SummQG: Question Generation with Summarization

Eileen Cho
New York University
ec3636@nyu.edu

Tin Luu
New York University
tinluu@nyu.edu

Davida Kollmar
New York University
djk519@nyu.edu

Sanat Batra
New York University
sab1086@nyu.edu

Abstract

While multiple neural question generation models succeed in creating well-formed questions, it is more difficult to create well-informed questions. In this project we propose using abstractive summarization on our context passage, and then using GPT-2 to generate questions based on these summaries. We find that our GPT-2 model achieves lower BLEU and ROUGE scores than earlier models. We also find that questions generated from summaries span a larger amount of the passage than questions generated from other methods.

1 Introduction

Building AI models that automatically generate coherent and relevant context-dependent questions would assist us in many ways. Yet, relative to other natural language understanding tasks such as question-answering (QA), summarization, and machine translation, advances in neural-network-based question generation (NQG) are still in their nascent stage. The goal of NQG is to learn to automatically produce questions that are well-formed (i.e. grammatically, syntactically, and semantically correct) and well-informed (i.e. relevant, testing appropriate knowledge, and at the appropriate level of cognition). Recent advances in text generation through transformers such as GPT-2 (Radford et al., 2019) and T5 (Raffel et al., 2019) have allowed generated questions to be well-formed. However, producing well-informed questions remains a challenge (Pan et al., 2019; Tuan et al., 2019).

In this project, we propose a two-stage procedure that leverages summarization to formulate questions. We are motivated by the intuition that good summarization, besides producing a succinct and coherent version of the original text, ought to identify and synthesize the most relevant and salient ideas from this text, forgoing less relevant details.

Passage: Despite his refined appearance and aristocratic bearing, the unknown (Maximilien Longueville) never tells his identity and seems interested in nobody but his sister, a sickly young girl … Several years after her marriage, Émilie discovers that Maximilien is … in fact a Vicomte de Longueville who has become a Peer of France.

BART Summary: Maximilien Longueville never tells his identity and seems interested in nobody but his sister, a sickly young girl.

Generated Question: What is the mystery of Maximilien Longueville?

Figure 1: Question generated using GPT-2 on NarrativeQA summaries. Note that an answer to the question spans more than one sentence of the passage.

Our hypothesis is that conditioning questions on summaries would result in generated questions whose answers span a greater portion of the text passage (in terms of the number of sentences necessary to correctly answer the question). We deem such questions to be high level and harder to generate overall because current popular QA datasets used for question generation contain mainly questions whose answers are confined to a single text sentence. Our two-stage procedure entails using (1) a summarization model off-the-shelf to produce a summary whose sentences act as salient answer candidates (albeit noisy) for an input text, and (2) a text generation model to generate a set of questions given this answer candidate.

Our work begins with evaluating the question-generation capability of GPT-2. We fine-tune and evaluate this transformer on SQuAD 1.1 (Rajpurkar et al., 2016) and NarrativeQA (Kočiskỳ et al., 2018) datasets. The following step involves us using BART (Lewis et al., 2019) to generate summaries from NarrativeQA passages, and using our fine-tuned GPT-2 to generate questions from those. A

sample of a generated question from NarrativeQA is in Figure 1. We find that, in terms of BLEU-4 and ROUGE, GPT-2 does not generate questions as well as previous NQG models. However, compared to other strictly GPT-2-based QG models, our fine-tuned GPT-2 outperforms them in BLEU-4. In addition, we find that generated questions conditioned on summaries do in fact span a larger portion of the passage, but are unsure if our results are significant.

2 Related Works

2.1 Question Generation

The earliest neural-network based approach to question generation was introduced by Du et al. (2017) and uses an attention-based sequence-to-sequence model to generate questions for an input *sentence*. A follow-up work uses sentences from a hierarchical neural sentence-labeling model that takes in a context paragraph and outputs which sentences are question-worthy, based on an extractive summarization methodology (Du and Cardie, 2017). While these models outperform prior rule-based methods, they still struggle with identifying salient and question-worthy ideas or concepts in an input *paragraph*. They propose encoding coreference knowledge into the model (Du and Cardie, 2018).

More recent models have improved on this work. Chan and Fan (2019) suggest using BERT for the entire question generation pipeline, improving the BLEU-4 score on SQuAD over the previous best model by over 5 points. Another model, proposed by Klein and Nabi (2019), uses BERT as text encoder with GPT-2 as question decoder. Our QG method shares a similar approach to one in Puri et al. (2020): identifying answers from text and then using solely GPT-2 to generate questions conditioned on the answers. The difference is that whereas they use BERT to extract answer spans from the SQuAD dataset, we also use NarrativeQA, a dataset more complex than SQuAD, and use BART for abstractive summarization to choose answer spans.

2.2 Summarization

Abstractive summarization emulates the way humans integrate concepts of a given document and generate synthesized sentences based on its salient portions (Narayan et al., 2018; Liu and Lapata, 2019; Yoon et al., 2020). Recent works show that pretrained language models can be successfully applied to the task of abstractive text summarization (Liu and Lapata, 2019; Lewis et al., 2019; Raffel et al., 2019). We use fine-tuned BART as a pretrained model for abstractive summarization, since it is the current state of the art. In this way, we can synthesize salient ideas over the entire text as answers to generate better-informed questions.

3 Data

3.1 SQuAD

As QG models since Du et al. (2017) were trained and evaluated mainly on SQuAD 1.1 (Rajpurkar et al., 2016), we also use this data set for baselining, performance comparison, and experimenting with various data input regimes. SQuAD, as a QA data set, contains question-answer pairs for short context paragraphs. An answer (or answer span) is presented as a sub-string lifted from the sentence containing it, along with an indicator pointing to the location in the context paragraph that the answer starts. A sentence then could have multiple answer spans, thus different questions that ask of the same single sentence. This distinction matters as it influences the structure of input data when we invert SQuAD for generating questions.

3.2 NarrativeQA

Since answers in SQuAD are dominantly factoid (dates, persons, noun phrases, locations, etc.) and confined within a sentence, we consider questions asked of them low-level. We want our GPT-2 to also learn structures of questions that are higher level and more complex, whose answers could span across the context; thus, we leverage NarrativeQA (Kočiskỳ et al., 2018). Known as a dataset that tests deeper reading comprehension, it builds on entire books and movie scripts, Wikipedia summaries of these narratives, and question-answer pairs written on these wiki summaries. The diversity of its questions include "why/reason" and "how/methods", and their answers are not confined. GPT-2 fine-tuned on NarrativeQA, we anticipate, would be able to formulate high-level questions on not just the provided answers, but also the abstractive summaries from BART.

To pre-process both datasets for GPT-2, for a training instance, we delimit it by `|<startoftext>|` and `|<endoftext>|` tokens. Between them, we add `[context]` and `[question]` to indicate the starts of contexts and questions, respectively. Note that the structure

of [context] changes with different training regime as will be described in Section 4.

Specifically to NarrativeQA, to adhere to GPT-2's restriction of at most 1023 tokens per prefix context for text generation, we remove all test examples whose lengths exceed 1023 tokens (tokens from summary + tokens from answer), leaving us with 78% of the original test set.

4 Experiment Setups and Models

4.1 Baseline SQuAD-SQ

For a fair comparison, we use the same processed SQuAD and train/dev/test splits used in Du et al. (2017) for fine-tuning our GPT-2 for QG, leveraging the fine-tuning pipeline built by Woolf[1].

We replicate the training process in Du et al. (2017) by conditioning question generation on only the *sentence* containing the answer span. Thus, sentences are the only [context] GPT-2 has to generate questions, and training input instance has the structure *(sentence, question)*.

Noted in 3.1, for a same sentence, it could have multiple answer spans, thus multiple questions linked to it. Therefore, for evaluation of SQuAD-SQ, on top of evaluating the entire SQuAD test set, we also evaluate on two splits: one consisting of only sentences that have only one answer span each, and another with sentences each linked to multiple questions. The former case is one-to-one question-answer mapping, the latter is one-to-many.

4.2 SQuAD-SQ + answers

We improve the question generation learning for GPT-2 by appending the answer span into the *[context]*, thus a training input instance is now *(sentence+answer, question)*, making all instances effectively one-to-one question-answer mapping. This answer-inclusion method follows the training regime of later NQG models (Kim et al., 2019).

4.3 NarrativeQA-SAQ

Following 4.2, we fine-tune GPT-2 on narrativeQA, where a training input instance is *(summary+answer, sentence)*. Note a difference: a *summary* is consists of multiple sentences, and overall longer than even a SQuAD paragraph. As there is no published result from previous NQG models on NarrativeQA, our work here effectively establishes a QG baseline for this dataset.

[1] https://github.com/minimaxir/gpt-2-simple

4.4 NarrativeQA-SAQ-DL

We attempt to improve QG performance by fine-tuning GPT-2 using a custom data loader that adds right-side padding to make each training instance 1023-token long. To do so, we adapt the Huggingface implementation of GPT-2 (Wolf et al., 2019). The hypothesis is that padding lengthens the distances among training instances, thus minimizing any causal relationship that could be learned across the input instances.

4.5 Generation

For these 4 models, once fine-tuned, we use the corresponding validation sets to determine values for text generation hyperparameters: temperature T = 0.1, top-k = 50, and top top-p = 0.9. We then sample 10 questions for each instance from the corresponding test sets, and pull out the most commonly generated one as the official model-generated question (pick at random if tie). We report the models' BLEU-4 and ROUGE$_L$ scores in Table 1.

4.6 GPT-2+Summarization

To test our main hypothesis that generated questions conditioned on abstractive summarization require a wider span of the text to answer than questions without, we used a human evaluation.

We sampled a passage from Narrative QA test set, then picked 5 random QA pairs on this passage, 4 random sentences extracted from this passage, and all 3 sentences of the BART summary on this passage. The random extracted sentences act as a baseline for testing the effect of BART summaries on generating higher level questions. For each of these as answers, we sampled 10 questions at T=0.5 and the remaining parameters the same as above. In total, there were about 70 unique questions (some questions were made by multiple methods).

Our human evaluators read the passage, then for each question, determine how many sentences from the passage would be required to answer. The evaluators also had the option of indicating that the question was ill-formed or that the answer not in text. We hypothesize that the questions generated from the sentences of BART would require more sentences from the passage to be able to answer.

5 Results and Analysis

5.1 Automatic Evaluation

As shown in Table 1, we find that with respect to SQuAD dataset, fine-tuned GPT-2 achieves 2.36

Model	Dataset	BLEU-4	ROUGE$_L$
(Du et al., 2017) LSTM-based	SQuAD-SQ (all)	12.24	39.75
GPT-2 (our model)	SQuAD-SQ (all)	8.99	34.65
	SQuAD-SQ (one-to-one)	9.45	35.61
	SQuAD-SQ (one-to-many)	8.80	34.29
(Klein and Nabi, 2019)	SQuAD-SQ+answers	7.84	34.51
GPT-2 (our model)	SQuAD-SQ+answers	11.35	31.97
	NarrativeQA-SAQ	9.31	31.34
GPT-2 + custom data loader (our model)	NarrativeQA-SAQ-DL	7.02	27.48

Table 1: BLEU and ROUGE Scores on Questions Generated for our models.

Answer type	Span	Accept Quest.	Reject Quest.
Narr. QA Gold	1.39	6	0
Narr. QA pairs	1.51	9	11
Extracted sentence	1.41	12	18
BART sentence	1.76	7	9

Table 2: Human evaluation results for our model.

points better for BLEU-4 when given the answers in addition to just the context sentence. ROUGE score, however, drops by 2.68. Without having answers as input (SQuAD-SQ all), GPT-2 performs between for one-to-one instances than for one-to-many instances. This shows that having multiple questions to the same sentence without answers as differentiating focal points distracts GPT-2. Thus, joining answer spans to context sentences helps direct the semantic of generated questions.

Our GPT-2-based QG models perform worse than (Du et al., 2017) across the board. However, when compared to (Klein and Nabi, 2019), the most recently published scores for another GPT-2-based QG model, our SQuAD-SQ+answers model outperforms by 3.51 points for BLEU-4 score, a significant margin, and our SQuAD-SQ model outperforms on both BLEU and ROUGE by smaller margins. This begs us to ask whether the poorer performance on GPT-2-based QG model originates from incorporating this transformer itself.

With respect to NarrativeQA, fine-tuned GPT-2 without the experimental custom data loader outperforms one with it by large margins. One reason, we suspect, is GPT-2 as a model does not use a pad token by default and therefore cannot handle it well when added in. A more probable reason is the average length of each training instance without padding is already long (since it consists the long summary text) and close to 1023 tokens, so adding padding tokens does not increase the perceived distances among training instances.

5.2 Human Evaluation

Table 2 presents results from our human evaluation, which asked if questions from abstractive summaries span greater portions of text than those from other methods. We received 21 responses to our survey. We discarded all questions where a majority of evaluators felt they were ill-formed or could not be answered by the passage, and took the average answer span for the remaining questions.

Evaluators did, indeed, indicate that questions generated from BART summaries required larger answer spans than those generated from other methods. However, it is unclear if this is statistically significant, due to our small pool of evaluators.

A high number of questions presented were rejected by our evaluators (NarrativeQA gold questions were ones originally in the dataset). This shows relevance of questions from summaries is no better than that of those from the other methods.

6 Conclusion

In this project, we generated questions using GPT-2 for SQuAD and NarrativeQA datasets. Our BLEU and ROUGE scores were lower than previous results for SQuAD, indicating that GPT-2 generates questions dissimilar to ones in SQuAD. More research is needed to determine if this indicates that the questions generated from GPT-2 are poor and that GPT-2 is unsuitable for question generation, or if they are merely different from, but semantically similar to, the original SQuAD questions.

We find that questions generated with abstractive summaries may require more complex answers than those from other methods. To test this hypothesis more thoroughly, we would needed to conduct our human evaluation on a larger scale.

7 Collaboration Statement and Code

Project Github: github.com/eileencho/SummQG

Tin oversaw project coordination, fine-tuned GPT-2 and generated questions on both SQuAD and narrativeQA, designed baseline experiments and setups, helped with survey design, cleaned data and output, analyzed results, debugged evaluation metrics, assisted in presentation prep, and wrote/edited the paper. Eileen worked on summarization methods and generated summaries, performed random sentence extraction, debugged evaluation pipeline, analyzed results from automatic evaluation, assisted in survey formulation, prepped and co-presented the PPT, and wrote/edited the paper. Davida assisted Sanat with GPT-2 data loader, created and analyzed results for the human evaluation survey, prepped and co-presented presentation, edited/wrote paper. Sanat implemented customized data loader for GPT-2, fine-tuned and generated questions from this version. To be fair to all members, all except Sanat contributed equally and majorly to the making and completion of this project.

References

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.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *Association for Computational Linguistics (ACL)*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Association for Computational Linguistics (ACL)*.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving neural question generation using answer separation. In *AAAI*.

Tassilo Klein and Moin Nabi. 2019. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds.

Tomáš Kočiskỳ, 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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

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

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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 e-prints*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Luu Anh Tuan, Darsh J Shah, and Regina Barzilay. 2019. Capturing greater context for question generation.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Wonjin Yoon, Yoon Sun Yeo, Minbyul Jeong, Bong-Jun Yi, and Jaewoo Kang. 2020. Learning by semantic similarity makes abstractive summarization better.