Enhancing Human Summaries for Question-Answer Generation in Education

Hannah Gonzalez, Liam Dugan, Eleni Miltsakaki, Zhiqi Cui,
Jiaxuan Ren, Bryan Li, Shriyash Upadhyay, Etan Ginsberg, Chris Callison-Burch
University of Pennsylvania
hannahgl,ldugan,elenimi,zhiqicui,rjx,bryanli,shriyash,etangins,ccb@seas.upenn.edu

Abstract

We address the problem of generating high-quality question-answer pairs for educational materials. Previous work on this problem showed that using summaries as input improves the quality of question generation (QG) over original textbook text and that human-written summaries result in higher quality QG than automatic summaries. In this paper, a) we show that advances in Large Language Models (LLMs) are not yet sufficient to generate quality summaries for QG and b) we introduce a new methodology for rewriting bullet point student notes into fully-fledged summaries and find that our methodology yields higher quality QG. We conducted a large-scale human annotation study of generated question-answer pairs for the evaluation of our methodology. In order to aid in future research, we release a novel dataset of 9.2K human annotations of generated questions.

1 Introduction

Automated generation of question-answer pairs for education can be used to assist students with self-guided reviews of educational materials or to support instructors with the creation of assessment materials. A key challenge for these question generation (QG) models is to ensure the relevancy of generated questions. Most human evaluation of QG models often emphasizes the grammaticality and fluency of the generated questions, rather than their relevance (Subramanian et al., 2017). For educational applications, this shortcoming is critical.

A recent study by Dugan et al. (2022) showed that providing QG models with human-written summaries as input, instead of original textbook text, increases question relevance, acceptability, and interpretability. The study also demonstrated that using automatically generated summaries as input improved QG quality over original textbook input, but not as much as human-written summaries. We investigate whether advances to large language models (LLMs) like GPT-3 have closed this gap and introduce a novel methodology for generating summaries using student notes in the form of bullet points as input.

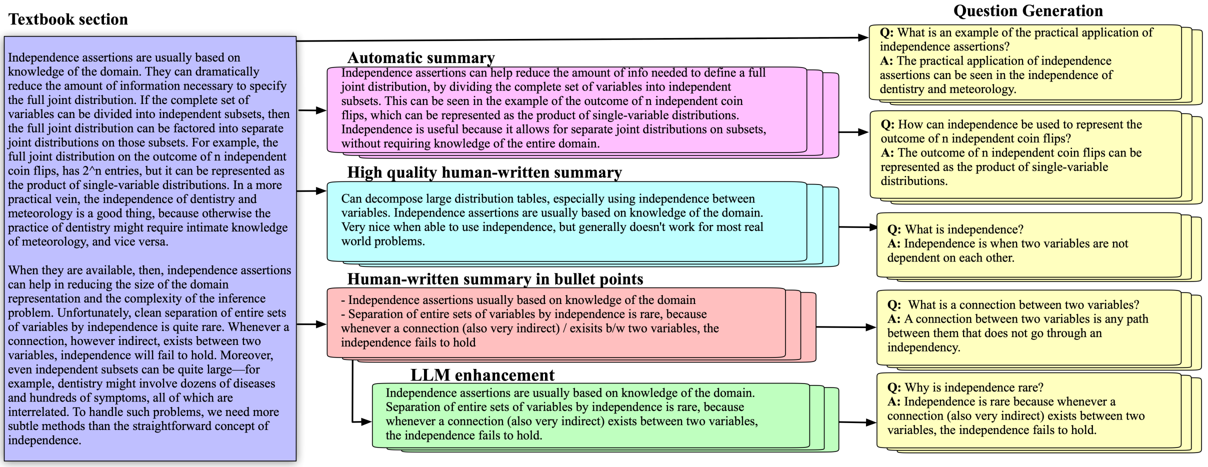
The main contributions of our research are:

1. We find that using human summaries as input to QG models still results in higher quality questions than generated summaries, even when using GPT-3 for summarization.
2. We propose a new methodology, Bull2Sum, that rewrites bullet point student notes into fully-fledged summaries.
3. We show that our Bull2Sum method of generating summaries as input to QG results in high-quality question-answer pairs.
4. We conduct a large-scale human evaluation study of generated question-answer pairs using our method and baselines.
5. To assist in future research, we release two datasets: a dataset with 9.2K human annotations of generated questions, as well as a dataset with summaries written by 392 students for 96 sub-chapters of two textbooks.

2 Related Work

Prior work in question generation has focused primarily on using sequence-to-sequence models to generate questions from a given context passage. These methods can either be answer-aware (i.e., an answer span is given to the model, along with the passage) or answer-agnostic (i.e., just the context passage is given). Our work focuses on the latter case, in which the model has the much more challenging task of generating the answer as well as the question.

Subramanian et al. (2018) accomplished this by decomposing the generation process into two stages: answer-phrase extraction and answer-aware


Textbook section
Automatic summary
High quality human-written summary
Human-written summary in bullet points
LLM enhancement
Question Generation

Independence assertions are usually based on knowledge of the domain. They can dramatically reduce the amount of information necessary to specify the full joint distribution. If the complete set of variables can be divided into independent subsets, then the full joint distribution can be factored into separate joint distributions on those subsets. For example, the full joint distribution on the outcome of n independent coin flips, has 2^n entries, but it can be represented as the product of single-variable distributions. In a more practical vein, the independence of dentistry and meteorology is a good thing, because otherwise the practice of dentistry might require intimate knowledge of meteorology, and vice versa.

When they are available, then, independence assertions can help in reducing the size of the domain representation and the complexity of the inference problem. Unfortunately, clean separation of entire sets of variables by independence is quite rare. Whenever a connection, however indirect, exists between two variables, independence will fail to hold. Moreover, even independent subsets can be quite large—for example, dentistry might involve dozens of diseases and hundreds of symptoms, all of which are interrelated. To handle such problems, we need more subtle methods than the straightforward concept of independence.

Independence assertions can help reduce the amount of info needed to define a full joint distribution, by dividing the complete set of variables into independent subsets. This can be seen in the example of the outcome of n independent coin flips, which can be represented as the product of single-variable distributions. Independence is useful because it allows for separate joint distributions on subsets, without requiring knowledge of the entire domain.

Can decompose large distribution tables, especially using independence between variables. Independence assertions are usually based on knowledge of the domain. Very nice when able to use independence, but generally doesn't work for most real world problems.

- Independence assertions usually based on knowledge of the domain
- Separation of entire sets of variables by independence is rare, because whenever a connection (also very indirect) / exisits b/w two variables, the independence fails to hold

Independence assertions are usually based on knowledge of the domain. Separation of entire sets of variables by independence is rare, because whenever a connection (also very indirect) exists between two variables, the independence fails to hold.

Q: What is an example of the practical application of independence assertions?
A: The practical application of independence assertions can be seen in the independence of dentistry and meteorology.

Q: How can independence be used to represent the outcome of n independent coin flips?
A: The outcome of n independent coin flips can be represented as the product of single-variable distributions.

Q: What is independence?
A: Independence is when two variables are not dependent on each other.

Q: What is a connection between two variables?
A: A connection between two variables is any path between them that does not go through an independency.

Q: Why is independence rare?
A: Independence is rare because whenever a connection (also very indirect) exists between two variables, the independence fails to hold.

Figure 1: Different types of summaries such as automatic summaries, fully-fledged human-written summaries, human-written summaries in bullet points, LLM enhanced summaries (with our proposed method), and textbook text as used input to QG.

QG. Follow-up work from Sun et al. (2018) introduced a position-aware component to localize answers in the input context. Work by Wang et al. (2020) added joint training between the two stages of the pipeline. Other work has found that transforming the input context passage can aid in answer-phrase extraction. Lewis et al. (2021) filtered out passages that are unlikely to contain answers to human-written questions. Qu et al. (2021) generated coarse keyphrases from input passages to help guide the answer extraction model. Zhao et al. (2022) used an "event-centric" summarizer to generate a sequence of events, allowing them to ask better questions.

More recently, Dugan et al. (2022) showed that providing answer extraction models with human-written or LM-generated summaries significantly improved the relevance and interpretability of generated questions. We build on this insight and further investigate the gap in question quality between human-written summaries and LM-generated summaries. Dugan et al. used a BART model (Lewis et al., 2019) for automatic summarization. However, recent work suggests that summaries generated by large language models such as GPT-3 are overwhelmingly preferred by human annotators (Goyal et al., 2022). In what follows, we report the results of experiments that we conducted to evaluate whether Large Language Models can, indeed, generate quality summaries for the task of generating question-answer pairs for educational materials.

Step - **Description**
1. Zero-shot - We generated new summaries from the human-written bullet style summaries with GPT-3 using the following prompt: *"Here's an outline, please expand it into full sentences and paragraphs: {human-written summary in bullet style with incomplete sentences}"*
2. Few-shot - We reviewed 10 examples by fact-checking and removing repeated phrases. We added these examples to the prompt and then generated 100 more summaries out of the hand-written summaries in bullet style.
3. Fine-tuning - We fine-tuned GPT-3's Davinci model with the 100 summaries generated from the few-shot stage. The format of the fine-tuned model was the following: StudentSummary: <bullet-point summary> GPT3Summary: <paragraph style generated summary>

Table 1: Description of bootstrapping process to modify the human-written summary style

3 Methodology

As mentioned earlier, the central goal of this study is to address the problem of providing quality summaries of educational materials to QA models in order to generate important and relevant QA pairs. To investigate this problem, we ran two groups of experiments. First, we evaluated if GPT-3 generates better QA pairs than T5 which was used for the same task in prior work. In the second group of experiments, we investigated the impact of different types of input on the quality of the generation of QA pairs in addition to different ways of obtaining summaries. To this end, we collected summaries written by college students on course textbooks and classified them into two major categories: fully-fledged summaries and bullet-point summaries. Fully-fledged summaries consisted of complete grammatical sentences that formed a coherent paragraph. Bullet-point summaries consisted of bullet points or other fragments taken as short notes. In addition to these two types of input generated by humans, we introduced and compared a new method of generating summaries from bullet-point notes, which we call Bull2Sum (from bullets to summaries). Bull2Sum takes as input bullet-point summaries and rewrites them into fully-fledged summaries.

3.1 Human-written Summaries

We collected human-written summaries from a total of 570 undergraduate and Master's students enrolled in a graduate-level Artificial Intelligence course. Students wrote summaries of 56 sections of 14 chapters of the Russell and Norvig (2020) textbook "Artificial Intelligence: A Modern Approach" and 40 sections of 6 chapters of the Jurafsky and Martin (2022) textbook "Speech and Language Processing." The collected summaries varied widely in terms of style. Some students wrote fully-fledged summaries with complete sentences organized into paragraphs. Others summarized the chapters in the form of bullet-point notes. The students were incentivized to write quality summaries because they were allowed to use them as supplementary material during the final exam. We release the summaries of a total of 392 students who agreed to share their anonymized summaries with the research community.

3.2 Bootstrapping Training Data for LLMs

In order to generate in-domain data for fine-tuning large language models, such as GPT-3, we employed a bootstrapping approach. We first generated a small amount of data pairs by using the model in a zero-shot fashion. We then manually reviewed the generated examples by fact-checking and removing repeated phrases. We then used this filtered set of synthetic data as in-context examples to generate a larger set of high-quality few-shot data. We used this final set of examples as our fine-tuning dataset.

3.3 Fine-tuned Model for Rewriting Bullet Points into Summaries

We introduce a fine-tuned model, Bull2Sum, that we trained in order to rewrite summaries written in bullet points or short notes into fully-fledged summaries. We built this model by fine-tuning GPT-3 using the same bootstrapping approach described in the previous section.[1] Table 1 outlines and compares all the methods in our experiments.

Step - **Description**
1. Zero-shot - We generated QA pairs with GPT-3 using the prompt *"Write 5 to 10 questions along with their corresponding answers from the summary."* + *"Summary: "* + *student_summary* + *"Question: <Text of question.>* + *"Corresponding answer: <Text of corresponding answer.>"*
2. Few-shot - We reviewed 20 examples by fact-checking and formatting. We added these examples to the prompt and then generated QA pairs out of summaries generated by Bull2Sum.
3. Fine-tuning - We fine-tuned a model with QA pairs generated from the few-shot stage.

Table 2: Description of bootstrapping process to generate QA pairs from a text.

3.4 Question Generation Models

For question generation, we again used a bootstrapping procedure to fine-tune GPT-3 to perform answer-agnostic question generation. We outline this procedure in Table 2. We generated questions from this model and compared them to questions generated from the same fine-tuned T5 model used in Dugan et al. (2022).

4 Experiments

We compare the performance of two LLMs trained to do QG in 5 text input conditions. So, we ran a total of 10 experiments. Each condition is a different type of input to the model, including a condition with summaries generated by a new model that we fine-tuned, Bull2Sum, which rewrites bullet points or short notes into fully-fledged sentences. We describe this model in Section 3.3.

Text input conditions

1. Original text from textbook.
2. Zero-shot summary generated by GPT-3.
3. Fully-fledged human-written summary.
4. Bullet-point human-written summary.
5. Summary generated by Bull2Sum.

In order to evaluate and compare the performance of both T5 and GPT-3 under the 1st condition (original text from textbook), we extracted 47 sections from the Russell and Norvig (2020) textbook (omitting figures, tables, and equations). For the 2nd condition, we used GPT-3 to summarize the

[1] We ran all the reported experiments in November 2022, using text-davinci-002.

	T5					GPT-3				
Type of Input to QG Model	**Acc.**	**Gram.**	**Interp.**	**Rel.**	**Corr.**	**Acc.**	**Gram.**	**Interp.**	**Rel.**	**Corr.**
1) Original text from textbook	35%	94%	68%	69%	52%	50%	79%	71%	77%	59%
2) GPT-3 generated summary from textbook text	48%	93%	72%	76%	59%	67%	93%	84%	86%	75%
3) Fully-fledged human-written summary	44%	88%	70%	85%	58%	**73%**	95%	**92%**	**95%**	79%
4) Bullet-point human-written summary	50%	86%	72%	88%	61%	53%	93%	86%	89%	66%
5) Bull2Sum summary	55%	93%	79%	93%	67%	70%	**96%**	90%	93%	**80%**

Table 3: Evaluation of questions generation by T5 and by GPT-3 using different types of summaries as input. Humans evaluated whether the questions were Acceptable, Grammatical, Interpretable, Relevant, and Correct.

passages from the first condition with the following prompt: *"Please summarize the following text using complete sentences:"* For the 3rd and 4th conditions, we used 96 fully-fledged human-written summaries and 96 bullet-point human-written summaries from Russell and Norvig (56 sections) and Jurafsky and Martin (40 sections). For the 5th condition, we used our fine-tuned model Bull2Sum described in Section 3.3 on the 96 bullet-point human-written summaries. There is a one-to-one mapping in conditions 3, 4, and 5 as they are from the same textbook sections. Conditions 1 and 2 are from a subset of these textbook sections. Table 4 in the Appendix provides detailed information and statistics about the data.

5 Evaluation

We performed a human evaluation study to measure the QG performance of the models GPT-3 and T5 under our 5 different input conditions, as described in Section 4. We had a total of 66 annotators, all University students enrolled in an advanced Computer Science course titled *Artificial Intelligence*. Prior to the annotations, students signed a consent form to participate in the experiment and were rewarded with extra credit for their participation. Moreover, we had a training session with the students to review the guidelines and demo the annotation tool. We employed the evaluation guidelines defined in Dugan et al. (2022). For each generated QA pair, the annotators evaluated the following criteria:

1. Acceptable: Would you directly use this question as a flashcard?
2. Grammatical: Is this question grammatically correct?
3. Interpretable: Does this question make sense out of context?
4. Relevant: Is this question relevant?
5. Correct: Is the answer to the question correct?

Our team created a web-based tool (as illustrated in Appendix Figure 2) in order to increase the scalability and ease of annotations. We randomly selected 10 QA pairs generated from each of our 5 input conditions by both the T5 and GPT-3 models. We divided our 66 annotators into groups of 3, for a total of 22 groups. Each group would annotate the same group of questions generated by the different models for the same data. Given that we had 22 groups of annotators, we collected 3,080 question-answer (QA) pairs annotated, i.e., 220 QA pairs annotated per input condition. We computed pairwise inter-annotator agreement (IAA) analysis using Fleiss's Multi-π method (Artstein and Poesio, 2008) for finding the agreement for more than two coders and found IAA rates between 0.39–0.44 for our 5 evaluation criteria. We report the results in Table 8.

6 Results and Discussion

Table 3 shows the percentage of generated QA pairs where the annotations were "yes" for both GPT-3 and T5. Unsurprisingly, the larger LM GPT-3 demonstrated superior performance in the question generation task compared to T5. It produced a) higher quality flashcards, b) more questions that were coherent out of context, and c) more accurate answers. We found that fully-fledged summaries are better input than GPT-3 generated summaries, which are better than bullet-point human-written summaries. Our methodology of applying our rewriting model Bull2Summ for rewriting the bullet summaries into fully-fledged summaries results in a substantial increase in the quality of the QA pairs. Specifically, the acceptability score improves from 53% (bullet points) to 70% (nearly equal to the 73% of fully-fledged human-written summaries).

Our experiments show that although GPT-3 performs better than T5 in QG, it is not sufficient to improve a) the quality of QA pairs and b) the quality of the automated summaries as input. Carefully written human summaries are still better than automatic summaries generated by GPT-3. However, our novel method of rewriting short bullet-point notes into summaries can be effectively used to generate quality QA pairs.

7 Limitations

In this work, we explored question generation for computer science textbooks. We have not yet explored a broader range of course subjects, and it may be that the prevalence of computer science knowledge on the Internet, including through forums like Stack Exchange, makes QG easier for this discipline than for others. Furthermore, we examine a relatively narrow range of question types. Other questions –like multiple choice questions, or compare and contrast questions– will require deeper exploration and substantial adaptation of the methodology that we proposed.

8 Ethics Statement

Potential risks : As with all large language models, the models used in our research have the potential to generate factually incorrect information. This is a potential risk given that our intended application is for education. As reported in our paper, our best-performing models produce acceptable quality flashcard questions only 70% of the time. The remaining 30% is significant enough that manual review by course is necessary before questions are deployed to students.

Intended Use : Our models and methods shown here are for research purposes only. They should not be deployed in the real world as solutions without further evaluation.

Potential applications : Bull2Sum could be utilized in the field of education to convert course slides into summaries, which can then be used to generate pertinent and significant questions for the course. This application could enhance and facilitate students' exam preparation.

Acknowledgements

This research is based upon work supported in part by the DARPA KAIROS Program (contract FA8750-19-2-1004), the DARPA LwLL Program (contract FA8750-19-2-0201), the IARPA HIATUS Program (contract 2022-22072200005), and the NSF (Award 1928631). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, IARPA, NSF, or the U.S. Government.

Furthermore, the University of Pennsylvania provided valuable support for this research through the Vagelos Undergraduate Research Grant.

Additionally, we would like to extend our gratitude to Suraj Patil for providing one of the fine-tuned question generation models (T5) that we used in our experiment.

Finally, we would like to thank Jack Collison for his helpful suggestions.

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational linguistics*, 34(4):555–596.

Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, DaHyeon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. A feasibility study of answer-agnostic question generation for education. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1919–1926, Dublin, Ireland. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3.

Daniel Jurafsky and James H Martin. 2022. *Speech and language processing (3rd Edition Draft)*. Prentice Hall NJ.

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.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Fanyi Qu, Xin Jia, and Yunfang Wu. 2021. Asking questions like educational experts: Automatically generating question-answer pairs on real-world examination data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2583–2593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stuart Russell and Peter Norvig. 2020. *Artificial Intelligence : A Modern Approach*. Pearson, Boston.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Yoshua Bengio, and Adam Trischler. 2017. Neural models for key phrase detection and question generation. *arXiv preprint arXiv:1706.04560*.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2018. Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88, Melbourne, Australia. Association for Computational Linguistics.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium. Association for Computational Linguistics.

Bingning Wang, Xiaochuan Wang, Ting Tao, Qi Zhang, and Jingfang Xu. 2020. Neural question generation with answer pivot. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9138–9145.

Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. Educational question generation of children storybooks via question type distribution learning and event-centric summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5073–5085, Dublin, Ireland. Association for Computational Linguistics.

Type of Input to QG Model	Average len of the text	Average sentence len	Average num of sentences	Num of T5 QA pairs	Num of GPT-3 QA pairs
1) Original text from textbook	2260	116	16	774	199
2) GPT-3 generated summary from textbook text	694	103	5	265	194
3) Fully-fledged human-written summary	784	74	9	834	374
4) Bullet-point human-written summary	930	378	4	399	279
5) Bull2Sum summary	687	89	6	605	433
6) Few-shot generated summary	751	108	7	609	447
7) Summary generated with our fine-tuned model	781	92	7	698	356

Table 4: Statistics of the different types of summaries as input. We report the average length of the text (in chars), the average sentence length (in chars), the average number of sentences, the number of T5 QA pairs, and the number of GPT-3 QA pairs.

Type of Input to QG Model	Summary
Original text from textbook	27.1 The Limits of AI 27.1.2 The argument from disability The “argument from disability” makes the claim that “a machine can never do X.” As examples of X, Turing lists the following: Be kind, resourceful, beautiful, friendly, have initiative, have a sense of humor, tell right from wrong, make mistakes, fall in love, enjoy strawberries and cream, make someone fall in love with it, learn from experience, use words properly, be the subject of its own thought, have as much diversity of behavior as man, do something really new. In retrospect, some of these are rather easy—we’re all familiar with computers that “make mistakes.” Computers with metareasoning capabilities (Chapter 5) can examine heir own computations, thus being the subject of their own reasoning. A century-old technology has the proven ability to “make someone fall in love with it”—the teddy bear. Computer chess expert David Levy predicts that by 2050 people will routinely fall in love with humanoid robots. As for a robot falling in love, that is a common theme in fiction,1 but there has been only limited academic speculation on the subject (Kim et al., 2007). Computers have done things that are “really new,” making significant discoveries in astronomy, mathematics, chemistry, mineralogy, biology, computer science, and other fields, and creating new forms of art through style transfer (Gatys et al., 2016). Overall, programs exceed human performance in some tasks and lag behind on others. The one thing that it is clear they can’t do is be exactly human.
GPT-3 generated summary from text	The "argument from disability" claims that machines can never do X, where X is a characteristic of humans. Some examples of X include being kind, resourceful, beautiful, friendly, having initiative, having a sense of humor, being able to tell right from wrong, making mistakes, falling in love, enjoying strawberries and cream, making someone fall in love with it, learning from experience, using words properly, being the subject of its own thought, having as much diversity of behavior as man, and doing something really new.

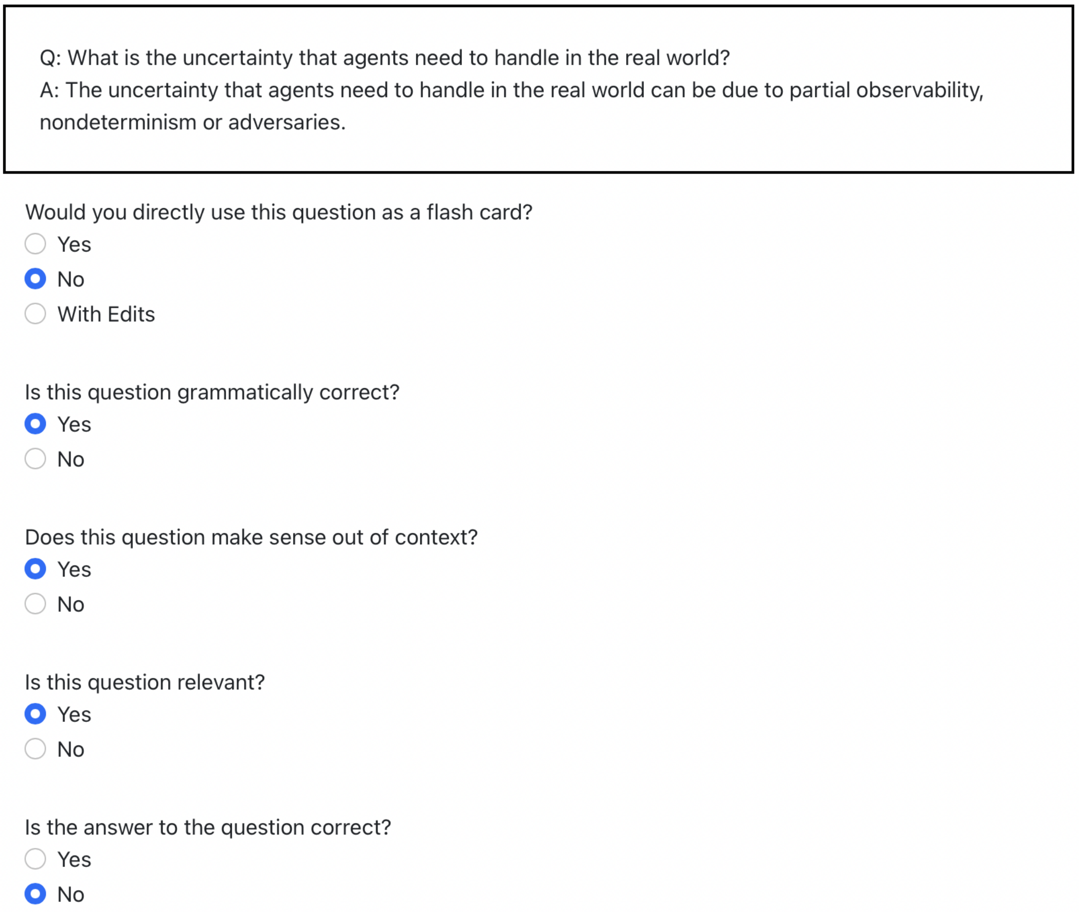
Table 5: Qualitative example of sub-chapter from original textbook text and its corresponding GPT-3 generated summary.

Type of Input to QG Model	Summary
Higher quality human-written summary	Perceptron can easily compute AND and OR, but cannot handle XOR at all! This is because perceptron is a linear classifier, and thus decision boundary is a line. XOR can be calculated by a layered network of units. Example using two ReLU-based units. Can view hidden layer of the network as forming a representation for the input. Neural networks big advantage is they can automatically learn to form these representations well. Note: XOR requires network of units with non-linear activation functions, i.e. CANNOT do it with just perceptrons (simple linear units). This due to fact that many layers of linear units can be reduced to one layer with a certain weights function, which we know can't solve the XOR problem.
Poorer quality human-written summary	• perceptron • a simple neural unit • has binary output • does not have a non-linear activation function • XOR: not linearly separable • Non-linear activation functions • a network formed by many layers of purely linear units can always be reduce to a single layer of linear units with appropriate weights
Summary expanded into full sentences and paragraphs by zero shot	A perceptron is a simple neural unit that has binary output and does not have a non-linear activation function. The XOR problem is not linearly separable, meaning that a network formed by many layers of purely linear units can always be reduced to a single layer of linear units with appropriate weights. Non-linear activation functions are necessary in order to solve the XOR problem.
Few-shot generated summary	The perceptron is a simple neural unit that has binary output and does not have a non-linear activation function. The XOR problem is not linearly separable. Non-linear activation functions are important because a network formed by many layers of purely linear units can always be reduced to a single layer of linear units with appropriate weights.
Summary generated with our fine-tuned model	A perceptron is a simple neural unit that has a binary output and does not have a non-linear activation function. An example of a perceptron is a machine that recognizes numbers written in digits. The machine would have to be trained first by being shown examples of the correct outputs for each input. The XOR problem is an example of a perceptron that is not linearly separable. A perceptron that can solve the XOR problem is said to be bi-dimensional. A perceptron that can solve the XOR problem is said to be bi-dimensional. A network formed by many layers of purely linear units can always be reduced to a single layer of linear units with appropriate weights.

Table 6: Qualitative example of our methodology to enhance summary quality.

Type of Input to QG Model	**GPT-3 QA**	**T5 QA**
Original text from textbook	Q: Are programs currently capable of doing all of these things? A: Programs are currently capable of doing some of these things, but not all of them.	Q: What is a century-old technology that can make someone fall in love with it? A: teddy bear
GPT-3 generated summary from text	Q: What is the argument from disability? A: The argument from disability claims that machines can never do x, where x is a characteristic of humans.	Q: What does the "argument from disability" claim machines can never do? A: X
Higher quality human-written summary	Q: How does a perceptron compute the and logical operation? A: A perceptron computes the and logical operation by adding the activation values of the multiple layers of units in the perceptron.	Q: Perceptron can easily compute what? A: AND and OR
Poorer quality human-written summary	Q: What does a perceptron not have? A: A perceptron does not have a non-linear activation function.	Q: What is a simple neural unit - has binary output - does not have a non-linear activation function? A: perceptron
Summary expanded into full sentences and paragraphs by zero shot	Q: What is the xor problem? A: The xor problem is a problem in which two inputs cannot be separated by a linear function.	Q: What problem is not linearly separable? A: XOR
Few-shot generated summary	Q: What is the xor problem not linearly separable? A: The xor problem is not linearly separable.	Q: A network formed by many layers of what can always be reduced to a single layer of linear units with appropriate weights? A: purely linear units
Summary generated with our fine-tuned model	Q: What is a perceptron? A: A perceptron is a simple neural unit that has binary output and does not have a non-linear activation function.	Q: A perceptron that can solve the XOR problem is said to be what? A: bi-dimensional

Table 7: Qualitative examples of QA pairs generated by GPT-3 and T5 from different types of summary or text as input.



Q: What is the uncertainty that agents need to handle in the real world?
A: The uncertainty that agents need to handle in the real world can be due to partial observability, nondeterminism or adversaries.

Would you directly use this question as a flash card?
○ Yes
● No
○ With Edits

Is this question grammatically correct?
● Yes
○ No

Does this question make sense out of context?
● Yes
○ No

Is this question relevant?
● Yes
○ No

Is the answer to the question correct?
○ Yes
● No

Figure 2: An example of annotation interface. You can find the annotation tutorial here.

IAA	
Acceptable	0.39
Grammatical	0.44
Interpretable	0.42
Relevant	0.42
Correct	0.39

Table 8: Mean of pairwise agreement in all 22 groups