**Article information**

**Article title**

*University Information chatbot: A generative AI-based chatbot for university information chatbot*

**Authors**

*Md. Shahidul Salim; Sk. Imran Hossain*

**Affiliations**

*Khulna University of Engineering & Technology*

**Corresponding author’s email address and Twitter handle**

*ss@cse.kuet.ac.bd*

**Keywords**

*LLM, Chatbot, University information*

**Abstract**

*Recently, the university information chatbot based on the large language model (LLM) has been increasing in popularity among people and can answer all university-related questions. We have proposed a framework to develop a university-intelligent information chatbot that can answer specific university-related questions and evaluate the performance of three models:Zepyhr, Mistral, and Llama2, with RAG (retrieval-augmented generation) techniques and ensemble retrievers, so that the model performance is good for this type of task. We have shown comparisons between different LLMs with RAG using our benchmark dataset. In addition, we proposed two user interfaces: one for gathering data from the students and the other for having the students evaluate the model. We found the Misral 7B model with the retriever ensemble to have shown better performance with 4.8.*

**Review Highlights**

*Include 1-3 bullet points that outline your review.*

**Graphical abstract**

*The graphical abstract should summarize the contents of your article in a concise, pictorial form. Graphical abstracts should be submitted as a separate file. Image size:****please provide one image with a minimum of 531 × 1328 pixels (h × w) or proportionally more****. The image should be readable at a size of 5 × 13 cm using a regular screen resolution of 96 dpi. Preferred file types: TIFF, EPS, PDF or MS Office files.*

**Specifications table**

*This table provides general information on the methodology you reviewed.*

|  |  |
| --- | --- |
| **Subject area** | *Please select a subject area from the list of* [*MEX categories*](https://legacyfileshare.elsevier.com/promis_misc/MethodsX-classifications.xlsx)*.* |
| **More specific subject area** | *Describe narrower subject area* |
| **Name of the reviewed methodology** | *Please write the name of the methodology you are reviewing in this article* |
| **Keywords** | *List specific keywords that are relevant to this review article* |
| **Resource availability** | *If applicable, list the databases validating the methods reviewed in the article.* |
| **Review question** | *Please formulate review questions\* and list here. (\*All the mentioned questions should be answered in this review)* |

**Background**

*Please describe your motivation and context behind providing this review (Max. 500 words).*

**Method details**

In the realm of artificial intelligence and natural language processing, the development of intelligent chat bots has gained significant momentum. A chatbot is a question-answering (QA) system, a specific type of information retrieval. Given a set of documents, a question-answering system attempts to find the correct answer in natural language**.**

Question Answering (QA) is a research area that combines research from different, but related, fields which are Information Retrieval (IR), Information Extraction (IE) and Natural Language Processing (NLP) \cite{1,2,3}. Question answering is a problematic form of information retrieval characterized by information needs that are at least somewhat expressed as natural language statements or questions. It was used as one of the most natural types of human-computer communication. For large language models, a good dataset is needed to identify context. Most of the chatbots are developed using pattern matching or a rule-based approach \cite{conv}. It differs from the data-driven question-answering system model, which can be created using data or conversation history that has been conducted, allowing for sufficient development to train the model with the available data \cite{lstm}.

According to some deep learning studies, neural network models produce results that are good enough to be utilized in a QA system. One of them achieves a good performance by using the sequence-to-sequence (Seq2Seq) approach \cite{nn1,nn2,nn3}.

Recently, some open-source large language models like Llama2, Mistral, and Zephyr have trained on huge amounts of data and understood the context correctly. It is easy to develop an intelligent chatbot for question answering, and furthermore, we can add RAG (retrieval augmented generation), which can increase the model for factual question answering.

Despite having open-source models, those models need to be fine-tuned to make them workable. Data collection, data curation, fine tuning, testing, and inference are needed to develop an intelligent question-answering system using LLM. But there is no software or framework to help develop this QA chatbot for technical or non-technical people. It is difficult to collect data individually, and furthermore, we need to preprocess and structure the data for fine tuning. After fine-tuning, it is important to check the model's performance, but there is no specific evaluation metric because the answers to questions depend on human evaluation. We added a human evaluation interface to collect user ratings about the models for specific questions provided by the user. In this method, we propose a framework and software to develop an intelligent chatbot using open-source models. Moreover, we apply this method to provide context using external sources; as a result, the user does not need to fine-tune the model each time, and easy-to-use software allows for directly applying the method without programming requirements.

How the software works

**Data collection.** The first tab, the data collection tab, is primarily used to collect information from the user. This tab is divided into two sub-tabs: existing questions and custom questions. In this first sub-tab, the user can answer the existing questions that are given by the administrator and then save the question and answer in an Excel file using the "save the answer" button. The user can generate a new question using "Generate a new question." In the second sub-tab, the user can enter custom questions and answers. Then save the answer using the "save the answer" button.

**Fine-tuning.** In the fine-tuning tab, three models are given for fine-tuning: Mistral, Zepyhr, and Llama2. First, we need to create an Excel file and name it data.xlsx. This file has two columns: Prompt and Reply. The data format is given in Table 1. Here, we need 24 GB of VRAM for fine-tuning using 8-bit quantization. We added a quantization technique for saving GPU memory. It does not lose the performance of the model. Training is mainly used to learn the context of university information. In our task, the model does not show significant performance in fine-tuning. Further, we add retrieval augmented generation(RAG) for a better answer. To train the model, we just need to click on the model’s name below. If anyone wants to edit the hyperparameter for the model, they click the edit option to change the hyperparameter.

|  |  |
| --- | --- |
| **Prompt** | **Reply** |
| What does KUET stands for? | KUET stands for Khulna University of Engineering & Technology. |
| How many halls in KUET? | KUET has a total of 7 halls: 6 for male students and 1 for female students. The largest hall, Amar Ekushey Hall, accommodates 563 students. |

Table 1: Finetune Data format

**Testing data generation.** This tab is mainly used for answer generation from the fine-tune model so that it helps the user or admin evaluate model performance. In this tab, the user first creates an Excel file named testing\_dataset.xlsx in the data folder. The user places the RAG data in the “rag\_data” folder because it helps the model generate a more accurate answer. The file format of the rag data is docx. Then, the user selects the model and clicks the “Generate the answer of the testing dataset” button. It needs time to generate the answer from the model. After this, the user or admin can evaluate the model in the human evaluation tab.

**Human evaluation.** The human evaluation tab is used to check the model's performance. In this case, we are not only checking the model performance but also the model with the rag technique. The first administrator determines the answer using a fine-tuned model. Then, this tab has a model name and no\_of\_question, which are given by admin. The user gives a rating to each question's answer, which is determined by the model. After clicking the "save" button, the question, answer, and rating are saved in an Excel file. Further, admin can take the file from the local folder and use it to find the performance of the model. This was evaluated directly by the user or students. "Save all the data in the dataframe" By clicking this button, the user can save all the data manually in a local folder. But after the save button, it always saves the data after 5 instances. There is also a "move to the question" button, which helps the user move to the question if they want to rate a particular question.

**Inference.** In the Inference tab, a chat interface is given. The administrator or user can select the model for inferencing. Three models are given for inferencing RAG data (Mistral, Zepyhr, and Llama 2). RAG data is placed in the "rag\_data" folder. We have to put the RAG in a docx file. The data format should be title and content format so that the retriever can easily find the relevant chunk from the vector database.

***Materials & Methods, Results sections are not included in a MethodsX review article****.*

**Conclusion**

*In this section you can provide the concluding remarks, take home messages etc. (Max. 250 words).*

**Ethics statements**

*MethodsX has ethical guidelines that all authors must comply with.*

***If your work involved data collected from social media platforms****, please include a statement here confirming that a) informed consent was obtained from participants or that participant data has been fully anonymized, and b) the platform(s)’ data redistribution policies were complied with:*

**CRediT author statement**

*CRediT is an initiative that enables authors to share an accurate and detailed description of their diverse contributions to a published work. You can use the categories listed* [*here*](https://www.elsevier.com/researcher/author/policies-and-guidelines/credit-author-statement)*.*

**Acknowledgments**

*All contributors who do not meet the criteria for authorship should be listed in an acknowledgments section.*

*In addition, please list any funding sources in this section. List funding sources in this standard way to facilitate compliance to funder's requirements:*

*If no funding has been provided for the research, please include the following sentence:*

*This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.*

**Declaration of interests**

*Please* ***tick*** *the appropriate statement below (please do not delete either statement) and declare any financial interests/personal relationships which may affect your work in the box below.*

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

*Please declare any financial interests/personal relationships which may be considered as potential competing interests here.*

**Supplementary material *and/or* additional information [OPTIONAL]**

*You may also submit supplementary material with your article. This is not compulsory. If you do submit supplementary files, you are welcome to provide supporting details in this OPTIONAL section. More information is available in the* [*Guide for Authors*](https://www.sciencedirect.com/journal/methodsx/publish/guide-for-authors)*.*

**References**

*Please list all the articles (max. of 100) that are reviewed here.*

***Reminder: Before you submit, please delete all the instructional text in dark blue, wherever it appears – including here.***

***Thank you!***