MLM Is All You Need

Kyumin Park Gyuseok Lee Minseon Gwak Jiwoo Park
Carnegie Mellon University, PA 15213
{kyuminpa, gyuseokl, mgwak, jiwoop}@andrew.cmu.edu

Abstract

In this project, we focus on how domain knowledge influences the quality of question generation (QG) and question answering (QA). For high-level QG and QA tasks, understanding the context around the questions and answers in a document is important. On the other hand, the context not only includes the sentences around the answer span, but it also contains the domain knowledge the document has. Therefore, we utilize domain-adaptive pretraining and task-adaptive pretraining for a model to understand the context. The performances of our QG and QA system could be said to be qualitatively plausible. In addition, by exploring the deeper concept of domain knowledge, we explored whether the knowledge in a specific category can help enhance both QG and QA for our given data set. Through this project, we were able to find out that the difference in performance between the domain adaptation model and the non-domain adaptation model is little.

1 Introduction

Question answering(QA) enables users to receive an answer rapidly and concisely with sufficient context to validate the answer. (Hirschman and Gaizauskas, 2001) In fact, QA systems are largely divided into two types according to the way answers are created.

- **Extractive QA** Extractive QA models extract the answer from a context and provide it directly to the user. It is usually solved with BERT-like models. The strengths of extractive QA models are that the speed to learn the models is fast and that it is easy to include proper nouns in the answer.

- **Generative QA** The model generates free text directly based on the context. It leverages Text generation models. Because generative QA models search for answers based on context, even words or sentences that are not directly specified in the text can be an answer.

On the other hand, question generation (QG) is mainly related to the task of "automatically generating questions from various inputs such as raw text, database, or semantic representation". (Liu et al., 2010)

Since QG and QA are fundamental problems in the natural language processing (NLP) field, they have been studied for a long time, and there have been many approaches to improve their performances.

For instance, QG was utilized to improve QA systems.(Duan et al., 2017) On one side, the QA model judges whether the generated question of a QG model is relevant to the answer. On the other side, the QG model provides the probability of generating a question given the answer, which is useful evidence that in turn facilitates QA.

Moreover, in several papers such as (Choi et al., 2018; Du et al., 2017), it is revealed that understanding the context around the questions and answers in a document is significant. In a slightly broader sense, the context does not mean only the sentences around the answer but also includes the domain knowledge that the document has.

Domain-adaptive pretraining and task-adaptive pretraining are key ideas for a model to understand the broad context. By exploring the concept of domain knowledge, we investigate whether the knowledge in a specific category can help enhance both QG and QA for our given data set. Moreover, we examine that cross-domain adaptation methods would be also helpful to generate questions or answers to the questions.

2 System Architecture

2.1 Relationship between QG and QA

Since we look into the influence of domain adaptation on both tasks, we can test sharing the weights

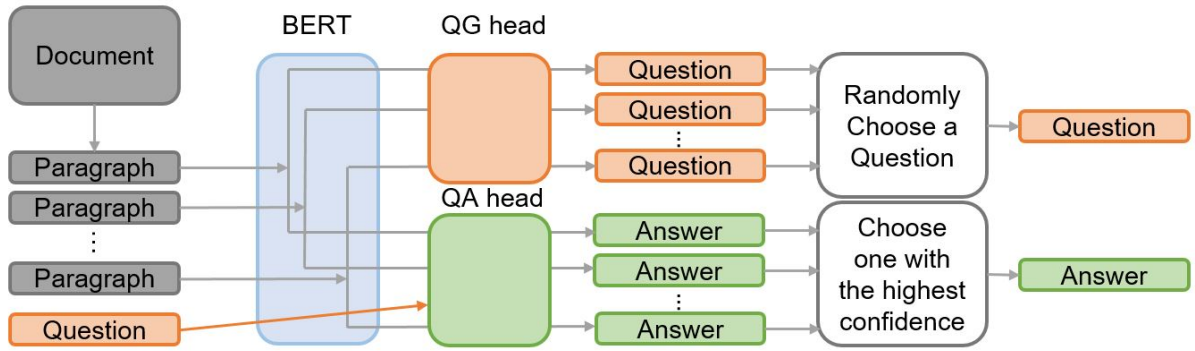

Figure 1: The overall structure of QG and QA systems

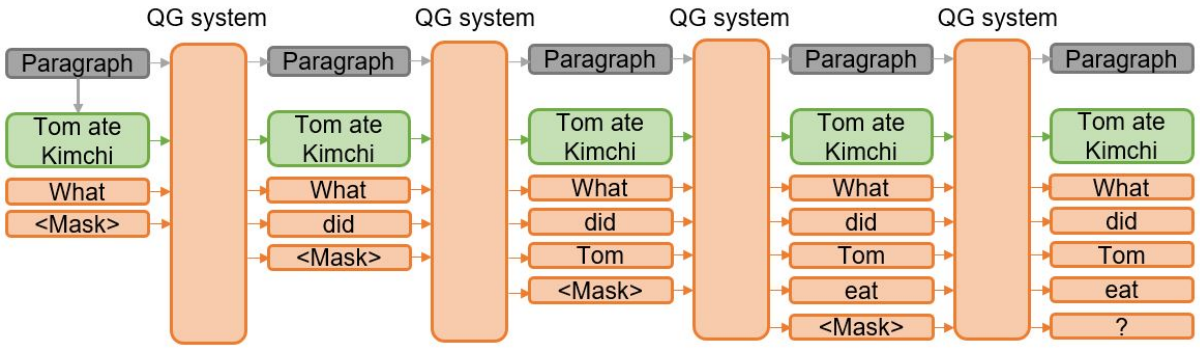

Figure 2: Flow of our QG System Structure

across the QG system and QA system. Using almost similar architecture with different heads, we will experiment with how domain adaptation influences two tasks.

Also, we investigate cross-task adaptation. If available, we will experiment to explore whether training for QG brings a positive impact on QA, and vice versa.

2.2 QG System Architecture

In this project, we will use BERT(Devlin et al., 2019) as a backbone model for the required QG system, and Figure 1 is the overall structure of our required QG and QA systems. Note that we are given a whole document as input for the required Question Generation system. We can't use a whole paragraph as input for BERT because the maximum length of input tokens is fixed. Therefore, we divide the input document into several paragraphs, where the number of tokens is less than the maximum length of input size of BERT, and only select paragraphs of some length as the input texts. The reason for this is that we want to extract questions only from paragraphs that have significant meaning. After that, we conduct question generation for each selected paragraph.

To be more specific about question generation, we give an explanation with Figure 2. We randomly choose a keyword in the paragraph as a topic. Moreover, we randomly choose one of the Question words(When, Where, Why, ...) as the initial word for the newly generated question. After that, masks are put on in turn to predict the next word based on the words found so far to complete the question.

2.3 QA System Architecture

We will also use BERT as a backbone model for the required QA system, and Figure 3 is the overall structure of our required QA system. We also divide the input document into several paragraphs. After that, we implement question answering using BERT for each paragraph and the given question. We choose one of the answers with the highest confidence value as the solution for our required QA system. Moreover, we will conduct MLM using BERT to support the easier QA tasks, as shown in Figure

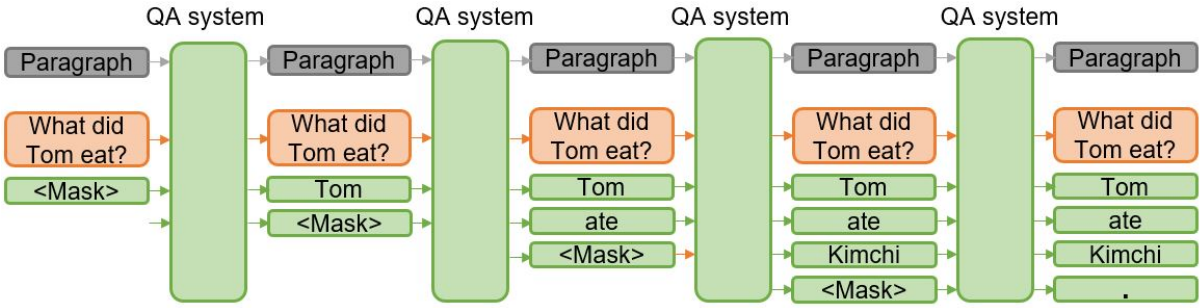

Figure 3: Flow of our QA System Structure

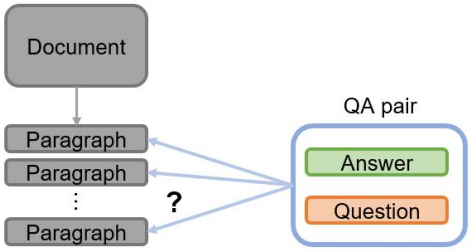

Figure 4: Development data is inappropriate for training

```
Document: /host/Users/data/set1/a1.txt
Q1 Why did he score a match against west ham united?
Q2 Which season did dempsey score in 2010 - 11?
Q3 Were the premier league wins against west ham united?
```

Figure 5: Output of our QG system

```
Q1 Who added 8 more goals in 2006?
A1 Clint dempsey

Q2 Who eventually lost the match on penalties and thus
were eliminated from europe?
A2 Clint dempsey

Q3 Who recorded the then fastest goal in us qualifying
history with a chest trap and sliding shot 53 seconds i
nto an 8-0 defeat of barbados?
A3 Clint dempsey
```

Figure 6: Output of our QA system

3 Experiment

3.1 Usage of development data

Development data are question-answer pairs generated from Wikipedia documents on several topics. The development data consists of a balanced amount of manually-made easy, medium, and hard-level question-answer pairs. However, it might be undesirable to directly use the development data to train a model from scratch. This is because the data are not validated and do not have positional information, which is necessary for extractive QA models. We don't even know which paragraph a given QA pair came from as in Figure 4. Instead, we can use the development data as pretraining data. Since Wikipedia documents have domains in clusters, a subset of development data can be used for domain adaptation. Furthermore, some of the question-answer pairs would be able to be used in task-adaptive pretraining, which will fit weights to the QG/QA tasks.

3.2 Dataset

Wiki QA is a publicly available set of question and sentence pairs, collected and annotated for research on open-domain question answering.(Yang et al., 2015) We used Wiki QA as the main dataset for our required QG/QA systems since all of our development data was extracted from Wikipedia, and WikiQA is the most suitable dataset for the Wikipedia domain. Moreover, we use another dataset SQUAD(Rajpurkar et al., 2018) as an auxiliary dataset to improve our QG/QA systems.

3.3 Results

To evaluate our results, we proceed with a qualitative evaluation of our Question Generation System and Question Answering System. As you can see in Figure 5, our QG system generates natural and non-repeated questions from the given document. This is because our system performs very well when the given input is just a paragraph and we conduct question generation on each paragraph of the given document. On the other hand, the performance of our QA system is slightly inferior. In Figure 6, the actual answers for each question (Q1, Q2, Q3) are Clint Dempsey, Tottenham, and Clint Dempsey, respectively. In other words, our QA system gives us the wrong answers for Question 2 while it gives us the right answers for Questions 1 and 3. In addition, the output answers of the model without domain adaptation are also 2 right answers and 1 wrong answer as you can see in Figure 7.

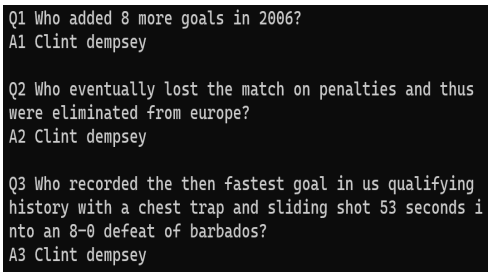

Figure 7: Output of our QA system without domain adaptation

4 Discussion

In this section, we will analyze our results in section 3, and discuss the reason why the application of domain adaptation did not lead to significant changes. First of all, our QG system generates natural and non-repeated questions from the given document. This is because we design our model as a template-based model, which means the generated question always starts with Question words (Why, How, What, ...). On the other hand, the QA task is a harder task than the QG task because the question for the QA system might not come from the given document. Moreover, significant parts that could be the basis for answers might have been removed because short paragraphs were removed. Also, the fact that we trained our model using only paragraphs makes our model hard to generate a proper answer for the whole document. Lastly, we lacked enough high-quality data and did not conduct sufficient experiments due to limitations of time and resources. Moreover, we speculated that no significant difference between our model and the model without domain adaptation would have occurred because the pre-training data was similar to the Wikipedia domain. In summary, we believe that if sufficient high-quality data is secured and enough experiments are added, a model with better performance can be implemented.

5 Conclusions

In this project, we made a Question Generation System and a Question Answering System. The input to our system was the entire document, but the document had too many paragraphs to be suitable as an input for BERT, our backbone model. Therefore, we split the document into several paragraphs and implement question generation/answering for each paragraph. Moreover, we also used masked language mode(MLM) BERT to improve the performance of easy QA tasks. The performances of our QG and QA system could be said to be qualitatively plausible. In addition, we were able to find out that the difference in performance between the domain adaptation model and the non-domain adaptation model is little through this project.

References

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

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.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.

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.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300.

Ming Liu, Rafael A Calvo, and Vasile Rus. 2010. Automatic question generation for literature review writing support. In *International conference on intelligent tutoring systems*, pages 45–54. Springer.

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

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.