A Feasibility Study of Answer-Agnostic Question Generation for Education

Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, Dayheon Choi, Chuning Yuan, Chris Callison-Burch
University of Pennsylvania
{ldugan,elenimi,shriyash,etangins,hannahgl,dhachoi,dianacny,ccb}@seas.upenn.edu

Abstract

We conduct a feasibility study into the applicability of answer-agnostic question generation models to textbook passages. We show that a significant portion of errors in such systems arise from asking irrelevant or uninterpretable questions and that such errors can be ameliorated by providing summarized input. We find that giving these models human-written summaries instead of the original text results in a significant increase in acceptability of generated questions (33% → 83%) as determined by expert annotators. We also find that, in the absence of human-written summaries, automatic summarization can serve as a good middle ground.

1 Introduction

Writing good questions that target salient concepts is difficult and time consuming. Automatic Question Generation (QG) is a powerful tool that could be used to significantly lessen the amount of time it takes to write such questions. A QG system that automatically generates relevant questions from textbook would help professors write quizzes faster and help students stay engaged when reviewing course material.

Previous work on QG has focused primarily on answer-aware QG models. These models require the explicit selection of an answer span in the input context, typically through the usage of highlight tokens. This adds significant overhead to the question generation process and is undesirable in cases where clear lists of salient key terms are unavailable. We conduct a feasibility study on the application of answer-agnostic question generation models (ones which do not require manual selection of answer spans) to an educational context. Our contributions are as follows:

- We show that the primary way answer-agnostic QG models fail is by generating irrelevant or uninterpretable questions.
- We show that giving answer-agnostic QG models human-written summaries instead of the original text results in significant increases in question acceptability (33% → 83%), relevance (61% → 95%), and in-context interpretability (56% → 94%).
- We show that, in absence of human-written summaries, providing automatically generated summaries as input is a good alternative.

2 Related Work & Background

Early attempts to use QG for educational applications involved generating gap-fill or “cloze” questions (Taylor, 1953) from textbooks (Agarwal and Mannem, 2011). This procedure has been shown to be effective in classroom settings (Zavala and Mendoza, 2018) and students’ scores on this style...
of generated question correlate positively with their scores on human-written questions (Guo et al., 2016). However, there are many situations where gap-fill questions are not effective, as they are only able to ask about specific unambiguous key terms.

In recent years, with the advent of large crowd-sourced datasets for extractive question answering (QA) such as SQuAD (Rajpurkar et al., 2018), neural models have become the primary methods of choice for generating traditional interrogative style questions (Kurdi et al., 2019). A common task formulation for neural QG is to phrase the task as answer-aware, that is, given a context passage $C = \{c_0, ..., c_n\}$ and an answer span within this context $A = \{c_k, ..., c_{k+l}\}$, train a model to maximize $P(Q|A, C)$ where $Q = \{q_0, ..., q_m\}$ are the tokens in the question. These models are typically evaluated using n-gram overlap metrics such as BLEU/ROUGE/METEOR (Papineni et al., 2002; Lin, 2004; Banerjee and Lavie, 2005) with the reference being the original human-authored question as provided by the extractive QA dataset.

The feasibility of using answer-aware neural QG in an educational setting was investigated by Wang et al. (2018), who used a BiLSTM encoder (Zhang et al., 2015) to encode $C$ and $A$ and a unidirectional LSTM decoder to generate $Q$. They trained on the SQuAD dataset (Rajpurkar et al., 2018) and evaluated on textbooks from various domains (history, biology, etc.). They showed that generated questions were largely grammatical, relevant, and had high n-gram overlap with human-authored questions. However, given that we may not always have a list of key terms to use as answer spans for an input passage, there is a desire to move past answer-aware QG models and evaluate the feasibility of answer-agnostic models for use in education.

Shifting to answer-agnostic models creates new challenges. As Vanderwende (2008) claims, the task of deciding what is and is not important is, itself, an important task. Without manually selected answer spans to guide it, an answer-agnostic model must itself decide what is and is not important enough to ask a question about. This is typically done by separately modeling $P(A|C)$, i.e., which spans in the input context are most likely to be used as answer targets for questions. The extracted answer spans are then given to an answer-aware QG model $P(Q|A, C)$. This modeling choice allows for more controllable QG and more direct modeling of term salience.

Previous work done by Subramanian et al. (2018) trained a BiLSTM Pointer Network (Vinyals et al., 2015) for this answer extraction task and showed that it outperformed an entity-based baseline when predicting answer spans from SQuAD passages. However, their human evaluation centered around question correctness and fluency rather than relevance of answer selection. Similar follow-up studies also fail to explicitly ask annotators whether or not the extracted answers, and subsequent generated questions, were relevant to the broader topic of the context passage (Wang et al., 2019; Cui et al., 2021; Wang et al., 2019; Du and Cardie, 2018; Alberti et al., 2019; Back et al., 2021).

In our study, we explicitly ask annotators to determine whether or not a generated question is relevant to the topic of the textbook chapter from which it is generated. In addition, we show that models trained for answer extraction on SQuAD frequently select irrelevant or ambiguous answers when applied to textbook material. We show that summaries of input passages can be used instead of the original text to aid in the modeling of topic salience and that questions generated from human-written and automatically-generated summaries are more relevant, interpretable, and acceptable.

3 Methodology

To perform answer-agnostic QG, we follow work done by Dong et al. (2019) and Bao et al. (2020) who show that language models, when fine-tuned for both QA and QG, perform better than models tuned for only one of those tasks. We assume that answer extraction will aid both QA and QG and thus use a model that was fine-tuned on all three. We considered using UniLM (Bao et al., 2020) or ProphetNet (Qi et al., 2020) but ultimately chose a T5 language model (Raffel et al., 2020) fine-tuned...
### Table 1: Analysis of summaries written by our three RAs. Key-Term Coverage is percentage of bolded textbook key terms present in the summary. Average sentence length reported in tokens (space-delimited).

<table>
<thead>
<tr>
<th>Summary</th>
<th>Key-Term Coverage ($\S$)</th>
<th>Total # Sents</th>
<th>Avg. Sent Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1’s Summary</td>
<td>77.6%</td>
<td>279</td>
<td>17.56</td>
</tr>
<tr>
<td>A2’s Summary</td>
<td>80.7%</td>
<td>243</td>
<td>19.28</td>
</tr>
<tr>
<td>A3’s Summary</td>
<td>53.4%</td>
<td>148</td>
<td>15.37</td>
</tr>
</tbody>
</table>

3 The three fine-tuning tasks that were used to train the model we used are illustrated in Figure 2. For question generation, the model is trained to perform answer-aware question generation by modeling $P(Q|A,C)$. For question answering, the model is trained to perform extractive QA by modeling $P(A|C,Q)$. Finally, for answer extraction, instead of modeling $P(A|C)$, the model is trained to model $P(A|C')$ with $C' = \{c_0, ..., c_s, ..., c_e, ..., c_{n+2}\}$ where $c_s$ and $c_e$ are highlight tokens that denote the start and end of the sentence within which we want to extract an answer span.

4 To generate questions, we iteratively highlight the start and end of each sentence in a given passage and extract at most one answer span per sentence. We then generate one question per extracted answer span using the same model in an answer-aware fashion. Passages longer than 512 tokens are split such that no sentences are divided between sub-passages and all sub-passages have a roughly equal number of sentences.

### 4 Experiments

Our first experiment evaluates the performance of the model on the original text extracted from Jurafsky and Martin (2020)’s textbook “Speech and Language Processing 3rd Edition.” To ensure proper comparison, we manually extracted the text from our three chapters of interest (Chapters 2, 3, and 4). When extracting text, all figures, tables, and equations were omitted and all references to them were either replaced with appropriate parenthetical citations or removed when possible. In total, we generated 1208 question-answer pairs from the original text.

Our second experiment evaluates the performance of the model on human-written summaries. We recruited three research assistants (RAs) as part of an undergraduate research experience to write abstractive summaries for each subsection of the same three chapters of the textbook. They were encouraged to make their summaries easily readable by humans rather than to be easily understandable by machines but otherwise no specific guidelines were given. We report some statistics about these summaries in Table 1 and include examples in Appendix E. From these three sets of summaries we generated a total of 667 question-answer pairs.

Our final experiment evaluates the performance of the model on automatically generated summaries. To perform this automatic summarization we used a BART (Lewis et al., 2020) language model which was fine-tuned for summarization on the CNN/DailyMail dataset (Nallapati et al., 2016). The same chunking procedure as described in Section 3 was performed on input passages that were larger than 512 tokens. The summarized output sub-passages were then concatenated together before running question generation. In total, we generated 318 question-answer pairs from our automatic summaries.

### 5 Evaluation

For evaluation, we randomly sampled 100 question-answer pairs from each of the three experiments to construct our evaluation set of 300 questions. We tasked the same set of RAs to evaluate the quality of the question-answer pairs. All 300 pairs were given to all three annotators. We asked the following yes/no questions:

(i). (Acceptable) Would you directly use this question as a flashcard?

(ii). (Grammatical) Is this question grammatical?

(iii). (Interpretable) Does the question make sense out of context?

(iv). (Relevant) Is this question relevant?

(v). (Correct) Is the answer correct?

We provided many example annotations to our annotators and wrote clear guidelines about each category.

---

3https://huggingface.co/valhalla/t5-base-qa-qg-hl
4This comes from a limitation of the answer extraction model. The model is highly likely to extract the same answer span when run on a sentence multiple times. Future work should seek to improve this weakness. There are many cases where asking multiple questions on one sentence is desirable.
5https://web.stanford.edu/jurafsky/slp3/
6RAs were compensated with inclusion as co-authors
7https://huggingface.co/facebook/bart-large-cnn
In Figure 3 we report the results of our evaluation across the three sources. We note that a majority of observed errors in the original text questions stem from them being either irrelevant or uninterpretable out of context. We also see that generating questions directly from human-written summaries significantly improves relevance and interpretability, resulting in over 80% being labeled as acceptable by annotators. Finally, in the case of automatic summaries, we see that relevance and interpretability are improved as compared to the original text questions while grammaticality suffers.

In Table 2 we evaluate the coverage of our generated questions. Coverage was calculated by extracting the bolded key terms from the textbook chapters and sub-string searching for each term among all questions and answers from a given source. Interestingly, if we think of the results from Figure 3 as precision scores and Table 2 as recall, we can see that human summaries have high precision high recall, original text has low precision high recall, and automatic summaries strike a balance between the two.

<table>
<thead>
<tr>
<th>Source</th>
<th>n</th>
<th>Qs</th>
<th>As</th>
<th>Qs or As</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Text</td>
<td>1209</td>
<td>70.9%</td>
<td>70.3%</td>
<td>88.6%</td>
</tr>
<tr>
<td>Auto Summary</td>
<td>318</td>
<td>44.9%</td>
<td>43.0%</td>
<td>60.1%</td>
</tr>
<tr>
<td>Human Summary</td>
<td>667</td>
<td>63.9%</td>
<td>68.4%</td>
<td>86.1%</td>
</tr>
</tbody>
</table>

Table 2: Coverage of bolded key terms from the textbook. Numbers represent percentage of bolded key terms present in any of the n question/answer pairs selected from the given source.

In Table 3 we report the pairwise inter-annotator agreement (IAA) as well as a per-annotator scoring breakdown. We use pairwise Cohen $\kappa$ instead of Fleiss $\kappa$ to better highlight the difference in agreement between certain pairs of annotators. While at first glance it may seem that agreement is low for grammaticality and correctness, this is somewhat expected for highly unbalanced classes (Artstein and Poesio, 2008). For the other three categories (relevance, interpretability, acceptability) we see pairwise agreement of approximately 0.4, suggesting a fair degree of agreement for such seemingly ambiguous categories.

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>Pairwise IAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptable</td>
<td>69.7</td>
<td>48.7</td>
<td>47.7</td>
<td>(0.41, 0.50, 0.33)</td>
</tr>
<tr>
<td>Grammatical</td>
<td>98.3</td>
<td>90.7</td>
<td>86.3</td>
<td>(0.16, 0.49, 0.10)</td>
</tr>
<tr>
<td>Interpretable</td>
<td>79.7</td>
<td>70.7</td>
<td>59.7</td>
<td>(0.51, 0.43, 0.32)</td>
</tr>
<tr>
<td>Relevant</td>
<td>79.0</td>
<td>71.3</td>
<td>69.0</td>
<td>(0.41, 0.29, 0.25)</td>
</tr>
<tr>
<td>Correct</td>
<td>91.7</td>
<td>90.7</td>
<td>90.0</td>
<td>(0.03, 0.08, 0.06)</td>
</tr>
</tbody>
</table>

Table 3: Comparison between our three annotators (A1, A2, A3) on all 300 questions across all categories. Numbers represent percentages of “Yes” answers. Pairwise Inter-Annotator Agreement is calculated by Cohen $\kappa$ and is reported in the order (A1-A2, A2-A3, A3-A1).

6 Conclusion and Future Work

In this work we show that answer-agnostic QG models have difficulty both choosing relevant topics to ask about and generating questions that are interpretable out of context. We show that asking questions on summarized text ameliorates this in large part and that these gains can be approximated by the use of automatic summarization.

Future work should seek to further explore the relationship between summarization and QG. Work done concurrently to ours by Lyu et al. (2021) already has promising results in this direction, showing that training a QG model on synthetic data from summarized text improves performance on downstream QA.

Additionally, future work should focus on further refining and standardizing the metrics used for both automatic and human evaluation of QG. As noted by Nema and Khapra (2018) n-gram overlap metrics correlate poorly with in-context interpretability and evaluation on downstream QA fails to address the relevance of generated questions.

8 Examples of questions for each category on which there was significant disagreement are listed in Appendix D.
Acknowledgements

We graciously thank Suraj Patil for providing the fine-tuned question generation model used in this project. His training and inference code provided a great starting point for our experiments. We’re very grateful for his support.

We would also like to thank Prof. Dan Jurafsky and Prof. James Martin for providing us with the raw latex files for their textbook. These files were very helpful for extraction purposes and saved us a lot of time.

Finally, we would like to thank the members of our lab for suggestions and feedback. In particular, Dan Deutsch and Alyssa Hwang were particularly influential in shaping the current version of this paper. Their great suggestions made the writing much clearer and much more understandable.

References


Daniel Jurafsky and James H Martin. 2020. Speech and language processing 3rd edition draft. Prentice Hall NJ.


Siyuan Wang, Zhongyu Wei, Zhihao Fan, Yang Liu, and Xuanjing Huang. 2019. A multi-agent communication framework for question-worthy phrase extraction and question generation. In AAAI.


A Software and Data

The code and data used in this project can be found in our project repository. The repository houses the 300 annotated questions, the 2,194 un-annotated questions, the text sources used (three chapters of cleaned text from Jurafsky and Martin, three sets of human summaries, one set of automatic summaries), and the code used to generate the questions. We also provide scripts to reproduce the coverage analysis as well as the analysis of our annotations.

B Annotator Guidelines

In Table 4, we report the annotation guidelines given to our annotators. In the original document, under each category, 3 or more example annotations were given, each containing an explanation as to why the selection was made. Categories such as grammaticality had 10 or more examples given to ensure maximum agreement between annotators. Several discussion sessions were held between the authors and annotators to ensure that the guidelines were well understood.

During annotation, annotators were given the original textbook chapters to use as reference material and were allowed to use online search engines to check for grammaticality and correctness.

9https://github.com/liamdugan/summary-qg
Would you directly use this question as a flashcard? (Yes / No):
A Yes answer to this question means that the generated question is salient, grammatically correct, non-awkwardly phrased and has one correct answer. If you answer Yes to this question you may skip the rest of the annotation for the given example – the answers for all other questions are assumed to be Yes. If you answer No, then please continue on to the rest of the questions. Importantly, if you *did* answer yes to all of the other questions, do not feel pressured to answer yes to this question. There are many reasons why you might not want to directly use a question as a flashcard (too easy, too general, etc.) that are not enumerated here.

Is this question grammatically correct? (Yes / No):
A Yes answer to this question implies that a question has no grammatical errors. Awkwardly worded questions that are grammatical should be annotated as such (answer Yes for these questions).

Does this question make sense out of context? (Yes / No):
This question asks if there are any references made by the question to other items that have been “previously discussed”. For our use case, questions should never refer to other specific items in the text from which they were drawn. A Yes answer to this implies that the question is interpretable when taken on its own and is a question that someone would ask if there was no pre-existing context.

Is this question relevant? (Yes / No):
A Yes answer to this question implies that the question being asked is important for understanding the main points that the chapter (and by extension the book) is attempting to teach. Questions that are relevant should be ones that would plausibly be asked on a quiz or a test from a fairly thorough course on computational linguistics. Questions that are about insignificant details or questions that are about specific illustrated examples that are not useful for understanding the main points of the chapter should be given a No. Anything that is relevant (or tangentially relevant) to computational linguistics should be given a Yes.

Is the answer to the question correct? (Yes / No):
A Yes answer to this question implies that the answer given is one of a multitude of plausible correct answers to the question. If the question has multiple correct answers and the given answer is one of them, it should be annotated as a Yes. If the question is bad/grammatical or underspecified to such an extent that you cannot judge the answer properly, you should annotate Yes. However, irrelevant questions that are grammatical and reasonably interpretable should be annotated properly.

Table 4: Guidelines given to our human annotators before annotating for the acceptability, grammaticality, interpretability, relevance, and correctness of generated questions.

<table>
<thead>
<tr>
<th># Questions</th>
<th>Chapter 2 (n = 139)</th>
<th>Chapter 3 (n = 93)</th>
<th>Chapter 4 (n = 66)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceptable</td>
<td>54.0%</td>
<td>58.1%</td>
<td>53.0%</td>
</tr>
<tr>
<td>Grammatical</td>
<td>94.2%</td>
<td>93.5%</td>
<td>93.9%</td>
</tr>
<tr>
<td>Interpretable</td>
<td>74.1%</td>
<td>76.3%</td>
<td>72.7%</td>
</tr>
<tr>
<td>Relevant</td>
<td>72.7%</td>
<td>81.7%</td>
<td>83.3%</td>
</tr>
<tr>
<td>Correct</td>
<td>95.0%</td>
<td>100%</td>
<td>98.5%</td>
</tr>
</tbody>
</table>

Table 5: Distribution of human evaluation scores across the three chapters of annotation. Labels are determined via majority vote among our three annotators.

C Comparison Across Chapters

In Table 5 we report the distribution of scores across chapters. We note that scores are largely consistent across the three chapters, with lower average relevance for Chapter 2 questions possibly owing to the source material containing many worked examples of regular expressions.

D Example Disagreements

In Table 6, we list questions for which there was at least one dissenting annotator. We see that for categories such as “Relevant” and “Interpretable”, annotations are often dependent on the level of granularity with which the topic is being discussed. For example, a question such as “Who named the minimum edit distance algorithm?” may or may not be relevant depending on how granular of a class the student is taking.

For categories such as “Correct” or “Acceptable” certain particularities about otherwise good questions can easily disqualify them from receiving a positive annotation. In the case of “What NLP algorithms require algorithms for word segmentation?”, keen-eyed annotators would notice that the question is non-sensical, however others may note that both Japanese and Thai do, in fact, require word segmentation. Particularities such as these make this task very difficult, even for expert annotators.

E Example Summaries

In Table 7 we list two examples of textbook sections with their accompanying human and automatic summaries. We see that length of summary varies drastically between our annotators, each of them making different decisions on whether or not to keep or discard certain pieces of information. We also note that automatic summaries are much more extractive in nature while human summaries are generally more abstractive.
Table 6: Questions for which there was disagreement on the label for the given category

<table>
<thead>
<tr>
<th>Acceptable</th>
<th>Q: What is another name for a corpus that NLP algorithms learn from? A: training corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grammatical</td>
<td>Q: What are words like uh and um called fillers? A: filled pauses</td>
</tr>
<tr>
<td>Interpretable</td>
<td>Q: What context do words that are in our vocabulary appear in a test set in? A: unseen</td>
</tr>
<tr>
<td>Relevant</td>
<td>Q: What word has the same lemma cat but are different wordforms? A: cats</td>
</tr>
<tr>
<td>Correct</td>
<td>Q: What is another important step in text processing? A: Sentence segmentation</td>
</tr>
</tbody>
</table>

Human Summary (A1): We remove some probability mass for more frequent events and reassign it to unseen events with known words, and this is called smoothing or discounting. We study four 4 main methods of smoothing: Laplace smoothing, add-k smoothing, stupid backoff, and Kneser-Ney smoothing.

Human Summary (A2): Smoothing or discounting is the procedure of transferring the probability mass of frequent events to other words that appear in the test set in an unseen context.

Human Summary (A3): Not assigning zero to the probability of an unseen word in the test set is called smoothing or discounting. There are different ways to do smoothing: Laplace, add-k smoothing, stupid backoff, Kneser-Ney smoothing.

Table 7: Examples of human and automatic summaries for two sections of “Speech and Language Processing”. The left text is from Section 3.4 “Smoothing” and the right text is from Section 4.6 “Naive Bayes as a Language Model”. We see that the automatic summaries tend to be more extractive while the human summaries are more abstractive.