Long-Answer Question Generation

Rakhi Batra[*], **Anuja Dixit**[*], **Vivekananda Sathu**[*], **Nikhil Pavan Kanaka**[*]
The Ohio State University
rakhi.1,dixit.89,sathu.1,kanaka.3@osu.edu

Abstract

Question Generation is an important standalone task as well as a valuable intermediate task for training models dealing with human annotated data like question answering. In literature, various models have been designed for automatic question generation, however, generating questions from a long text still remains a challenge. Integrating the methods proposed by (Nie et al., 2023) and (Pan et al., 2021) to collect answers incorporating long-range dependency and to generate question answer (QA) pairs from given context, we have implemented a project utilizing LongFormer attention and (Pan et al., 2021) question generation models to generate long-answer questions from a given academic text. In this report, we discuss the model that was implemented using a pipeline that included span collection, span linking, answer aggregation, and question generation modules. We evaluated the model using a blend of automatic and manual evaluation process, where results indicates that about 42.3% generated questions were contextually and semantically accurate (satisfied both claim 1 and 2). The observation of questions showed that the field of question generation (specially from long text and context specific questions) requires further work to produce semantically accurate results.

1 Introduction and Problem Statement

Natural language processing tasks like question answering need human annotated data for training. Generation of human annotated data is a time consuming process and unavailability or limited availability of the same can affect the quality of the model. Thanks to the recent advancements in machine learning, we can utilize unsupervised learning to generate high quality data for such tasks.

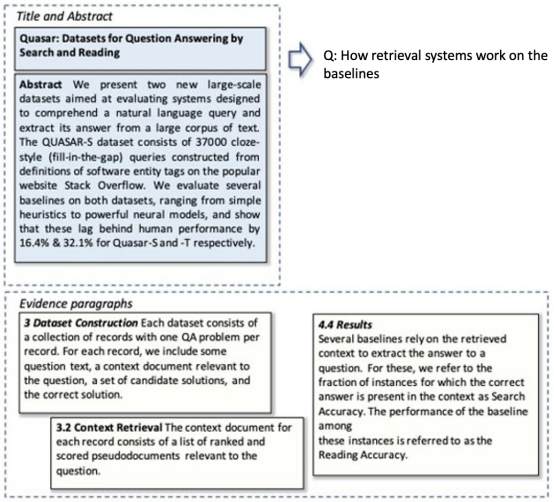

Figure 1: An example instance taken from QASPER.

Question answering, in particular, is an important problem that forms the basis of chat-bot systems and also an important sub-task for training systems that will interact with humans. Human interaction is always unpredictable. Thus, more the training data for question answering, the better it can perform in production. The training data must incorporate different types of questions as well as questions covering all topics of the subject for the trained model to be able to answer any question. That makes data collection/formation a challenge in itself. Unsupervised QA generation can play an important role by generation human-like QA without human intervention.

Question generation methodology depends on the type of question the model is going to generate. For questions like (Rajpurkar et al., 2016) a segment of sentence serves as an answer. These type of questions are single-hop QA that considers information from a single sentence to generate the question. Similarly, there is another type of questions, known as multi-hop questions (Pan et al., 2021), that require combining information from more than one section in large text to generate a QA pair. Example of a long answer question is

[*]All authors contributed equally to this research.

given in Figure 1. For this project, we are referring to such type of questions as "long-answer" question. Thus, long-answer essentially means the answer is formed by combining information from multiple sections making it multi-sentence. Hence in this project we propose following claims:

- Generate long-answer questions from research articles.
- The generated questions are are answerable and complex, thereby facilitating deeper engagement and understanding of the material.

To generate a question in unsupervised way, collecting and forming the answer is a crucial step. The quality of question depends upon the aggregated answer. For questions that can be answered in one sentence, answer generation is easier as compared to multi-hop question. This project focuses on aggregating an answer from various sections of long text and passing it to a question generation module to get a question corresponding to the assembled long-answer.

As mentioned in (Nie et al., 2023), methodology consists of identifying candidate spans, linking related spans and aggregating the pieces to form a coherent and single answer. Some of the challenges include identifying spans that actually contain important information, filtering the collected spans to remove redundancy and rephrasing the collected spans while maintaining semantics and meaning. In this project, we are trying to address these challenges based on the methodology mentioned above.

We have used Qasper dataset to implement the proposed method (Dasigi et al., 2021a). This dataset consists of natural language processing (NLP) related 1585 research papers along with 5049 information-seeking question-answer pairs, where the questions are asked by regular readers of NLP papers and answered by a separate set of NLP practitioners. To define the "long-answer" in our context, we analyzed the answers from the Qasper dataset. In this dataset, the answers are of two types: 1) extracted spans (where answer is exactly from the text, similar words) and 2) free form answers (synthesized from the text to answer the questions). We are interested in free form answers and corresponding questions because they are suitable for the scope of this work. The average length of free form answers is 15.7 words. Thus, we will consider a question as long-answer question if its answer has multiple sentences and total word length is greater than fifteen.

To summarize, in this project, we are dealing with generating long-answer questions from the given academic text. Among the multiple challenges, we will mainly focus on the following experiments in this project:

1. Using a model with more number of attention heads to increase the number of relationship captured between neighbouring tokens while calculating attention scores.
2. Experimenting with the threshold value for attention score to filter the spans solely based on the attention scores. Creating a filter that does not miss important information with minimum to no redundancy.

2 Previous work

Creating Long-Answer questions is a process that entails pinpointing questions that require detailed and extensive responses. This involves recognizing underlying themes or subtopics that closely align with a specific question or concept.

Initial research in this field aimed to create questions by constructing a semantic representation of random text (Olney et al., 2012) in order to analyze the concepts within the text. In the case of (Labutov et al., 2015), the approach initially transformed the original text into a compact ontology, then gathered potential high-level question templates through crowd sourcing, and finally extracted relevant templates for a new section of text. In (Pan et al., 2020), semantic graphs were introduced to improve the representations of input documents, and the approach involved simultaneous training for content selection and question generation. Semantic Templates for Generating Long-From Technical Questions (Pal et al., 2021) proposes using semantic templates to generate questions from technical texts to generate semantically valid questions that require long answers which are more than 5 sentences.

According to (Zhang et al., 2021) Question generation can be done using multiple methodologies like Rule-based methods like template-based, syntax-based, semantic-based approaches or Neural network-based approaches which function by learning the pattern of question generation from dataset.

3 Methodology

Our approach is based on the techniques mentioned in (Nie et al., 2023). It consists of creating a long answer by collecting information spans from various sections of the paper and generating questions based on collected answers. There are four modules in the method: Span Collection, Span Linking, Answer Aggregation and Question Generation. Details of each modules are discussed in the subsequent subsections.

3.1 Span Collection

In this module, spans with potential important information are collected. Candidate spans are shortlisted using constituent parsing and those are used to create masked text for further processing. All instances are passed to pre-trained T5 model for reconstruction of the masked text and calculating the reconstruction loss. Reconstruction loss refers to the difference of information in actual text and text predicted by T5 model. Thus, greater reconstruction loss loss suggests that the span contains important information. Spans with high values of reconstruction loss are selected for the next steps of span linking.

(Nie et al., 2023) presents a detailed ablation study related to various components of the models that motivated this project. As per the first ablation study, QA pairs generated from random selection of 32 spans contain noisy information leading to reduced quality QA pairs. To solve this problem, we decided to filter the spans solely based on the attention scores. We increased the threshold value of attention to include only closely related spans and remove the less important ones. This reduces the number of candidate spans in early execution phase itself and eliminate the need to use the filters after finding the linked spans.

3.2 Span Linking

As per (Nie et al., 2023), Span Linking module is further divided into two modules: Span Graph Construction and Attention-based Graph Walking.

Span graph construction module consist of forming a graph for collected spans where nodes of the graph refer to the spans and edge weights represented by global and local attention between tokens of the spans. (For local and global attention, please refer to the appendix). If there is local attention between two tokens of spans S1 and S2, then there is a direct edge between those two spans.

For calculating global attention, </s> token is appended before each paragraph. The token is represented using K highest attention scores of the span. Then, L top attention scores between </s> token of two paragraphs is considered and M highest attention scores from the representation of second </s> token are considered. The global attention between those two spans is calculated with the help of K and M local attention scores between tokens of spans as well as the attention score between the two </s> tokens. This graph is then passed to the next module.

During the first pass of the execution, only local attention is considered for graph construction and generated QA pairs are used to fine tune the LED model. The fine tuned LED encoder is used to calculate the local and global attention scores during the second pass and final questions are generated. The two pass scheme helps the model in better understanding the relationship between tokens and adjusting the attention mechanism to generate more accurate questions.

The importance of global attention is also presented in an ablation study in (Nie et al., 2023). Since the root of global attention score calculation lies in local attention scores, we decided to work on improving the local attention scores by increasing the number of attention heads that allows the model to capture multiple facets of inter-token relationship potentially improving the local attention scores. Enhanced local attention scores combined with fine tuning in second pass generates better global attention scores leading to linking related spans more accurately.

Attention-based graph walking module plays an important role in successful execution of the methodology by pruning the graph with predetermined threshold for attention scores (edge weights) and linking the spans that are most related to each other. To get the linked spans, we performed Depth-first-search (DFS) on the graph and spans that are in a single connected component are considered to be linked. We experimented with Breadth First Search (BFS) for graph walking to see what difference does level-wise traversing makes.

3.3 Answer Aggregation

Linked spans collected in previous steps are reconstructed using BART to form a long-answer. Basically, this step takes multiple linked spans, combines them into a single text by reconstructing the

text to form a semantically sensible paragraph.

3.4 Question Generation

Unsupervised Multi-hop Question Answering by Question Generation proposes a framework for question answering that has question generation as an intermediate stage. We are utilizing the question generation model used by (Pan et al., 2021) to generate question based on the answer aggregated in previous steps. Some limitations introduced by BART rephrasing are discussed in section 6.

3.5 Two-Pass Scheme for Long-Range Reasoning

In the pre-trained LED model, the matrices for global attention (query, key, and value) are initially replicated from those in local attention. We introduce a two-phase process to enhance long-range reasoning in creating long-document QA pairs. The first phase employs only local attention in the Span Graph Constructor and generates QA pairs. Subsequently, the LED model is fine-tuned on these QA pairs with both local and global attention (details in Appendix A.2), aimed at enhancing the matrices, particularly for global attention. In the second phase, the fine-tuned LED model utilizes both attentions to form the span graph for attention walking, thereby integrating additional global attention insights into the final QA pairs.

4 Execution details

4.1 OSC

Pipeline proposed includes multiple models each responsible for individual tasks, to enable faster execution of the overall pipeline we opted to load all the models to the GPU at once and parallel process the pipeline using multi-threading on a single node. To run this setup we utilised Ascend Cluster on Ohio Super Computer with NVIDIA A100 80GB GPU. Dependency management for the project was taken care by conda environment management with all dependencies listed in a requirement.txt file.

4.2 Faced challenges

In the span graph construction phase, there is an edge between two spans if there is local attention between two tokens of different spans. This stood as a major challenge for our experiment of filtering spans based only on attention scores. When we wanted to get linked spans by increasing the threshold value, we get a lot of single spans i.e single nodes since the edges were getting dropped because of high threshold. That reduced the overall quality of the results. We tried to improve the attention scores as mentioned in previous section, but both experiments combined gave us results similar to the baseline.

Default filtration in baseline consists of randomly selecting 32 spans became a hurdle while assessing the results generated by the experiment of improving attention scores. Since the selection was random, pinpointing the exact measure of effects was challenging.

5 Evaluation

Evaluating the quality of questions generated by long-form question generators is challenging (Mulla and Gharpure, 2023). One of the main challenges is that a question that is suitable for one purpose may not be suitable for another. For example, a question that stimulates new insights in a research project may not engage students in a classroom setting. Another challenge is the difficulty in measuring question quality automatically (Xu et al., 2023). Certain aspects, such as the question's difficulty or its alignment with the curriculum, are challenging to quantify. Nevertheless, there are methods to evaluate long-form question generators. For instance, (Su et al., 2022) proposed a method by comparing the generated questions with a set of human-written questions using several metrics, such as the number of shared concepts, question length, and coherence. Moreover, (Xu et al., 2023) suggested evaluating them using a machine-learning model trained on a dataset of human-written questions and answers, with the model's accuracy and its correlation with human judgments as evaluation metrics. By combining human evaluation, automated evaluation, and task-specific evaluation, we can gain a comprehensive understanding of the quality of questions generated by long-form question generators. This approach can help measure question validity. Potential answers generated in the process can be used to assess the effectiveness of questions in eliciting long and varied answers across different parts of the source text.

Using simple automatic and manual evaluation methods, we evaluated our model's results based on the claims of this work:

Claim 1: Generate long-answer questions from research articles. To evaluate this claim we considered a question long if the length of its answer is

greater than 15 words.

Claim 2: The long-answer questions generated are answerable and complex. To evaluate this claim we considered a question complex if the answer is long and there are multiple number of evidence, meaning the answer spans over multiple paragraphs. Moreover, the answerability of the question was indicated by the presence of evidence from the given text.

Further, to evaluate the quality we selected a sample of questions and manually checked their quality by evaluating the semantics of the questions, the required long answer, and the context within the paper.

By using the proposed model, we generated questions for 163 papers from Qasper test dataset. We observed that on average 26 questions were generated per paper. On average 23 were long whereas among long questions only 12 were complex. For manual evaluation, we selected 300 questions that were long and complex. Out of 300, only 127 questions were semantically correct and were generated within the context of provided text (See table 1). Table 2 presents a few generated questions that were long, complex, and as per quality criteria.

No. questions per paper (Average)	No. long questions (Claim 1)	No. complex questions (Claim 2)	Manual quality evaluation
26	23	12	127

Table 1: Evaluation Statistics

5.1 Observations

In reviewing the outcomes of the automated long-answer question generation model, several patterns emerge in the types of questions generated. One category includes generic questions that, while relevant to the content of the paper, may not be particularly useful for someone aiming to understand the paper in depth. Examples of such questions are "What is BERT?", "What is Word2Vec?", or "What is PyTorch used for?". These questions touch on key concepts or tools mentioned in the paper, but they do not delve into the specific contribution or implications of these concepts within the context of the paper.

Another category comprises vague questions, which lack specificity and clarity, that may make them less effective for in-depth understanding. An example of this is, "What are the opinions of the author of the work?". This question is too broad and does not guide the reader towards a specific aspect or argument presented in the paper.

Question	IsLong	Complex and quality measure
How can we confirm empirically that we have the same magnitude as the theoretical value of 2.15?	Yes. 145 words	The answer explains the whole empirical evaluation process.
How is a fixed-length candidate set dynamically updated?	Yes. 142 words	The answers explain the methods to dynamically update pairs.
How does the use of images help the translation?	Yes. 53 words	The answer describe details along with reference to the figure in the paper.
What have we tried for the image modality?	Yes. 90 words	The answer explains different ways for image modality along with benefit and limitations of each.

Table 2: A Sample of Generated Questions

Additionally, there are non-context specific questions that do not reflect about the paper's actual content. For example, a question like "Who partially supported us to join the conference?" is related more to the acknowledgments rather than the core content of the paper.

Lastly, the model sometimes produces questions with a masking technique, where only the last word of a sentence is omitted. An example of this is, "A significant amount of sentences don't have what?". This approach often results in questions that are too simple or that do not require comprehensive knowledge of the paper's content.

6 Limitations and future work

Semantic drift (Pan et al., 2021) problem is a side effect of incorporating paraphrasing/reconstruction in our methodology. This problem causes shift in the original meaning of the text leading to semanti-

cally inaccurate question generation that introduces noise in the data.

Improved global attention: Currently, global attention between spans in multiple paragraphs is computed using two </s> tokens as a bridge between paragraphs. Although this approach works well in terms of computational costs, it linearizes the attention values into a single value, leading to potential information loss between paragraphs and resulting in poor connections. Further research in this domain is required to ensure better long-range connections.

Better question generation: current Question Generation module lacks contextual awareness and only generates questions based on the provided text. Most of the generated questions tend to be closely related to the last word masked questions. To address this issue, more research is needed to develop methods that incorporate user preferences regarding domain and question type.

7 Conclusion

This project focused on generating questions answered by employing information from multiple sections of the given academic text. With a method based on (Nie et al., 2023), we considered four modules: span collection, span linking, answer aggregation, and question generation. We performed some experiments to improve the local attention scores that help get a better global attention score. We also implemented a better filtration mechanism in span collection and tried multiple graph walking techniques in span linking. Despite the challenges, manual evaluation remained vital for checking the semantic accuracy of the generated questions. Among the 300 questions evaluated manually, only 42.3% were contextually and semantically accurate. The generic nature of the questions showed that the field of question generation requires further work to produce semantically accurate results. With improvements, question generation can be extended to develop particular types questions based on the user.

References

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021a. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021b. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China. Association for Computational Linguistics.

Nikahat Mulla and Prachi Gharpure. 2023. Automatic question generation: A review of methodologies, datasets, evaluation metrics, and applications. *Prog. in Artif. Intell.*, 12(1):1–32.

Yuxiang Nie, Heyan Humuang, Wei Wei, and Xian-Ling Mao. 2023. Attenwalker: Unsupervised long-document question answering via attention-based graph walking.

Andrew Olney, Arthur Graesser, and Natalie Person. 2012. Question generation from concept maps. *Dialogue Discourse*, 3.

Samiran Pal, Avinash Singh, Soham Datta, Sangameshwar Patil, Indrajit Bhattacharya, and Girish Palshikar. 2021. Semantic templates for generating long-form technical questions. In *Text, Speech, and Dialogue*, pages 235–247, Cham. Springer International Publishing.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Unsupervised multi-hop question answering by question generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5866–5880, Online. Association for Computational Linguistics.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stanford University.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for*

Computational Linguistics: ACL 2022, pages 744–756, Dublin, Ireland. Association for Computational Linguistics.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Trans. Inf. Syst.*, 40(1).

A Appendix

A.1 Local and Global attention

Local attention attends to only limited surrounding tokens for given token. The number of neighboring tokens is decided by the window size. **Global attention** considers relationship between all tokens of the text sequence.

A.2 Details in Fine-Tuning the LED Model

Similar to the input setting in (Dasigi et al., 2021b), for a long document, we prepend a special token </s> before each paragraph. And then we send the preprocessed long document into an LED model. For example, assume that there is a long doc- ument: [t1,1,t1,2,...,tp,1,tp,2,...,tP,PN1,tP,PN], where ti,j is the i-th token in paragraph j, P is the number of paragraphs, PN is the number of tokens in paragraph P. After in-serting the special token </s>, the input can be [</s>,t1,1,t1,2,...,</s>,tp,1,tp,2,...,tP,PN1,tP,PN].