DOI: 10.25968/opus-3451

Identifying Problem Types in Automated Question Generation

Joshua Berger*, Markos Stamatakis†, Anett Hoppe†‡, Ralph Ewerth†‡, Christian Wartena*
*Hochschule Hannover, Data|H – Institute for Applied Data Science Hannover
†TIB – Leibniz Information Centre for Science and Technology
‡Leibniz University Hannover, L3S Research Centre
joshua.berger@hs-hannover.de

***Abstract*—Automated question generation holds great promise in many fields, such as education, to reduce the workload and automate an otherwise tedious task. However, major challenges remain regarding the quality of generated questions. To identify and address these challenges generated questions are evaluated either automatically or manually. While several automated metrics, mostly based on the comparison with a gold standard, exist, their usefulness is limited and human evaluation is often used for more accurate assessment. Our research generates questions using several models and methods, including fine-tuning, zero-shot and few-shot. We compare model performance by classifying the generated questions using a multi-label approach. This approach evaluates by sorting generated questions into zero or more binary problem classes and attempting to identify different problems with the generated questions. Our results show that different models tend to generate questions that fit into different problem classes. Additionally, the problem classification evaluation is capable of recognizing these differences and weighing the classes for the models accordingly, creating model-specific distribution characteristics.**

***Index Terms*—Automated Question Generation, AQG, Problem Categorization, NLP, Transformers**

I. Introduction

One of the best ways to test understanding is by asking questions. This simple method is used in many fields, from education where students answer questions on tests, to workers who answer questions after watching worker training videos. The creation of these questions often falls to highly specialized people who know how to ask meaningful questions and who know the source material well enough to be able to ask fitting questions. This is a time-consuming endeavor that often wastes the time of people who are overqualified for this task.

To solve this problem questions can be generated automatically. The recent advancements in natural language generation, fuelled by the transformer architecture [1], have led to a significant leap in automated question generation research. While most transformer-based models are not specifically trained for question generation, their large pre-training base allows for easy fine-tuning to fit them to the task of question generation. Additionally, chat models like ChatGPT can be simply asked to generate questions (zero-shot approach) or be shown just a few examples to generate the desired output (few-shot approach). However, the problem of evaluation remains. Automatic metrics for question generation have the problem that they are mostly based on candidate-reference comparisons that assume a ground-truth solution [2], [3]. While this can be a useful question quality indicator, the ground-truth question is often not the only one that is a good fit. To mitigate this problem in automatic question generation, questions are often also evaluated manually [4]. This leads to a more accurate assessment of question quality but is several magnitudes more time and resource-intensive. Most human evaluation methods have annotators rate the question on a Likert scale regarding different aspects like *relevancy* [5] or *correctness* [6]. To the best of our knowledge an evaluation metric that sorts questions into binary problem classes, like the approach used in our work, does either not exist or has not found widespread use in popular question generation research.

Our research generates questions by using six different models based on the transformer architecture and utilizing three different methods for question generation (i.e. fine-tuning, zero-shot, and few-shot). We evaluate by having one annotator sort the output of question generation models into different binary problem classes, with each class attempting to identify a different aspect that might be wrong with the generated question. In doing so, we try to find useful problem classes that identify core issues to understand the limits of automated question generation better. Our work contributes a comparison of multiple different methods for automated question generation. Especially the comparison of zero-shot and few-shot with fine-tuned models provides insight into the feasibility of using pre-trained models out-of-the-box.

II. Datasets and Models

We use a total of three different datasets: two for training and one for testing. The datasets used are the following:

- **Stanford Question Answering Dataset (SQuAD)** [7] consists of 100.000 text-question-answer triples, the texts are paragraphs from Wikipedia articles, while the question and answer are human-generated questions and answers for these paragraphs.
- **Tutorial Video Question Answer (TutorialVQA)** [8] consists of text-question-answer triples, the texts are audio transcripts of internet tutorial videos for software (e.g. Adobe Photoshop), while the question is human-generated

Published in: H. Homann, C. Rohbani, and J. C. Will, Eds., KI-Forum 2024: AI 4 Students – AI in Research – AI Showroom. Hannover: HsH Applied Academics, 2024. Pp. 34–37.

and the answers consist of spans indicating where the answer is in the text.

- **Technische Informationsbibliothek AV-Portal Top 100 (TIB-AV-100)** (https://av.tib.eu/) consists of audio transcripts from a diverse set of educational videos ranging from documentaries to recordings of lectures or TED talks.

While SQuAD and TutorialVQA include human-generated questions and answers, TIB-AV-100 only contains audio transcripts. The models use SQuAD and TutorialVQA as a training basis. The datasets are split into training and validation sets as recommended by the dataset creators. TIB-AV-100 is only used for evaluation. We use a distinct dataset for evaluation to test the models on unfamiliar data and better mimic a real-life use case. Overall we distinguish between the following models:

- **BART-SQuAD** is a BART [9] model fine-tuned on the SQuAD dataset with the Wikipedia paragraph as input and the question as the target output.
- **BART-SQuAD-QA** is a BART model fine-tuned on the SQuAD dataset with the Wikipedia paragraph as input and a question-answer pair as target output.
- **BART-TutorialVQA** is a BART model fine-tuned on the TutorialVQA dataset with the audio transcript as input and the question as the target output.
- **BART-TutorialVQA-QA** is a BART model fine-tuned on the TutorialVQA dataset with the audio transcript as input and a question-answer pair as the target output.
- **Mistral7B-Instruct-Zero-Shot** is the chatbot version of the Mistral7B [10] model that uses a zero-shot prompt as the input.
- **Mistral7B-Instruct-Few-Shot** is the chatbot version of the Mistral7B model that uses a few-shot prompt as the input.

While the BART models were fine-tuned by simply using the texts as input, the Mistral models were given prompts to obtain a question as output. The following prompt was used for the zero-shot model:

> *"Generate a question about the most important aspect of the following text: TEXT INSERTED HERE"*

The few-shot model on the other hand used five examples of texts and questions, taken from the SQuAD dataset, forming the following command:

> *Few-shot examples + "Now generate a short question for the following text: TEXT INSERTED HERE".*

Since a one-shot test run has shown that the model emulates the example in a way that tends to copy the question word from the example, each example question for the few-shot model was chosen so that it starts with a different question word. The QA versions of the BART model were chosen to see whether giving an answer as part of the output increases the likelihood of generating a question that can be answered and focuses on asking about important text aspects.

III. Experimental Setup

A. Model Training

To train the models we mostly used commonly accepted standard values for the hyperparameters. We trained the BART models for five epochs each with a learning rate of $2e-5$, 500 warmup steps, and a batch size of four. The loss was calculated using cross-entropy loss using the Adam optimizer. The context window for the input was 512 tokens. The training was done on an NVIDIA RTX3090 24GB GPU. For evaluation, the epoch with the lowest validation loss was used. Since we used a zero-shot/few-shot approach for the Mistral7B model fine-tuning was not necessary. To infer these models we used 4bit quantization to be able to fit them on an RTX3090 GPU.

B. Question Generation

To set up the *TIB-AV-100* dataset for question generation, we first broke each video transcript down into smaller paragraphs. For this, we separated the transcripts into chunks of 106 words, as this is the average length of a paragraph from the SQuAD dataset. To not start or end on cut-off sentences, each incomplete sentence at the beginning or end was fully added. This leads to a minimal overlap in chunks, the end sentence of one chunk could be the beginning sentence of the next. Due to the even chunking based on word length, the last chunk in each video script varies greatly in length and could be much shorter than the aforementioned average.

Using this method to split the video transcripts into chunks resulted in 4,717 short video transcript chunks, forming the test set for the models used. Each model was fed the entire test data set, leading to 28,302 generated questions.

C. Evaluation Metric and Annotation

To evaluate we deviated from the more typical approach of rating the generated questions on a Likert scale regarding aspects like relevancy or correctness. Instead, in an initial manual inspection of some of the questions generated, we identified ten typical recurring issues that serve as problem classes. We use this type of metric to get a more detailed breakdown of issues with generated questions than possible with a broader Likert scale analysis. These issues were split into five broader categories after annotation:

1) Does not fit the text
    a) Can't be answered due to missing information in the text
    b) Irrelevant or off-topic
2) Can't be understood without reference to the text (e.g.: *What theory did he publish?* instead of *What theory did Charles Darwin publish?*)
3) Nonsensical
4) Asks for more than one thing or is vague
5) Unnatural or wrong phrasing
    a) Grammatically incorrect
    b) Unnecessarily long
    c) Wrong pronoun (e.g.: *What medication did he take before giving birth?*)

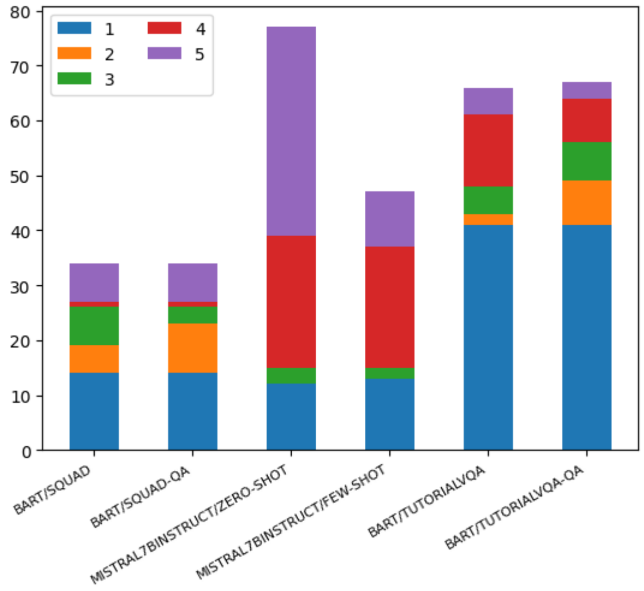

Fig. 1. Problem class annotation split by model. We annotated 324 questions (54 per model) in total. (self-created)

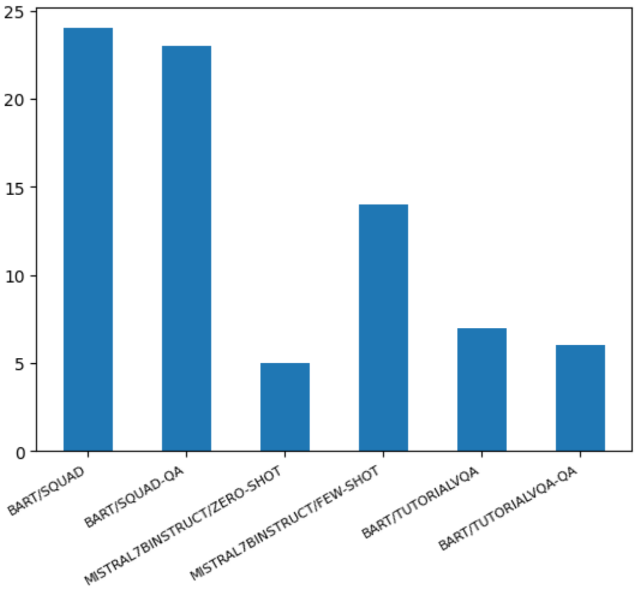

Fig. 2. Amount of questions that were not categorized in either of the ten classes. We annotated 324 questions (54 per model) in total. (self-created)

d) Wrong question word
e) Unusual wording

The error classes are not mutually exclusive. In total we annotated the questions for 54 video transcripts. Since six questions were generated for each transcript (one per model) a total of 324 questions were annotated by the authors. These manual annotations can be found in Fig. 1.

IV. Results

The generated questions differ greatly depending on the model and training source. Several examples can be seen in Table I. These examples quite effectively illustrate the differences in question generation encountered throughout the annotated subset. The questions generated by the Mistral7B-Instruct model (and especially the zero-shot variant) tend to be very long and semantically complicated questions that often directly repeat text information or directly reference the text (e.g. *"[...] as described in the text, [...]"*). The few-shot model still tends to do this but in a manner that is decidedly less extreme than the behavior exhibited by the zero-shot model. The few-shot examples seem to have directed the model in the right direction. The models fine-tuned on TutorialVQA almost exclusively generated questions in the manner displayed in the example. This was expected, as the dataset itself mostly includes questions asked in this manner. Although the use of the singular pronoun *you* may be considered the optimal choice for this question, the pronoun *I* was deemed acceptable and not subjected to any penalties by the annotator.

In Fig. 1 we break down the problem classes by model. As can be seen in the figure, the most obvious difference between models occurs in class five, which, in the case of the Mistral models, mostly consists of *unnecessarily long* questions. As the example shows, Mistral particularly suffers from this, because this model tends to generate long and complicated questions. This also coincides with class four, which consists of questions *that ask for more than one thing or are very vague*. The longer and more complicated questions of the Mistral models demand a longer and more complex answer. While the SQuAD-trained models and Mistral models have about the same amount of questions that are *irrelevant or could not be answered*, the TutorialVQA-trained models have a far higher number. This indicates that the format of the questions generated by the TutorialVQA-trained models does not lend itself very well to question generation. An advantage of the long Mistral questions is that they rarely rely on text knowledge to make the question understandable, while the BART models have at least some questions that rely on text knowledge.

Fig. 2 shows the number of questions that were not categorized in either of the ten classes. It is important to note that this does not mean that these questions are flawless but only that they do not fit in any of the proposed problem classes. Although these classes were chosen to encompass many of the problems that arise with automatically generated questions, they are not all-encompassing and there might still exist small imperfections in the questions counted in Fig. 2. The figure shows that the questions generated by the SQuAD-trained had by far the highest amount of uncategorized questions, while the zero-shot Mistral model had the least amount. Interestingly enough, the few-shot Mistral model had almost triple the amount of uncategorized questions, indicating that the few-shot prompt approach was helpful to the model and nudged the question direction in the intended direction.

TABLE I

EXAMPLE OF GENERATED QUESTIONS FOR THE DIFFERENT MODELS. ALL EXAMPLE QUESTIONS ARE BASED ON THE SAME VIDEO TRANSCRIPT.

Model	**Generated Question**
BART-SQuAD	What is used to quickly switch the ACP VCC supply?
BART-SQuAD-QA	How many outputs does the benchtop DC power supply have?
Mistral7B-Instruct-Zero-Shot	What is the role of the 74 series 4053 analog switch and the benchtop DC power supply in controlling the normal level and glitch level voltages for communication with the ACP via SPI on the X-megaboard, and how are they connected on the breadboard?
Mistral7B-Instruct-Few-Shot	What components are used on the breadboard to switch the ACP's voltage supply and provide necessary clock inputs?
BART-TutorialVQA	how do i change the glitch level?
BART-TutorialVQA-QA	how do i control the glitch?

V. CONCLUSION AND FUTURE WORK

Our work utilizes different methods and models to generate questions and evaluates them by sorting them into problem classes rather than rating them on a Likert scale. This evaluation method was used to annotate questions generated by six models that differ in architecture and training data. The models generated vastly different questions, each with their own set of problems. Our results show that the method is capable of detecting differences in the generated questions per model by showing problems a model might have based on the number of questions per class. Few-shot and fine-tuned model outputs differed mostly in length and precision.

While this approach is promising, there is still work that remains to be done. Zero-shot and few-shot performance could be greatly improved by utilizing prompt engineering to optimize the prompt for the task. Additionally, the classes introduced in our research were chosen by evaluating only a small number of generated questions. A further and deeper analysis might lead to more accurate and refined classes. Another interesting approach would be to attempt an automatization of this method by trying to train a classifier on these classes to classify questions automatically which would lead to a much bigger data set.

ACKNOWLEDGMENT

This work was funded by the Lower Saxony Ministry of Science and Culture with funds from the zukunft.niedersachsen program of the VolkswagenStiftung as part of the VidQA project.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[2] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 74–81.

[3] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, 2002, pp. 311–318.

[4] J. Amidei, P. Piwek, and A. Willis, "Evaluation methodologies in automatic question generation 2013-2018," in *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, 2018, pp. 307–317.

[5] X. Jia, W. Zhou, X. Sun, and Y. Wu, "EQG-RACE: examination-type question generation," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 2021, pp. 13 143–13 151.

[6] Z. Liu, K. Huang, D. Huang, and J. Zhao, "Semantics-reinforced networks for question generation," in *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, 2020, pp. 2078–2084.

[7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016, pp. 2383–2392.

[8] A. M. Colas, S. Kim, F. Dernoncourt, S. Gupte, D. Z. Wang, and D. S. Kim, "Tutorialvqa: Question answering dataset for tutorial videos," in *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, 2020, pp. 5450–5455.

[9] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2020, pp. 7871–7880.

[10] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," *CoRR*, vol. abs/2310.06825, 2023.