

QGen Studio: An Adaptive Question-Answer Generation, Training and Evaluation Platform

Movina Moses^{*1}, Mohab Elkaref^{*2}, James Barry², Shinnosuke Tanaka², Vishnudev Kuruvanthodi², Nathan Herr^{3†}, Campbell D Watson^{1,2}, Geeth De Mel²

¹IBM Research

²IBM Research Europe

³University College London

{movina.moses, mohab.elkaref, james.barry, shinnosuke.tanaka, vishnudev.k}@ibm.com, uceenhe@ucl.ac.uk, cwatson@us.ibm.com, geeth.demel@uk.ibm.com

Abstract

We present QGen Studio: an adaptive question-answer generation, training, and evaluation platform. QGen Studio enables users to leverage large language models (LLMs) to create custom question-answer datasets and fine-tune models on this synthetic data. It features a dataset viewer and model explorer to streamline this process. The dataset viewer provides key metrics and visualizes the context from which the QA pairs are generated, offering insights into data quality. The model explorer supports model comparison, allowing users to contrast the performance of their trained LLMs against other models, supporting performance benchmarking and refinement. QGen Studio delivers an interactive, end-to-end solution for generating QA datasets and training scalable, domain-adaptable models. The studio will be open-sourced soon, allowing users to deploy it locally.

Introduction

Large language models (LLMs) have shown remarkable capabilities in text generation, enabling the creation of large synthetic datasets with minimal human input. This offers significant potential for tasks like question-answering (QA) (Duan et al. 2017), where access to high-quality data is crucial for training models. However, niche domains often lack tailored datasets, causing LLMs to perform poorly, as generic datasets may not align with specialized tasks (Ling et al. 2024).

While LLMs can help mitigate this issue by generating domain-specific datasets (Wei et al. 2022), the quality and relevance of these datasets can vary, complicating their alignment for specialized tasks (Gudibande et al. 2023). The lack of standardized evaluation metrics further challenges their effectiveness (Nema and Khapra 2018). There is a need for better methods to evaluate and ensure these datasets tailor to requirements.

To address these challenges, we present QGen Studio, an adaptive platform for question-answer generation, training, and evaluation. Users can upload documents and interactively generate QA pairs using LLMs from OpenAI, IBM

watsonx, and HuggingFace using Apple’s MLX framework. The platform provides an interactive interface that allows users to input example prompts while viewing the source text, giving them more control over the QA generation process. Generated QA pairs are visualized using a dataset viewer, highlighting metrics and contextualizing the data for better analysis.

QGen Studio also enables users to train models using the generated datasets and compare the performance of these custom models against existing ones. The studio supports a streamlined, end-to-end solution for synthetic QA dataset creation, visualization, and model refinement by addressing limitations of existing approaches and allowing users to build scalable, domain-adaptable QA models.

Related Work

The use of LLMs for generating synthetic data has been widely explored, demonstrating their potential in creating and augmenting datasets for NLP tasks such as question-answering (QA) (Long et al. 2024; Brown et al. 2020). Methods using rule-based generation (Khullar et al. 2018), pre-trained models (Dong et al. 2019), and adversarial training (Rao and Daumé III 2019) have been explored, but they often depend on existing datasets and struggle to adapt to niche domains with limited data emphasizing the need for more flexible, domain-adaptive QA generation methods. Additionally, the quality and relevance of these generated datasets can be inconsistent (Yu et al. 2023).

While several platforms and tools have been developed to support dataset creation and model training (Patel, Raffel, and Callison-Burch 2024; Yu, Lu, and Yu 2024), they often limit user control over the generation process, which hinders effective exploration and understanding of the datasets.

QGen Studio addresses these limitations and tailors to the needs of niche domains with limited data availability where users would like more control over the generation process.

System Overview

QGen Studio follows a six-step pipeline that guides the user through the process of creating their datasets, training their models and evaluating its performance. In this section, we describe the technologies used in each step.

^{*}These authors contributed equally.

[†]Work done while at IBM.

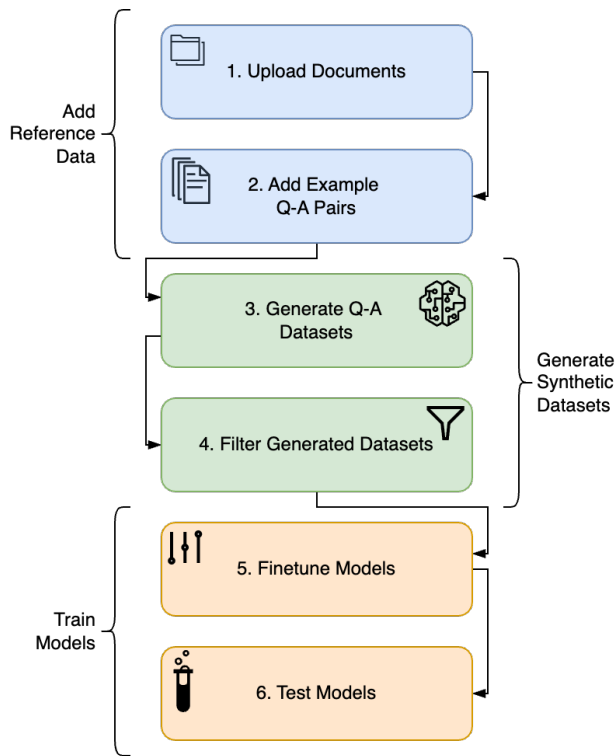


Figure 1: Overview of QGen Studio

Document Upload The document upload section allows users to create document groups under which they can upload various context sources such as JSON files, PDFs and URLs to PDFs. PDFs and URLs are automatically converted to structured text using Docling (Auer et al. 2024), split into headings and paragraphs and assigned unique IDs. The files are organized as documents within their respective document groups. The interface allows users to add, view and delete documents within each group.

Example Prompts The example prompts page allows users to scroll through documents within each group and inspect, add, or delete question-answer pairs. These pairs can be used as examples within few-shot prompts to guide dataset generation, providing more control over the type of QA pairs generated from a specific document.

Dataset Generation In the dataset generation step, users generate QA pairs from selected document groups using LLMs from OpenAI¹ and IBM’s watsonx², or HuggingFace models via Apple’s MLX library (Hannun et al. 2023). After selecting an LLM and document groups, users configure generation parameters like text chunking, number of questions, and evaluation metrics to use for ranking. Available metrics include computational metrics such as Bleu, Rouge, and Meteor, as well as TFIDF and cosine similarity scores, all calculated between the generated QA pairs and their source context. Users can choose between zero-shot or

¹<https://platform.openai.com/docs/overview>

²<https://www.ibm.com/watsonx>

few-shot prompts, with few-shot examples drawn from the Example Prompts page. Once configured, QA pairs are generated, evaluated, and stored by document group.

Dataset Viewer The dataset viewer enables users to view and interact with the generated datasets. Users can select a document group, choose evaluation metrics, and highlight either the question, answer, or context. The spans in the context are highlighted based on question and answer tokens, and if “context” is chosen, the viewer highlights the most likely sentence from which the QA pair was generated. Users can filter and sort QA pairs by metrics, such as showing only pairs with Bleu-2 scores above 0.8. The viewer lets users visualize QA pairs for each document. If desired, they can adjust parameters, metrics, or examples to refine the generation process.

Model Training The model training section enables users to fine-tune models to better adapt them to the generated datasets. This process utilizes MLX to fine-tune an LLM with low-rank adaptation (LoRA) (Hu et al. 2021). Currently, this feature is limited to Apple Silicon users, but others can download the generated datasets and train models independently. Users can choose how the dataset is processed, including options to include context, shuffle the dataset, and adjust the test and validation splits. They can modify training parameters, such as the learning rate, number of iterations, and LoRA layers. This setup allows users to utilize one model for generation and a different model for training, enhancing customization and performance for specific tasks.

Model Explorer The model explorer page allows users to compare two models side by side. Users can select a document group, document, and models they wish to evaluate. They can choose between existing models and their fine-tuned models. While viewing the document, users can pose a question related to its content, and the studio will run inference on the selected models. The explorer displays the generated responses side by side, allowing users to compare and contrast the models’ performance. They can assess how the generated datasets influenced the models’ by reviewing the answers in the context of the selected document. This layout helps users evaluate the strengths and weaknesses of each model, facilitating informed decisions about which model is better aligned for use.

Conclusion and Future Work

In this paper, we introduced QGen Studio, a comprehensive framework for generating and evaluating question-answer pairs using various models and custom datasets. This tool can be used to create QA datasets across different domains and adapt models to diverse requirements. Users can explore their datasets and fine-tuned models while visualizing performance, enabling them to make necessary adjustments in the generation and training processes.

Future work will focus on supporting additional frameworks and enabling multi-document querying for insights across diverse sources. We also plan to incorporate generation for complex question types, support multilingual generation, and extend functionality for other downstream tasks.

References

- Auer, C.; Lysak, M.; Nassar, A.; Dolfi, M.; Livathinos, N.; Vagenas, P.; Ramis, C. B.; Omenetti, M.; Lindlbauer, F.; Dinkla, K.; Mishra, L.; Kim, Y.; Gupta, S.; de Lima, R. T.; Weber, V.; Morin, L.; Meijer, I.; Kuropiatnyk, V.; and Staar, P. W. J. 2024. Docling Technical Report. arXiv:2408.09869.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.
- Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. arXiv:1905.03197.
- Duan, N.; Tang, D.; Chen, P.; and Zhou, M. 2017. Question Generation for Question Answering. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 866–874. Copenhagen, Denmark: Association for Computational Linguistics.
- Gudibande, A.; Wallace, E.; Snell, C.; Geng, X.; Liu, H.; Abbeel, P.; Levine, S.; and Song, D. 2023. The False Promise of Imitating Proprietary LLMs. arXiv:2305.15717.
- Hannun, A.; Digani, J.; Katharopoulos, A.; and Collobert, R. 2023. MLX: Efficient and flexible machine learning on Apple silicon.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685.
- Khullar, P.; Rachna, K.; Hase, M.; and Shrivastava, M. 2018. Automatic Question Generation using Relative Pronouns and Adverbs. In Shwartz, V.; Tabassum, J.; Voigt, R.; Che, W.; de Marneffe, M.-C.; and Nissim, M., eds., *Proceedings of ACL 2018, Student Research Workshop*, 153–158. Melbourne, Australia: Association for Computational Linguistics.
- Ling, C.; Zhao, X.; Lu, J.; Deng, C.; Zheng, C.; Wang, J.; Chowdhury, T.; Li, Y.; Cui, H.; Zhang, X.; Zhao, T.; Panalkar, A.; Mehta, D.; Pasquali, S.; Cheng, W.; Wang, H.; Liu, Y.; Chen, Z.; Chen, H.; White, C.; Gu, Q.; Pei, J.; Yang, C.; and Zhao, L. 2024. Domain Specialization as the Key to Make Large Language Models Disruptive: A Comprehensive Survey. arXiv:2305.18703.
- Long, L.; Wang, R.; Xiao, R.; Zhao, J.; Ding, X.; Chen, G.; and Wang, H. 2024. On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics ACL 2024*, 11065–11082. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics.
- Nema, P.; and Khapra, M. M. 2018. Towards a Better Metric for Evaluating Question Generation Systems. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3950–3959. Brussels, Belgium: Association for Computational Linguistics.
- Patel, A.; Raffel, C.; and Callison-Burch, C. 2024. DataDremer: A Tool for Synthetic Data Generation and Reproducible LLM Workflows. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3781–3799. Bangkok, Thailand: Association for Computational Linguistics.
- Rao, S.; and Daumé III, H. 2019. Answer-based Adversarial Training for Generating Clarification Questions. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 143–155. Minneapolis, Minnesota: Association for Computational Linguistics.
- Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652.
- Yu, X.; Lu, Y.; and Yu, Z. 2024. LocalRQA: From Generating Data to Locally Training, Testing, and Deploying Retrieval-Augmented QA Systems. In Cao, Y.; Feng, Y.; and Xiong, D., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, 136–151. Bangkok, Thailand: Association for Computational Linguistics.
- Yu, Y.; Zhuang, Y.; Zhang, J.; Meng, Y.; Ratner, A.; Krishna, R.; Shen, J.; and Zhang, C. 2023. Large Language Model as Attributed Training Data Generator: A Tale of Diversity and Bias. arXiv:2306.15895.