A Feasibility Study of Answer-Unaware Question Generation for Education

Anonymous ACL submission

Abstract

We conduct a feasibility study into the applicability of answer-unaware question generation models to textbook passages. We show that a significant portion of errors in such systems arise from asking irrelevant or un-interpretable 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 textbooks would help professors write quizzes faster and help students spend more time reviewing flashcards rather than writing them.

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-unaware 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-unaware QG models fail is by generating irrelevant or un-interpretable questions.

• We show that giving answer-unaware QG models human-written summaries instead of the original text results in a significant increase in acceptability of generated questions (33% -> 83%).

• 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 crowdsourced 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 QA 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 uni-directional LSTM decoder to generate \( Q \). They trained on the SQuAD dataset (Rajpurkar et al., 2018) and evaluated on textbooks from various domains (history, sociology, biology). 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 convenient 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-unaware models for use in education.

Shifting to answer-unaware 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-unaware model must itself decide what is and is not important enough to ask a question about. Previous work primarily accomplishes this 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 human annotators whether or not the extracted answers, and subsequent generated questions, were relevant to the broader topic of the context passage (Willis 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-unaware 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 on SQuAD due to the clean separation between...
We perform this splitting such that no sentences are present in the highlighted sentence, the answer is discarded.

The three fine-tuning tasks that were used to train our model 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 directly modeling \( P(A|C) \), a new context \( C' = \{c_0, ..., c_s, ..., c_e, ..., c_{n+2}\} \) is generated where \( c_s \) and \( c_e \) are highlight tokens that denote the start \( (s) \) and end \( (e) \) of the sub-sequence within which we want to extract an answer span. The answer extraction fine-tuning task thus becomes modeling \( P(A|C') \) where \( A = \{c_k, ..., c_{k+l}\} \) such that \( k \geq s \) and \( k + l \leq e \).

Because T5 has a fixed maximum context length of 512 tokens, input passages that contain \( n > 512 \) tokens must be split up into smaller sub-passages. We perform this splitting such that no sentences are divided between sub-passages and all sub-passages have a roughly equal number of sentences. Finally, to generate questions, we iteratively choose the start and end of each sentence in a given sub-sentence as our \( c_s \) and \( c_e \) and extract at most one answer span per sentence. 3 We then generate one question per extracted answer span using the same model in an answer-aware fashion.

### Table 1: Analysis of the summaries written by our three RAs.

<table>
<thead>
<tr>
<th></th>
<th>Key-Term Coverage</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>

Tasks afforded by T5’s task-specific prefixes such as “generate question:” and “extract answer:”.

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).\(^5\) 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 recruited three expert annotators, all undergraduates in computer science, to evaluate the quality of the question-answer pairs. All 300 pairs were given to all three annotators. We asked the annotators to answer the following yes/no questions: a) Would you directly use this question as a flashcard?, b) Is this question grammatical?, c) Does this question make sense out of context?, d) Is this question relevant? and e) Is the answer to this question correct? We report these in our tables as “Acceptable?”, “Grammatical?”, “Interpretable?”, “Relevant?”, and “Correct?” respectively. We provided many annotation examples to our annotators and wrote clear guidelines about each category to

\(^{1}\)https://huggingface.co/valhalla/t5-base-qa-qqg-hl

\(^{2}\)https://huggingface.co/facebook/bart-large-cnn

\(^{3}\)https://huggingface.co/facebook/bart-large-cnn

\(^{4}\)https://web.stanford.edu/~jurafsky/slp3/
Table 2: Recall of bolded key terms from the textbook. Numbers represent percentage of terms present in any of the n question/answer pairs selected from the given source.

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>69.7</td>
<td>98.3</td>
<td>79.7</td>
<td>79.0</td>
<td>91.7</td>
</tr>
<tr>
<td>A2</td>
<td>48.7</td>
<td>90.7</td>
<td>70.7</td>
<td>71.3</td>
<td>90.7</td>
</tr>
<tr>
<td>A3</td>
<td>47.7</td>
<td>86.3</td>
<td>59.7</td>
<td>69.0</td>
<td>90.0</td>
</tr>
<tr>
<td>Pairwise IAA</td>
<td>0.41, 0.50, 0.33</td>
<td>0.16, 0.49, 0.10</td>
<td>0.51, 0.43, 0.32</td>
<td>0.41, 0.29, 0.25</td>
<td>0.03, 0.08, 0.006</td>
</tr>
</tbody>
</table>

Figure 3: Results of our human evaluation for each input method. Numbers represent the proportion of questions that were labeled as having the given attribute (as determined by majority vote among our three annotators).

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

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

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

6Examples of questions for each category on which there was significant disagreement are listed in Appendix D
References


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


Weizhen Qi, Yu Yan, Yeyun Gong, Dayieng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for


**A Software and Data**

To assist with reproducibility, included in the supplementary materials are:

1. The 300 questions used in our study as well as the annotations collected.

2. The 2,194 un-annotated questions with associated source and section number.

3. The text sources used to generate the questions (three chapters of cleaned text from Jurafsky and Martin, three sets of human summaries, one set of automatic summaries).

4. The script used to generate questions (some details redacted to preserve anonymity).

The full software will be released after the end of the review process.

**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 upwards of 10 or more examples given to ensure maximum possible agreement between annotators. Several discussion sessions were held between the authors and annotators to ensure that these guidelines were well understood and that they were sensible for the task.

During annotation, annotators were not given the original source text from which the question was generated. Instead, they were given the original textbook chapters to use as reference material for relevance and were allowed to use online search engines to check for grammaticality and correctness.

**C Comparison Across Chapters**

In Table 5 we report the distribution of scores across chapters. The titles of the three chapters are “Regular Expressions, Text Normalization, Edit Distance”, “N-gram Language Models”, and “Naive Bayes and Sentiment Classification” respectively. We note that the scores of the generated questions
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/ungrammatical 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 = 90)</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.

are largely consistent across the three chapters, with lower average relevance for Chapter 2 questions likely owing to the source material containing many worked examples of regular expressions and other application-specific details.

D Example Disagreements

In Table 6 we list questions for which there was at least one dissenting annotator for the given category.

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 summaries from our three annotators and the pre-trained BART model. We see that the length of summary varies drastically between our three human annotators, each of them making different decisions on whether or not to keep or discard certain pieces of information. Another thing to note is that automatic summaries are much more extractive in nature while the human summaries are generally more abstractive.
We see that the automatic summaries tend to be more extractive while the human summaries are more abstractive.

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”.

| Correct? | Original Text: What do we do with words that are in our vocabulary (they are not unknown words) but appear in a test set in an unseen context (for example they appear after a word they never appeared after in training)? To keep a language model from assigning zero probability to these unseen events, we’ll have to shave off a bit of probability mass from some more frequent events and give it to the events we’ve never seen. This modification is called smoothing or discounting. In this section and the following ones we’ll introduce a variety of ways to do smoothing: Laplace (add-one) smoothing, add-k smoothing, stupid backoff, and Kneser-Ney smoothing. | Original Text: As we saw in the previous section, naive Bayes classifiers can use any sort of feature: dictionaries, URLs, email addresses, network features, phrases, and so on. But if, as in the previous section, we use only individual word features, and we use all of the words in the text (not a subset), then naive Bayes has an important similarity to language modeling. Specifically, a naive Bayes model can be viewed as a set of class-specific unigram language models, in which the model for each class instantiates a unigram language model. Since the likelihood features from the naive Bayes model assign a probability to each word \( P(\text{word} | \text{class}) \), the model also assigns a probability to each sentence. |
|---|---|---|
| Acceptable? | Q: What is another name for a corpus that NLP algorithms learn from? A: training corpus | Q: What encode some facts that we think of as strictly syntactic in nature? A: Japanese and Thai |
| Grammatical? | Q: What are words like uh and um called fillers? A: filled pauses | Q: What isn’t big enough to give us good estimates in most cases? A: s or nothing |
| Relevant? | Q: What word has the same lemma cat but are different wordforms? A: cats | Q: What do algorithms have to deal with? A: ambiguities |
| Table 6: Questions for which there was disagreement on the label for the given category |