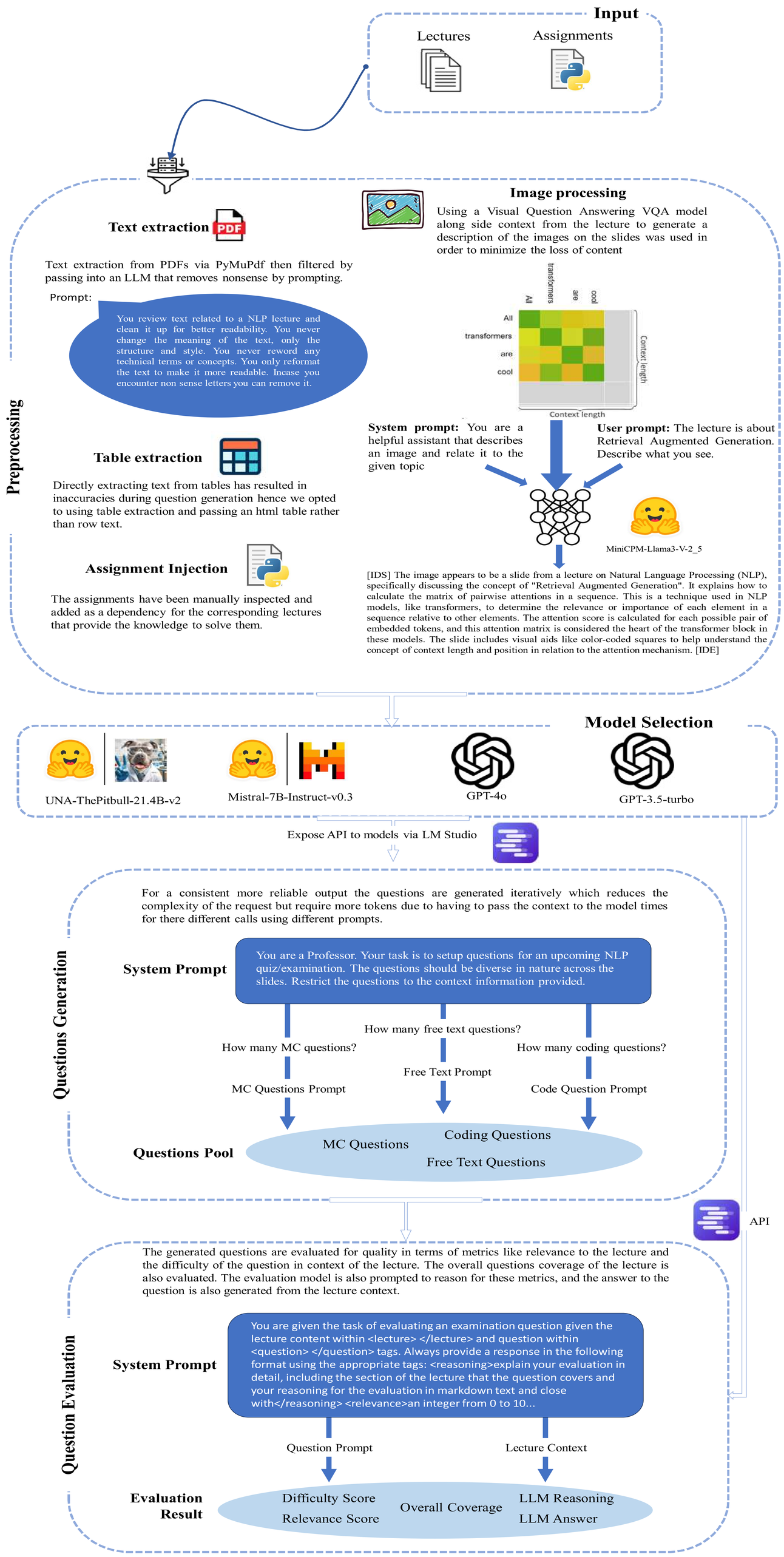
Automatic Exam Question Generation (Lecture2Exam)

Mohammad Saknini, Darius Arbabha, Shinas Shaji

Workflow

.



Motivation

Creating exam questions is tedious for the professors and often time-consuming, detracting from their primary responsibilities of teaching and research. Developing a large language model (LLM) pipeline to generate exam questions from lecture notes aims to streamline this process, enhancing efficiency and accuracy. This automated solution ensures comprehensive coverage of course material, producing high-quality, consistent questions. It allows customization to suit different question formats and difficulty levels. Ultimately, this pipeline should enable educators to focus more on student engagement and the quality of instruction, rather than stressing about the exam questions.

Pipeline

The proposed LLM pipeline for generating exam questions from lecture notes encompasses several key stages: preprocessing, model selection, question generation, and question evaluation.

Preprocessing The preprocessing stage involves converting various input formats, such as PDFs, images, and structured data, into a clean text format. The text is carefully cleaned up for better readability while preserving the meaning and technical terms. This step is crucial for ensuring that the input data is well-organized and ready for accurate processing by the model. Text is reformatted for clarity, with non-sensical letters removed to improve the overall quality of the input. Note: Essentially we planned to do use Retrieval Augmented Generation (RAG) approach, but it seemed unreasonable to group up the lectures to then split them up again, hence we decided against it.

Model Selection The models were based on two main factors:

1. **Computational efficiency**: The main problem in LLMS currently is that they require a large amounts of compute, which was a limiting factor when choosing models.
2. **Context length**: For smaller models a typical context length as 16k tokens, but for some lecture we required up to 25k in order to fit the content in one context window.

Additionally Fraunhofer INT has agreed to provide us with an API key for the purpose of this project in order to test the capabilities of larger close sourced models such as GPT3.5-Turbo and Omni.

Question Generation During question generation, the selected model is tasked with creating a diverse set of questions based on the lecture content. The process is initiated by a prompt given to the model, specifying the type and number of questions required. The model is able to generate multiple-choice questions, coding questions, and free-text questions, ensuring a comprehensive assessment that covers various cognitive skills and difficulty levels.

Example multiple choice question:

What is the primary purpose of training language models with human feedback according to the lecture content?

A) To make the models align with their users' preferences.
B) To reduce the cost of manual data labeling.
C) To increase the complexity of large prompt datasets.
D) To automate the process of fine-tuning language models.

Question Evaluation

The final stage involves evaluating the generated questions to ensure their relevance, difficulty, and overall coverage of the lecture material. This evaluation is performed by assessing the alignment of each question with the lecture content, using specific criteria such as difficulty score, relevance score, and overall coverage. The model's reasoning is also reviewed to confirm the appropriateness and clarity of the questions. Detailed answer is also provided, which allows us to understand the reason behind the models decisions.

Contact

Saknini Mohammad, Darius Arbabha, Shinas Shaji
Hochschule Bonn-Rhein-Sieg
Email: firstname.lastname@smail.inf.h-brs.de

—Github Repository—

Acknowledgement

Thanks to Prof. Dr. Jörn Hees and Tim Metzler for the materials and the insightful lecture **Natural Language Processing**.
We want to also thank Fraunhofer INT for allowing us to use an OpenAI API key for this project.

b-it Bonn-Aachen International Center for Information Technology

Hochschule Bonn-Rhein-Sieg
University of Applied Sciences