosen for Baseline

6 Baseline and Result Analysis

In this section, we detail out and discuss our baselines, results and their analysis. We show Statistical analysis, comparisons among other datasets and baseline scores.

6.1 Statistical Analysis of the Dataset

To see the statistical information about our dataset, let us refer to the following table:

	Our Dataset		
Metric	**Question Answering (around)**	**Single Document QBSUM (around)**	**Multi-Document QBSUM (around)**
No.of Data	38k		2k (each with 3 different documents)
No. of Unique Context	30k		5k
Avg. Context Length (Words/Char)	486 / 3k		
Avg. Question Length (Words/Char)	7 / 45		
Avg.Answer Length (Words/Char)	3 / 18	58 / 343	
Avg. Lexical Overlapping (Words) [Question and Ans Sentence]	1.25	1.33	

Table 2: Quantitative Analysis on our Dataset

In table 2, we see the statistical information about our dataset, including: number of samples, unique context, average length of question, context, and answers etc. About how much answers and what type of answer do we have, is described in figure 6.

6.2 Comparison Among QA Datasets

An in-detail comparison among the three datasets of the QA is given below, and also stated in the later sections:

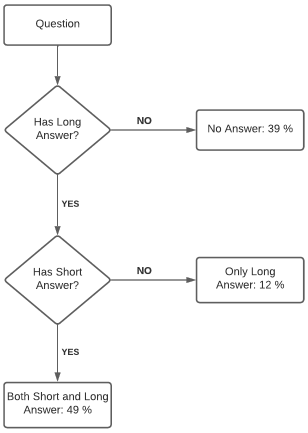
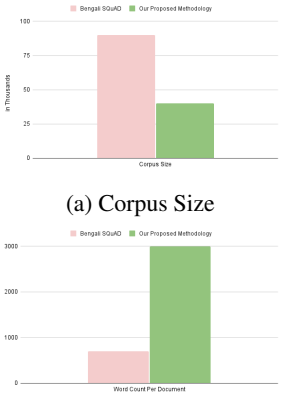


Figure 6: Dataset Types of Answers Count



(a) Corpus Size

(b) Word Count Per Document

Figure 7: Our Dataset compared to Bengali SQuAD

Metric	Bengali SQuAD (around)	TyDiQA (around)	Our Dataset (around)
No.of Data	**90k**	10k	38k
No. of Unique Context	14k	1.8k	**30k**
Avg. Context Length (Words/Char)	99.5/705	104/641	**486/3k**
Avg. Question Length (Words/Char)	**8/56**	9/48	7/45
Avg.Answer Length (Words/Char)	2/13	2/14	**3/18**
Avg. Lexical Overlapping (Words) [Question and Ans Sentence]	1.96	1.8	**1.25**

Table 3: Quantitative Analysis on QA Datasets

6.2.1 Our Dataset vs Bengali SQuAD

The main difference between our work and that of Bengali SQuAD will be in how the answers will be documented. In contrast to their machine translation approach, we plan to annotate our answers through humans. Even though the corpus size of our context, Wikipedia articles (around 30 thousand), is smaller in comparison to Bengali SQuAD (around 90 thousand), contexts of our corpus will still contain more words on average per document (3000 words on average) compared to Bengali SQuAD (which contains 700 words on average). These comparisons are visualized with bar graph in figures 7a and 7b. And lastly, the questions to be used in our dataset will be natural and taken from Google NQ, and translated to Bangla, whereas Bengali SQuAD translated the questions (which were synthetic) from the SQuAD 2.0 dataset.

6.2.2 Our Dataset vs AQUAMUSE

The first way in which our proposed methodology differs is in the annotation process. AQUAMUSE followed a predominantly discriminative approach, where they already had the summaries (from long answers in Google NQ), and their task was simply to identify the documents in Common Crawl that best match these summaries, using semantic similarity. However, we will follow a combination of discriminative and generative approach. We will first match the documents (i.e. Wikipedia articles) with those from the Common Crawl, and once we do that we will generate a summary which will be done by humans.

6.3 Evaluation Metric

All our measurements are done on Exact Matching. Exact Matching refers to complete full string matching between the Gold Answer (actual answer) and the Predicted answer. A prediction is counted as positive if the prediction matches with the Gold Answer.

6.4 Baseline

This section refers to the baseline scores that we have obtained by training and testing various models on our dataset. The models are mentioned in the Experimental Setup.

6.4.1 Baseline Comparing Among Question Answering Datasets

For comparing the baselines among the different question answering datasets, we have first separated 3 training sets and 3 test sets that come from Bengali SQuAD, TyDiQA and our dataset. These test sets were used to calculate the exact matching scores. Then the trained datasets were evaluated on each of the 3 test sets.

		Trained On		
		Bengali SQuAD	TyDiQA	Our Dataset
Tested On	Bengali SQuAD	4.0	6.0	2.0
	TyDiQA	6.0	11.0	8.0
	Our Dataset	4.0	14.0	42.0

Table 4: Baseline - Question Answering (using pretrained mT5 model on all the datasets)

The mT5 model trained on Bengali SQuAD per-

forms poorly on all the test sets. This is consistent with our belief that Bengali SQuAD contains a lot of noise and inconsistencies in the dataset so consequently it does not produce satisfactory results. Furthermore, the models trained on other datasets also gave poor performance on the Bengali SQuAD and this suggests that Bengali SQuAD is unsuitable for any sort of training or testing.

On the other hand, the model trained on TyDiQA had unsatisfactory performance on our test set and similarly the model trained on our dataset had similar results on the TyDiQA test set. We can attribute this fact to the different nature of the two datasets. We have used the TyDiQA gold passage dataset for training, and in this dataset every example is guaranteed to have an answer. However, in our dataset, the examples may or may not have an answer.

6.4.2 Baseline of Question Answering on Our Dataset

In the first part of the experiment, we apply a zero-shot setting, where the model has no idea about the training dataset, and it was simply used to perform inference on the test set and the output scores were reported. After training the models, we observe a significant rise in performance of all the models. This shows that our dataset is suitable for training a model for the task of question answering.

Model No.	Model Name	Zero-Shot Setting (Not Trained on Our Dataset)	Trained on Our Dataset
1	mT5	6.0	42.0
2	mBERT	26.0	34.0
3	xlm-ROBERTa	28.0	40.0

Table 5: Baseline - Question Answering (Exact Match expressed as a percentage (%))

6.4.3 Baseline of Single Document and Multi-Document Query Based Summarization on Our Dataset

Model No.	Model Name	Zero-Shot Setting (Not Trained on Our Dataset)	Trained on Our Dataset
1	xlm-r-100langs-bert-base-nli-stsb-mean-tokens	18.0	76.0
2	paraphrase-multilingual-mpnet-base-v2	16.0	46.0
3	paraphrase-xlm-r-multilingual-v1	22.0	57.9

Table 6: Baseline - Single-Doc and Multi-Doc QBSUM (Exact Match expressed as a percentage (%))

For both SD-QBSUM and MD-QBSUM, different multilingual SBERT models were used as they were suitable for this task since summarization involves generating summaries with most relevant informations. We used a zero-shot setting here as well and we observed a similar rise in performance on our test set. The first model in table 6 proved to be the best among the three models on which the experiments were carried out.

7 Conclusion and Future Work

In this work, we show pipeline and approach to create datasets for 3 different tasks of NLP - QA, SD-QBSUM and MD-QBSUM, and continue to build them. We plan to work on the shortcomings and unfinished work and address them later. Specially, the completion of the dataset with manual annotation, creating multi-document dataset with proper similarity measurements and outside of wikipedia are some. Additionally, we can extend the work and continue to the task of topic modeling, cross-lingual question answering, and overall summarization.

We hope that our work, now and later when fully completed, will help accelerate QA, QBSUM research in Bangla NLP, and will inspire similar endeavours in other low resource languages as well.

References

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.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. Aquamuse: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint arXiv:2010.12694*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

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.