Generation and Assessment of Multiple-Choice Questions from Video Transcripts using Large Language Models

Taimoor Arif
University of Michigan
Ann Arbor, MI, USA
taimoora@umich.edu

Sumit Asthana
University of Michigan
Ann Arbor, MI, USA
asumit@umich.edu

Kevyn Collins-Thompson
University of Michigan
Ann Arbor, MI, USA
kevynct@umich.edu

ABSTRACT

We present an empirical study evaluating the quality of multiple-choice questions (MCQs) generated by Large Language Models (LLMs) from a corpus of video transcripts of course lectures in an online data science degree program. With our database of thousands of generated questions, we conducted both human and automated judging of question quality on a representative sample using a broad set of criteria, including well-established Item Writing Flaw (IWF) categories. We found the number of average IWFs per MCQ ranged from 1.6 (rule-based verification) to 2.18 (LLM-based). Among the most frequently identified MCQ flaws were lack of enough context (17%) or answer choices with at least one implausible distractor (57%). Both human and automated assessment identified implausible distractors as one of the most frequent flaw categories. Results from our human annotation study were generally more positive (51–65% good items) compared to our automated assessment study results, which tended toward greater flaw identification (15–25% good items), depending on evaluation method.

CCS CONCEPTS

• **Information systems → Multimedia information systems**; • **Applied computing → Computer-assisted instruction**; • **Computing methodologies → Natural language generation**.

KEYWORDS

Question Generation, Large Language Models, Educational Video

ACM Reference Format:
Taimoor Arif, Sumit Asthana, and Kevyn Collins-Thompson. 2024. Generation and Assessment of Multiple-Choice Questions from Video Transcripts using Large Language Models. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale (L@S '24), July 18–20, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3657604.3664714

1 INTRODUCTION

AI-based applications are increasingly being used to automatically create educational assessments and other resources such as practice question sets [12], personalized question set recommendations to students [10] and lecture summaries [7]. In particular, Large Language Models (LLMs) allow easy curation of data from existing course materials (text, audio and video) through simple natural language prompts [9]. However, while applications of LLMs are rapidly growing due to their ease of on-demand content generation, more insight is needed into the quality of the generated metadata and resources, as part of a broader challenge of mapping out where LLMs can provide value in order to integrate their capabilities for improving educational outcomes [5].

We describe how we generated and curated structured natural language data using LLMs for online courses, which are heavily video-based. We provide an extensive empirical study of the quality of the resulting multiple-choice questions, based on both human judges and automated rule-based and LLM-based scoring approaches. This is the first study to our knowledge to examine large-scale LLM question generation from video transcripts. We also discuss how our results connect with related work on evaluating question quality, reflect on the use of LLM-provided explanations and automatic correction of flawed questions, and discuss the problem of providing complete and correct question context.

2 RELATED WORK

Our work connects with several prior studies on generating and assessing the quality of questions using Large Language Models. Kurdi et al. [10] provide a comprehensive overview of earlier NLP research on question generation, before the widespread deployment of recent LLMs. The most closely related work to ours is the recent study by Moore et al. [11] that used both rule-based and LLM-based methods for evaluating the quality of multiple-choice question (MCQs), based on 19 different quality attributes drawn from Item-Writing Flaws (IWF) guidelines. A rule-based approach uses a specific programming method to check each of the IWF criteria. Using the implementation by [11] as a starting point, we extended their evaluation framework to incorporate additional context-based quality measures to handle video content. Our rule-based evaluation of our generated question dataset followed their core set of 19 IWF rules, and our LLM-based evaluation is based on a subset of those rules, applied to our own video-based dataset. In addition, we conducted a comparison of human annotators, GPT 3.5 Turbo, and GPT-4 results.

Since a single lecture video transcript typically covers multiple concepts and topics, segmenting it into sections allows more focused questions about specific details at key moments in a lecture. Our approach to segmenting the video transcript into moments is related to how Bhat et al. [1] implemented question generation in a two-step process: first, summarization of the source content; and

L@S '24, July 18–20, 2024, Atlanta, GA, USA.

ACM ISBN 979-8-4007-0633-2/24/07
https://doi.org/10.1145/3657604.3664714

second, generating the question using the title and summary content as context in prompting ChatGPT (GPT 3.5 Turbo). The same study performed a limited evaluation that was restricted to assessing the complexity of their generated questions using three criteria that included the equivalent of the 'Unclear Information' IWF. Their questions were also not multiple-choice and were designed to be used for long-form question answering.

Yuan et al. [13] examined how to improve question generation diversity using zero-shot prompt-based approaches to choosing high-quality questions from a set of LLM-generated candidates. Their evaluation of question quality used two methods: comparison against a set of reference questions, and human evaluation on a subset of the data. The meta-questions for human evaluation covered seven criteria similar to IWFs, including Grammatical Correctness, Offensiveness, Clarity, Relevance, Specificity, and Answerability. The generated questions were typically short-answer, not multiple-choice, so criteria that were important to our analysis, such as distractor plausibility, did not apply. We also focused on video transcripts, instead of Wikipedia articles and short stories.

Elkins et al. [6] conducted a human evaluation of educational questions generated by using LLMs with few-shot prompting, using teachers to assess quality and usefulness. Their quality metrics were Relevance, Grammatical Correctness, and Answerability, with an additional criterion for Adherence (whether the question was an instance of a desired difficulty or knowledge taxonomy level). Their results, based on teacher feedback, demonstrated that the questions generated were high quality and sufficiently useful, showing their promise for widespread use in the classroom setting. Bhowmick et al. [2] developed a modular framework for LLM generation of MCQs from textual content, with distinct modules for question generation, correct answer prediction, and distractor formulation. They used eight criteria for question quality that approximately correspond to Grammatical Correctness, Clarity, Answerability, Correctness, Predictability, and Distractor Implausibility. Finally, we note that LLMs are increasingly being used in technical domains to generate multiple-choice questions for high-stakes assessments: the study on application to medical exams [3] being a representative example.

3 METHODS

3.1 Dataset Generation

We generated a database containing rich metadata for all lecture videos in 11 courses from a graduate-level online degree program at a large U.S. university. The courses used for data generation were Machine Learning and Data Science courses like Supervised Learning, Unsupervised Learning, and Data Manipulation, each of which had at least 15 lecture videos. The metadata included our generated questions and was extracted from the lecture transcripts of these courses through an automated pipeline using the OpenAI GPT API (versions 3.5 and 4.0). The lecture transcripts were obtained from the hosting learning platform, which used a high-quality, human-assisted transcription process with a very low Word Error Rate so that (in general) noise or badly transcribed technical terms were not a problem. We started by using zero-shot prompting with GPT to summarize the lecture for a student who wants to review it and attempt questions from it. Then, we use insights from NLP research that effective segment boundaries should split passages into units that are topically coherent [8]. We leveraged GPT to segment the lecture into a small number (4 or 5) of non-overlapping segments by using a prompt that defines a segment as "a group of consecutive lines in the text where topics within the group are semantically more similar to each other than topics outside the group". The use of segments not only allows us to reduce the input size for the model for a quicker inference time and lower resource usage but ultimately results in more focused questions.

For each segment, we used a commercial LLM (GPT 3.5 or 4) to generate 15 multiple-choice questions from the segment transcript. For each question item, the LLM returned a question statement, four options, the correct option, and the explanation for the correct answer. In addition, we prompted the LLM to map each question onto five topics from the Machine Learning topic set defined by Wikipedia[1]. We stored all the generated data in a cloud-based database. Currently, there are thousands of questions in our database that span multiple courses and topic areas.

3.2 Evaluation Metrics

For any data generated or extracted using LLMs, it is critical to assess quality. Since we aim to use these MCQ items in student-facing applications for courses in the online degree program, we developed a multi-stage pipeline for evaluating the generated MCQ items according to multiple specific quality criteria. We complemented this automated, rule-based assessment with an assessment on a subsample by two human judges as well as a separate LLM-based assessment, as described next.

3.2.1 Human Annotation. Our first evaluation method was human annotation of the MCQ items, using two categories of quality metric: question-level, which focused exclusively on the question text, and option-level, which focused on the set of possible answer choices shown to the learner. Our question-level metrics were:

(1) **Relevance**: Does the question generated fall within the context of the lecture?
(2) **Grammar**: Are there any grammatical inconsistencies that might make the question difficult to understand?
(3) **Answerability**: Does the source text directly contain or support the answer to a question? Answerability can be of three types: **Direct**, **Indirect**, or **None**.
(4) **Difficulty**: Difficulty level of the question, with three levels: **Beginner**, **Intermediate**, and **Advanced**.
(5) **Clarity**: Does the question have an ambiguous question statement or option?
(6) **Contextual Specificity**: Does the question rely on specific content not available to the learner so that the question cannot be answered without the missing context?

The contextual specificity metric is especially important to measure for our scenario of question generation from video transcripts, in which the instructor may reference something that appears visually that is not described in the text. For example, in our use case, we found that GPT ended up generating a significant fraction of questions (reported below) that were relevant to a very specific context within the lecture because of GPT's reliance on lecture text. An example of such a question is, "What is the precision of the orange

[1] https://en.wikipedia.org/wiki/Machine_learning

Metric	Judge 1	Judge 2	IRR (Kappa)
Relevance	89%	94%	0.89
Grammar	99%	100%	0.99
Answerability	94% Direct, 4% Indirect	97% Direct, 3% Indirect	0.93
Difficulty	60% Beginner, 40% Intermediate	98% Beginner, 2% Intermediate	0.60
Clarity	94%	100%	0.94
Contextual Specificity	91%	89%	0.92
Question-Option Disjoint	97%	93%	0.90
Distractor Homogeneity	84%	75%	0.77
Distractor Plausibility	77%	54%	0.65

Table 1: Human annotation results (100-item question set) showing individual judge scores for question-level and option-level metrics, with corresponding IRR scores. Higher scores are better.

class in the displayed example?". While such context-dependent questions may be appropriate for in-video assessment, we wanted our question database to include more broadly applicable questions in scenarios where the original video may not be available. Thus, it was important for us to have a metric that identifies these context-dependent questions, both to avoid giving learners unanswerable questions, but also to distinguish between questions that could be used for in-video quizzes vs ones that could not. The option level metrics we used to evaluate the quality of the set of possible answers were as follows. In question item descriptions, the term 'distractor' refers to an incorrect option that is part of the item (but typically designed to appear plausible).

(1) **Question-Option Disjoint**: Is there a mismatch between the question and any option(s) given in the question that might confuse the question-taker?
(2) **Distractor Homogeneity**: Are the distractors offered in the question similar to the correct option?
(3) **Distractor plausibility**: Can the distractors be considered a viable answer to the question, and can they potentially confuse the question-taker as to which is the correct answer?

For human evaluation, we asked two graduate students, each with a strong background in Machine Learning and Data Science, to use the metrics described above to annotate 100 randomly selected MCQ items generated based on videos from two different masters-level courses in data science. The full set of original questions was generated using the GPT-3.5 turbo model. The annotators did an initial round of labeling on a smaller training set of items (not in the final question set). They met to discuss items where there was disagreement, worked to clarify any ambiguity or gap in the annotator instructions, and then did a second full labeling pass on the entire 100-item question set.

3.2.2 Automated annotation. For automated evaluation of the questions, we used the two automated methods of using the Item Writing Flaws rubric reported in the study by Moore et al. [11] to evaluate multiple choice questions. We applied the full rubric to the questions using the rule-based method and, for cost and efficiency reasons, used a selected subset of the most salient metrics for the GPT-based method. For automated evaluation, we generated a large pool of questions using both GPT-3.5 turbo and GPT-4. We used the rule-based method to tag all the questions and used the GPT-based tagging on 100 randomly selected questions from each of the GPT-3.5 and GPT-4 pools. This allowed us to compare the evaluation results based on the different methods and results from two different LLM versions. For automated evaluation, we used a larger pool of questions than human evaluation because of the speed of automated evaluation. All the questions that were evaluated by humans were also evaluated by the automated method.

4 RESULTS

We summarize our evaluation of question generation quality based on human judges, rule-based scoring, and LLM-based scoring.

4.1 Human Annotation Results

Overall, the human annotation yielded scores that suggested high overall MCQ item quality, with particularly high scores for question-level metrics. The results are summarized in Table 1. For all metrics, a higher score means higher quality. In the case of contextual specificity and question-option disjoint, the scores in the tables represent the questions that were not contextually specific and questions where there was no question-answer disjoint. The percentage of total MCQ items in this set rated by human judges as having no flaws (all positive ratings for the human annotation metrics) ranged from 51% to 65%. The inter-annotator agreement (IAA) for each metric using Cohen's Kappa [4] scores is also shown. The IAA is at or above 0.90 for all metrics except "Distractor Homogeneity" (0.77), "Distractor Plausibility" (0.65), and "Difficulty" (0.60). The especially low lack of agreement in the difficulty metric scores may be attributed to the more subjective perception of difficulty, which can be influenced by an annotator's level of expertise in the domain, prior exposure to the content, and understanding of the questions.

4.2 Automated Evaluation Results

In addition to human scoring, we explored automated quality metrics, using the IWF implementation from Moore et al. [11] as a starting point that provided both rule-based estimates and LLM estimates. A complete list with a description of each IWF used here is given in Table 6 of the appendix. For our automated evaluation, we used a much larger question dataset that comprised 1709 questions generated by GPT-3.5-turbo and 1850 questions generated by GPT-4 to give us a comparison between the two models.

4.2.1 Rule-based evaluation. We first calculated the IWF metrics scores using a set of rule-based code, for the larger automated

evaluation dataset described above. The results for the listed IWFs are shown in Table 2, with summary error statistics in Table 3.

Item Writing Flaw Diagnostic	GPT-3.5-turbo	GPT-4
Unclear information	97%	96%
Implausible Distractors	76%	83%
None of the above	97%	90%
Longest option correct	87%	82%
Gratuitous information	100%	100%
True/False question	99%	99%
Convergence cues	66%	63%
Logical cues	97%	98%
All of the above	100%	100%
Fill-in-the-blank	98%	98%
Absolute terms	85%	85%
Word repeats	98%	97%
Unfocused stem	100%	100%
Complex or K-type	98%	98%
Grammatical cues	72%	75%
Lost sequence	99%	99%
Vague terms	99%	99%
Negative worded	97%	97%
More than one correct	73%	97%

Table 2: Rule-based automated evaluation results (n=1850). Higher scores are better, indicating a question passes the check for that IWF and does not contain that flaw.

For all metrics above, a higher score means higher quality: most metrics display a very high score. Like human annotation scores, the distractor plausibility metric displays a lower score than other metrics, showing that the quality of the distractors in our questions can be improved: we are exploring a multi-step approach and refined prompt engineering to address this.

4.2.2 LLM based evaluation. The results of LLM-based quality evaluation are shown in Table 4, with summary error statistics in Table 5. We analyzed a representative sample of IWFs to avoid excessive computation costs, evaluating each MCQ item as a whole rather than specific parts. In general, questions generated by GPT-4 scored the same or higher across almost all question quality metrics compared to those generated by GPT-3.5, with lower average IQF failures per MCQ (1.86) compared to GPT-3.5 (2.18).

5 DISCUSSION AND CONCLUSION

We presented a comprehensive quality evaluation of a multiple-choice question dataset with over one thousand items that were generated by LLMs from video lecture transcripts. Using both human and automated assessment, we found that while the overall quality of the LLM-generated questions was generally good, less then 20% of questions were able to pass all quality metric tests, with an average of 1.6 Item Writing Flaws per MCQ for both GPT-3.5 and GPT 4-generated questions. While our quality findings were generally in accord with other recent studies of LLM-generated question quality, we also explored issues specific to the use of video transcripts, such as lack of appropriate contextual specificity. We defined and applied a new Item Writing Flaw evaluation method for detecting that problem. Potential future extensions include (1) hybrid correction models where a LLM is used to identify missing context, and a human expert provides additional edits if needed; and (2) using the qualitative reasoning output of LLMs as an additional diagnostic for detecting and correcting more challenging flaws such as lack of contextual specificity.

Statistic	GPT-3.5-turbo	GPT-4
Passes all metrics (%)	15	18
Passes at least half metrics (%)	100	100
Fails one or no metrics (%)	48	50
Fails two or fewer metrics (%)	80	79
Average IWF (failures) per MCQ	1.62	1.61

Table 3: Summary stats for rule-based evaluation (n=1850)

IWF Diagnostic	GPT-3.5-turbo	GPT-4
Unclear information	81%	83%
Implausible Distractors	43%	50%
Gratuitous information	84%	84%
Logical cues	69%	75%
Word repeats	69%	79%
Unfocused stem	67%	68%
Grammatical cues	86%	92%
Contextual Specificity	83%	83%

Table 4: LLM-based automated evaluation results (n=100). Higher scores are better, indicating a question passes the check for that IWF and does not contain that flaw.

Statistic	GPT-3.5-turbo	GPT-4
Passes all metrics (%)	25	22
Passes at least half metrics (%)	87	91
Fails one or no metrics (%)	46	55
Fails two or fewer metrics (%)	63	69
Average IWF (failures) per MCQ	2.18	1.86

Table 5: Summary stats for LLM-based evaluation (n=100)

Our study could be extended in a number of ways. We had high-quality video transcripts, which eliminated the need for additional error correction or noise removal. For transcript systems with higher Word Error Rates (WER), it would be critical to understand how sensitive question quality metrics are to changes in WER and to devise robust strategies for mitigating or correcting noise in the source content. Our video data was limited to one degree program and one particular STEM domain in English. A more complete evaluation would look at samples from a diverse variety of lectures in different domains and languages. Formative assessments that give meaningful adaptive feedback could also be integrated into our generation framework. Finally, the difference in automated and human evaluation makes it difficult to cross-compare them: a future unified rubric for both automated and human evaluation will help us better understand the efficacy of AI-based evaluations, and whether AI can be a viable alternative to human evaluators.

Acknowledgements. We thank the reviewers for their comments. This research was sponsored in part by a grant from the Michigan Institute for Data Science (MIDAS), with additional support from the University of Michigan School of Information.

REFERENCES

[1] Meghana Moorthy Bhat, Rui Meng, Ye Liu, Yingbo Zhou, and Semih Yavuz. 2023. Investigating Answerability of LLMs for Long-Form Question Answering. *arXiv preprint arXiv:2309.08210* (2023). arXiv:2309.08210 [cs.CL] https://arxiv.org/abs/2309.08210

[2] Ayan Kumar Bhowmick, Ashish Jagmohan, Aditya Vempaty, Prasenjit Dey, Leigh Hall, Jeremy Hartman, Ravi Kokku, and Hema Maheshwari. 2023. Automating question generation from educational text. *arXiv preprint arXiv:2309.15004* (2023). arXiv:2309.15004 [cs.CL]

[3] BHH Cheung, GKK Lau, GTC Wong, EYP Lee, D Kulkarni, CS Seow, R Wong, and MT Co. 2023. ChatGPT versus human in generating medical graduate exam multiple choice questions-A multinational prospective study (Hong Kong S.A.R., Singapore, Ireland, and the United Kingdom). *PLoS One* 18, 8 (2023). https://doi.org/10.1371/journal.pone.0290691

[4] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[5] Yogesh K Dwivedi, Nir Kshetri, Laurie Hughes, Emma Louise Slade, Anand Jeyaraj, Arpan Kumar Kar, Abdullah M Baabdullah, Alex Koohang, Vishnupriya Raghavan, Manju Ahuja, et al. 2023. "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management* 71 (2023), 102642.

[6] Sabina Elkins, Ekaterina Kochmar, Jackie C. K. Cheung, and Iulian Serban. 2023. How Useful are Educational Questions Generated by Large Language Models? *arXiv preprint arXiv:2304.06638* (2023). arXiv:2304.06638 [cs.CL]

[7] Hannah Gonzalez, Jiening Li, Helen Jin, Jiaxuan Ren, Hongyu Zhang, Ayotomiwa Akinyele, Adrian Wang, Eleni Miltsakaki, Ryan Baker, and Chris Callison-Burch. 2023. Automatically Generated Summaries of Video Lectures May Enhance Students' Learning Experience. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. Association for Computational Linguistics, Toronto, Canada, 382–393. https://doi.org/10.18653/v1/2023.bea-1.31

[8] Marti A. Hearst. 1997. Text Tiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics* 23, 1 (1997), 33–64. https://aclanthology.org/J97-1003

[9] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. Promptmaker: Prompt-based prototyping with large language models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. ACM, New Orleans, USA, 1–8.

[10] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education* 30 (2020), 121–204.

[11] Steven Moore, Huy A. Nguyen, Tianying Chen, and John Stamper. 2023. Assessing the Quality of Multiple-Choice Questions Using GPT-4 and Rule-Based Methods. *arXiv preprint arXiv:2307.08161* (2023). https://arxiv.org/abs/2307.08161

[12] Lidiya Murakhovska, Chien-Sheng Wu, Philippe Laban, Tong Niu, Wenhao Liu, and Caiming Xiong. 2021. Mixqg: Neural question generation with mixed answer types. *arXiv preprint arXiv:2110.08175* (2021).

[13] Xingdi Yuan, Tong Wang, Yen-Hsiang Wang, Emery Fine, Rania Abdelghani, Pauline Lucas, Hélène Sauzéon, and Pierre-Yves Oudeyer. 2022. Selecting Better Samples from Pre-trained LLMs: A Case Study on Question Generation. *arXiv preprint arXiv:2209.11000* (2022). arXiv:2209.11000 [cs.CL]

APPENDIX A: ERROR CATEGORIES, SAMPLE METADATA AND LLM PROMPTS

Extracted metadata

Sample additional non-question metadata extracted from a lecture on machine learning is given below in Table 7.

LLM Prompts

The prompts we used to generate/extract the data are below:

- **Summary**: Following is the transcript of a lecture: {lecture_text}. Please summarize this in a way that it can be used by a student to review the lecture and attempt questions.
- **Lecture Segmentation**: Please split the transcript of the lecture into a minimum of 3 and a maximum of 4 segments. A segment is a group of consecutive lines in the text where topics within the group are semantically similar to topics across all the sentences. Return the full text for each segment from the lecture. Return the segments in JSON format. The Segment number for each segment will be the key, and the text for the associated segment will be the value. Try to keep the size of each segment the same/similar. Every line in the lecture must fall in some segment.
- **Key Topics**: Please get me the five most important Machine Learning topics (in the form of very short phrases) that are discussed in {lecture_segment}. Only include topics from {wikipedia_set} Return the topics in the form of a JSON, with the key being "Concepts" and the list of topics being the value. Please do not include any extra text in the response
- **Key Definitions**: Please give me the key definitions, if any, that are discussed in {lecture_segment}. Give the definitions in JSON format, with the words as the keys and the definitions as the values. Please do not include any extra text or characters like the next line character in the response
- **Key Examples**: Please give me the explanation of the key examples (if any), used to explain a concept, in this text: {lecture_segment}. Give the response in the form of a JSON format. The key will be the name of the example - i.e., the text being referenced from the lecture, and the value will be the explanation. Please do not include any extra text like "Example <number>" or characters like the next line character in the response
- **Procedural Knowledge**: Please give me the "how to" explanations that are given, if any, in {lecture_segment}. Please directly start the response with the "how to" explanations and do not include any extra text in the response. Return the explanations in a JSON form The keys of the JSON should be "How to <the procedure>" and the value should be the explanation. Keep the explanation in the form of a paragraph.
- **Questions**: Please write a minimum of 10 and maximum of 15 unique multiple choice questions, with four choices each, from the text: {lecture_segment}. The questions will be used to test the knowledge of the students regarding the different concepts and examples covered in the text, hence the questions need to cover them. The questions should be good enough to be given in a technical exam. The questions need to be returned in a JSON format, with the keys being "Question <question number>" and values being another JSON. The sub-JSON containing the question data needs to be in this format:
  "Question": <The question statement>
  "A": <Option 1>
  "B": <Option 2>
  "C": <Option 3>
  "D": <Option 4>
  "Correct answer": <The correct answer A, B, C or D>
  "Explanation": <Explanation for the correct answer>

We gave specific formatting instructions to facilitate automated parsing and storage of the data. To make sure GPT was consistent with its output, we used the JSON key-value format for our data.

Sample evaluated questions

- Sample question that passed all human evaluation metrics:
  – Question: What is the main drawback of overfitting?

Item Writing Flaw	Description of desired question attribute
Unclear information	Questions and all options should be written in clear, unambiguous language.
Implausible Distractors	Make all distractors plausible as good items depend on having effective distractors.
None of the above	Avoid none of the above as it only really measures students ability to detect incorrect answers.
Longest option correct	Avoid having the correct option be longer and include more detailed information, since this may attract students to this option.
Gratuitous information	Avoid unnecessary information in the stem that is not required to answer the question.
True/False question	The options should not be a series of true/false statements.
Convergence cues	Avoid convergence cues in options where there are different combinations of multiple components to the answer.
Logical cues	Avoid clues in the stem and the correct option that can help test-savvy students identify the correct option.
All of the above	Avoid all of the above options as students can guess correct responses based on partial information.
Fill-in-the-blank	Avoid omitting words in the middle of the stem that students must insert from the options provided.
Absolute terms	Avoid the use of extreme absolute terms (e.g. never, always, all) in the options, as students are aware that these are almost always false.
Word repeats	Avoid similarly-worded stems and correct responses, or words repeated in the stem and correct response.
Unfocused stem	The stem should present a clear and focused question that can be understood and answered without looking at the options.
Complex or K-type	Avoid questions that have a range of correct responses, so that students need to select from a number of possible combinations of the responses.
Grammatical cues	All options should be grammatically consistent with the stem and should be parallel in style and form.
Lost sequence	All options should be arranged in chronological or numerical order.
Vague terms	Avoid the use of vague terms (e.g. frequently, occasionally) in the options as there is seldom agreement on their actual meaning.
Negative worded	Negatively worded stems are less likely to measure important learning outcomes and can confuse students.
More than one correct	In single best-answer form, questions should have exactly one best answer.

Table 6: Description of the IWF categories used in this study for automated evaluation (rule-based and LLM-based).

Metadata	Example
Procedural Knowledge	**How to create an ensemble model**: An ensemble model is created by combining multiple individual learning models to produce an aggregate model that is more powerful than any of its individual learning models alone. This is effective because different learning models, although each of them might perform well individually, they'll tend to make different kinds of mistakes on a data set. Typically, this happens because each individual model might overfit to a different part of the data. By combining different individual models into an ensemble, we can average out their individual mistakes to reduce the risk of overfitting while maintaining strong prediction performance.
Key concepts	Ensembles Bagging Boosting Random forest, Decision trees, Overfitting, Supervised Learning, Regression
Key definitions	**Ensembles**: A method in machine learning that involves creating learning models by combining multiple individual learning models to produce an aggregate model that is more powerful than any of its individual learning models alone. **Overfitting**: A modeling error in machine learning occurs when a function is too closely fit to a limited set of data points.
Key Examples	**Random Forests**: Random forests are given as an example of the ensemble idea applied to decision trees. They are widely used in practice and achieve very good results on a wide variety of problems. Random forests can be used as classifiers via the scikit learn random forest classifier class or for regression using the random forest regressor class both in the sklearn ensemble module. The use of random forests helps to overcome the disadvantage of using a single decision tree, which is prone to overfitting the training data.

Table 7: Sample (non-question) metadata from a machine learning lecture

- A: The model doesn't capture the trends in the data
- B: The model captures both the general trend and the noise in the data
- C: The model focuses too much on local variations
- D: The model generalizes well to test data

- Sample question that failed most human evaluation metrics:
  - Question: How many data set samples are present in the regression problem?
  - A: 10
  - B: 50
  - C: 100
  - D: 200
- Sample question that passed all automated evaluation metrics:
  - Question: What are the two main types of data leakage?
  - A: Leakage in the training data and leakage in features
  - B: Leakage in the testing data and leakage in labels
  - C: Leakage in the prediction data and leakage in algorithms
  - D: Leakage in the validation data and leakage in models
- Sample question that failed 6 important automated evaluation metrics:
  - Question: What is a common issue when fixing data leakage problems?
  - A: Data leakage problems are usually easy to fix and do not require much effort
  - B: Fixing one leaking feature can reveal the existence of a second one
  - C: Fixing data leakage problems often leads to a decrease in model performance
  - D: Data leakage problems are typically isolated and do not affect other features