UtilitasMathematica

ISSN 0315-3681 Volume 122, 2025

Automated Educational Quiz System Using RAG for Student Assessment

Sri Charan Maheswarla, Ajay Varma Kammampati, Sarath Sai Jupalli
Dr. P Satyanarayana
Department of IoT, Koneru Lakshmaiah Educational Foundation
kammampatiajayvarma@gmail.com, sarathsai4602@gmail.com, mrajvesh@gmail.com

***Abstract*— Searching and finding relevant information through text books and other sources has always been a hassle for students. Ironically, text books are one of the best sources for students to find appropriate information. We wanted to build a system that could fetch the necessary data from text sources precisely and also help students to test what they have learnt. So, DocVQA is one way in which we could implement this. Document visible question answering (DocVQA) pipelines, which reply to enquiries derived from documents, has many makes use of. Current methodologies concentrate on handling single-web page documents the usage of multi-modal language fashions (MLMs) or rely on text-based totally retrieval-augmented technology (RAG) that use text extraction technologies like optical character reputation (OCR). To optimise our pipeline's performance, we used M3DOCRAG. M3DOCRAG is an modern multi-modal RAG framework that adeptly handles diverse report contexts (both closed-area and open-area), query hops (single-hop and multi-hop), and proof modalities (textual content, charts, figures, etc.). M3DOCRAG identifies relevant documents and responds to enquiries via a multi-modal retriever and a masked language model, allowing it to efficaciously manage single or many files even as retaining visual records. This device facilitates the availability of a QA assistant and an MCQ generator, enabling college students to assess their information. Students can be able to do examinations (both more than one-choice and descriptive) the use of our method. Descriptive responses are assessed the use of a textual content similarity degree that quantifies the resemblance among human-written and AI-generated replies, ultimately assigning rankings based in this evaluation.**

I. INTRODUCTION

To address the constraints of modern DocVQA methodologies, we've hired M3DOCRAG (Multi-modal Multi-web page Multi-Document Retrieval-Augmented Generation), an revolutionary multi-modal RAG framework that adeptly contains various file contexts (each closed-area and open-domain), query complexities (single-hop and multi-hop), and proof sorts (textual content, charts, figures, etc.). The M3DOCRAG framework retrieves pertinent file pages via a multi-modal retrieval model, which include ColPali, and formulates solutions to enquiries primarily based on the received pages through a multi-modal language model (MLM), including Qwen2-VL. Coming to Question generation, We used a model from the same Qwen series which is Qwen-32B. Once the data is extracted from the documents, it is further chunked and stored in a database. When a student hits QUIZ option, stored chunks are fetched and are sent to Qwen-32B model in a modified instruction format. As, the model has already been trained on instruct based datasets, our modified instriuction is understood by the model without any hallucinations. Our project not only aims to provide MCQs but also descriptive questions. Descriptive question are answered by students physically and the answer sheets are digitized. Once the text is extracted from the answer sheets, Qwen-32B model also generated the answers for the questions with prescribed context that is deduced from the question. These two answers are compared and are evaluated with the help of a hybrid similarity metric.

II. RELEVANT DOCUMENT RETRIEVAL VIA M3 DOCRAG

*a) **Creation of image embeddings**:* We have used ColPali model for this purpose. Documents are first converted into images and are sent to the model. The images of documents and thus converted into embeddings and we store them in a database for quicker fetching. We have stored the embeddings in SupaBase through PostGreSQL. Whenever the model is queried, the data is fetched from here so that we don't have to generate the embeddings every time the model is queried.

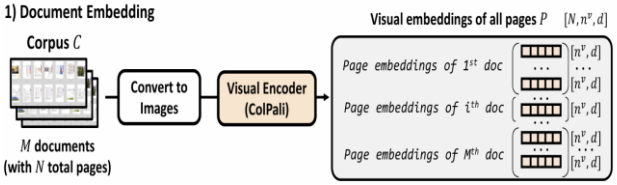


Fig 1.0: Embeddings from document images through ColPali.

Lets say there are M number of documents **(a**s shown in the above figure(Fig 1.0)), these documents are converted into images with the help of a python module called 'pdf2image'[10]. The encoder of ColPali model vectorizes the images. These images are sent to ColPali model and thus the image embeddings are generated. The reason we use this process is, the textual features present in the image are replicated in the embeddings generated by the model, this makes us easy to compare text embeddings and image embeddings are both are of same datatype.

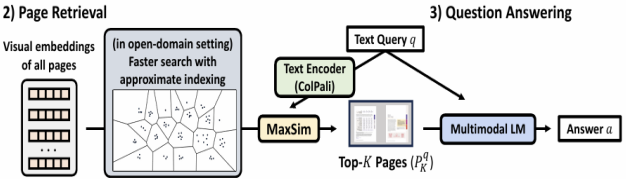


Fig 1.1: Page Retrieval from visual embeddings

The text inquiry and each document insertion includes a metric of similarity, such as the similarity of cosmos. The most important pages of TOP_K are selected for text extraction and masking images.

*b) **Image-Masking:*** Before giving the fetched documents to a text extractor, we have to separate the text from images, tables and other irrelevant data present in the document. We use a YOLO

model for detecting the text, images and tables present in a document. Detected text part is sampled from the document and is given to pytesseract-OCR.

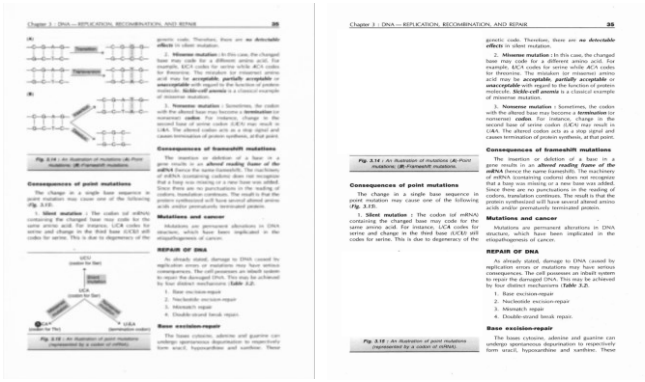


Fig 1.2: This figure shows how documents are masked before text extraction

As shown in the above figure(Fig 1.2), the images that are detected by a detector are masked before sending the document for text extraction, so that we can exclude the text from the images that could pollute the text extracted from the textual part solely.

We have used YOLO-v5[16] for detection of images from documents. The model has been trained on annotated images collected from NCERT text books of various domains. The example we have shown above (Fig 1.2) is from NCERT 12th standard Biology textbook. There are other better option for efficient image masking but as of now we have chosen this model for the specified purpose.

c) **Question-Answering:** We provide a QA pipeline in our project so that students can actively clarify their doubts. Once the OCR model generates text from masked documents, the data is then chunked through double merging chunking. A sentence transformer model calculates the embeddings for the question and the chunks. The resemblance score is calculated between the query and segments and the most important segments of Top_k are obtained. The QA extraction model then finds an answer in the loaded segments and provides an identified response. This not only answers the doubts but also generated answers for the question that are given in descriptive exams in our project. These answers are used to calculate the score of the students by cross validating the AI generated answers and answers written by students.

d) ***Question-Generation for Quizzes****:* Once the text fetched from documents is chunked, a Question-Generation model takes one chunk at a time and generated a maximum of four questions for each. The answers can be found through our QA model. This created a QA map and is used for conduction a simple exams like quiz. We have trained our QA model such that it generates open ended questions as well. So, we can conduct regular exams as well.[5]

We use the same Qwen-32B model for question generation as well. The model is pretrained on various instruct datasets so that it could understand most of the common instructions to language models. A hybrid instruction is made by concatenating 'generate question' query and the text chunk. This instruction is given to the model and the model generates question in the form of MCQs with options and answers to them. Later, we filter the output received and display only the questions and corresponding options for the students. The answers to the questions are stored again in our database. Once the test is completed our model evaluates the answers and displays the score accordingly.

III. EVALUATION METRICS AND TEXT SIMILARITY

There are many methods to find similarity between two texts. We have found a method through '**A novel hybrid methodology of measuring sentence similarity**' paper to efficiently calculate the similarity score between the answers generated by AI and student. But, the score calculated through this way is scaled to only 50%. The rest of the score is dependent on the physical evaluator(human).

Answers generated by AI may vary from time to time and taking these answers as they are for evaluation is absurd. So, we use an architecture called GCNN to magnify the theme of the answers so that the minor changes in the answers would be shadowed by these magnified features.

A) **GCNN:** G-CNN uses three simultaneous CNN with different core sizes to extract representative semantic information [15]. G-CNN amalgamates have maps derived from three CNN to a singular function and then using the best representative function map by means of maximum association. The G-CNN equation is as follows. [16]

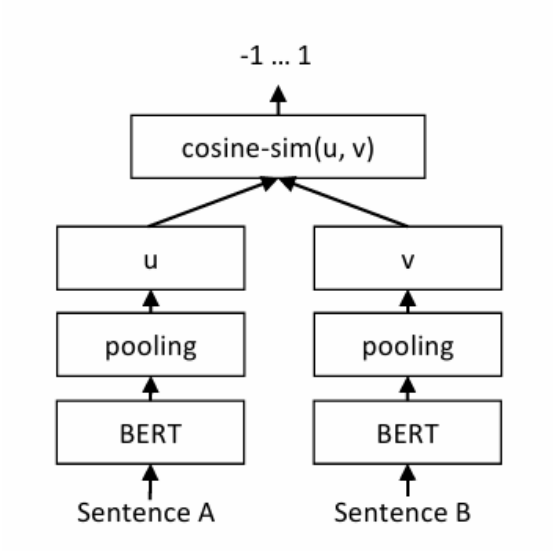
$$G\text{-}CNN_i = \max(CNN_i^k, CNN_i^{k+2}, CNN_i^{k+4})$$

Fig 2.0: Mathematical representation of GCNN layer

Initially the answers from AI and student differ in size which makes it difficult for comparison. So, a sentence transformer is used to vectorize the answers into same sized vectors. These vectors are then passed through GCNN layer and the final outputs are taken for similarity scoring.

B) **Sentence Transformer:** Sentence-BERT (SBERT), is an adaptation of Bert architecture, which uses the Siamese and triplet network to generate semantically significant insertion of the sentence. This allows Bert to be used for some new activities that were not used for previously. These tasks include extensive semantic comparisons, grouping and obtaining information through semantic search. [14]

Similarity measures can be implemented with remarkable efficiency on current technology, which allows SBERT to facilitate semantic search for similarity and groupings. The time required to identify the most popular pair of sentence in a set of 10,000 sentences will be reduced from 65 hours using Bert to approximately 5 seconds to generate 10,000 sentences using SBERT, followed by another 0.01 seconds to calculate the similarity of spaces. The use of optimized index structures can shorten the time to identify the most compatible topic Quora from 50 hours to just milliseconds.



"Fig 2.1: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function".

Bert [18] is a pre -trained transformer network that has set new state -of -the -art results for several NLP tasks, including answers to questions, sentence categorization and sentence regression. Input for Bert in pairs are two sentences defined by a specific token [SEP]. More heads are used in 12 layers in the basic model or 24 layers in a large model, while the output is processed by a simple regression function to make the final label.

SBERT includes the technique of association into Bert/Robert output to generate a fixed -size sentence. We are exploring three association strategies: we use the CLS-token output, calculation of the diameter of the output vectors (average technology) and determining the maximum over the time of the output vectors (maximum strategy). Standard configuration is average. [14]

IV. ANALYSIS

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Avg. GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.0	72.87	81.52
Avg. fast-text embeddings	77.96	79.23	91.68	87.81	82.15	83.6	74.49	82.42
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.8	69.45	84.94
BERT CLS-vector	78.68	84.85	94.21	88.23	84.13	91.4	71.13	84.66
InferSent - GloVe	81.57	86.54	92.50	**90.38**	84.18	88.2	75.77	85.59
Universal Sentence Encoder	80.09	85.19	93.98	86.70	86.38	**93.2**	70.14	85.10
SBERT-NLI-base	83.64	89.43	94.39	89.86	88.96	89.6	**76.00**	87.41
SBERT-NLI-large	**84.88**	**90.07**	**94.52**	90.33	**90.66**	87.4	75.94	**87.69**

"Table 1: Evaluation of SBERT sentence embeddings using the SentEval toolkit. SentEval evaluates sentence embeddings on different sentence classification tasks by training a logistic regression classifier using the sentence embeddings as features. Scores are based on a 10-fold cross-validation. This table is taken from [14]".

SentEval (Conneau and Kiela, 2018)[17] is a popular toolkit to evaluate the quality of sentence embed dings. Inserting a sentence serves as attributes for logistic regression classifier. The logistical regression classifier is trained on many tasks using a 10 -fold cross validation frame and the accuracy of prediction is calculated for the test fold. [14]

• MR: Sentiment prediction for movie reviews snippets on a five-start scale (Pang and Lee, 2005).

• CR: Sentiment prediction of customer product reviews (Hu and Liu, 2004).

• SUBJ: Subjectivity prediction of sentences from movie reviews and plot summaries (Pang and Lee, 2004).

• MPQA: Phrase level opinion polarity classification from newswire (Wiebe et al., 2005).

• SST: Stanford Sentiment Treebank with bi nary labels (Socher et al., 2013).

• TREC: Fine grained question-type classification from TREC (Li and Roth, 2002).

• MRPC: Microsoft Research Paraphrase Cor pus from parallel news sources (Dolan et al., 2004).

The findings are given in Table 1. SBERT will achieve excellent performance in 5 out of 7 challenges. The average performance improves by approximately 2 percentage points in relation to opponents and the universal sentence. Although transmission learning is not the primary goal of SBERT, it overcomes the other wind techniques into this work.

Method	# Pages	Evidence Modalities					Evidence Locations			Overall	
		TXT	LAY	CHA	TAB	IMG	SIN	MUL	UNA	ACC	F1
Text Pipeline											
LMs											
ChatGLM-128k	up to 120	23.4	12.7	9.7	10.2	12.2	18.8	11.5	18.1	16.3	14.9
Mistral-Instruct-v0.2	up to 120	19.9	13.4	10.2	10.1	11.0	16.9	11.3	24.1	16.4	13.8
Text RAG											
ColBERT v2 + Llama 3.1	1	20.1	14.8	12.7	17.4	7.4	21.8	7.8	**41.3**	21.0	16.1
ColBERT v2 + Llama 3.1	4	23.7	17.7	14.9	**24.0**	11.9	25.7	12.2	38.1	**23.5**	19.7
Multi-modal Pipeline											
Multi-modal LMs											
DeepSeek-VL-Chat	up to 120	7.2	6.5	1.6	5.2	7.6	5.2	7.0	**12.8**	7.4	5.4
Idefics2	up to 120	9.0	10.6	4.8	4.1	8.7	7.7	7.2	5.0	7.0	6.8
MiniCPM-Llama3-V2.5	up to 120	11.9	10.8	5.1	5.9	12.2	9.5	9.5	4.5	8.5	8.6
InternLM-XC2-4KHD	up to 120	9.9	14.3	7.7	6.3	13.0	12.6	7.6	9.6	10.3	9.8
mPLUG-DocOwl 1.5	up to 120	8.2	8.4	2.0	3.4	9.9	7.4	6.4	6.2	6.9	6.3
Qwen-VL-Chat	up to 120	5.5	9.0	5.4	2.2	6.9	5.2	7.1	6.2	6.1	5.4
Monkey-Chat	up to 120	6.8	7.2	3.6	6.7	9.4	6.6	6.2	6.2	6.2	5.6
M3DocRAG											
ColPali + Idefics2	1	10.9	11.1	6.0	7.7	15.7	15.4	7.2	8.1	11.2	11.0
ColPali + Qwen2-VL 7B	1	25.7	21.0	18.5	16.4	19.7	30.4	10.6	5.8	18.8	20.1
ColPali + Qwen2-VL 7B	4	**30.0**	**23.5**	**18.9**	20.1	**20.8**	**32.4**	**14.8**	5.8	21.0	**22.6**

"Table 2: Closed-domain DocVQA evaluation results in MMLongBench-Doc. This is a report of generalized accuracy (ACC) across five evidence source modalities: text (TXT), layout (LAY), chart (CHA), table (TAB) and image (IMG), and three evidence locations: single-page (SIN), cross page (MUL) and unanswerable (UNA). The whole report is taken from [7]".

V. CONCLUSION

In conclusion, the project showcased the potential of integrating OCR and RAG to build an intelligent educational tool capable of automatic question generation. The system demonstrated reliable performance in extracting printed text and generating relevant questions, making it useful for educational purposes. The Next.js-based user interface provided an intuitive experience for document upload and question interaction, while the backend efficiently processed requests and managed data.

Despite the success, challenges such as OCR inaccuracies for handwritten text, limited question diversity, and scalability issues were identified. Relying on the accurate extraction of the text indicated that any inaccuracies in OCR endangered the quality of issued questions. The inability of the system to provide higher or more complicated questions emphasized the need for further training and improvement of the RAG model.

Overall, the project confirms that such a system can significantly aid in educational settings by automating question generation. However, enhancing the accuracy, scalability, and question variety would make the tool more robust and versatile for broader applications.

This project developed an Intelligent Educational System for Automatic Question Generation and Retrieval using Retrieval-Augmented Generation (RAG) and Optical Character Recognition (OCR). The primary objective was to create a tool that could scan educational materials, extract text content, and automatically generate contextually relevant questions to assist students in self-assessment and learning.

VI. REFERENCES

[1] Retrieval-Augmented Generation for Large Language Models: A Survey Yunfan Gaoa, Yun Xiongb, Xinyu Gaob, Kangxiang Jiab, Jinliu Panb, Yuxi Bic, Yi Daia, Jiawei Suna, Meng Wangc, and Haofen Wang.

[2] Scalable Educational Question Generation with Pre-trained Language Models Sahan Bulathwela, Hamze Muse and Emine Yilmaz Centre for Arti cial Intelligence, University College London, United Kingdom.

[3] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks atrick Lewis , Ethan Perez, Aleksandra Piktus , Fabio Petroni, Vladimir Karpukhin , Naman Goyal , Heinrich Küttler , Mike Lewis , Wen-tau Yih , Tim Rocktäschel , Sebastian Riedel , Douwe Kiela.

[4] emrQA: A Large Corpus for Question Answering on Electronic Medical Records Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng.

[5] Automatic question generation and answer assessment: a survey. Bidyut Das, Mukta Majumder, Santanu Phadikar, Arif Ahmed Sekh

[6] A Survey on Neural Question Generation: Methods, Applications, and Prospects. Shasha Guo, Lizi Liao, Cuiping Li, Tat-Seng Chua.

[7] M3DOCRAG:Multi-modal Retrieval is What You Need for Multi-page Multi-document Understanding. Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, Mohit Bansal, Chapel Hill.

[8] Lee, Ming Che, et al. "Sentence similarity computation based on POS and semantic nets." 2009 Fifth International Joint Conference on INC, IMS and IDC. IEEE, 2009.

[9] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Han naneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.

[10] Mathieu Fenniak and PyPDF2 Contributors. The PyPDF2 library, version 2, 2022.

[11] Yihao Ding, Siwen Luo, Hyunsuk Chung, and Soyeon Caren Han. Pdfvqa: A new dataset for real-world vqa on pdf doc

uments. In Joint European Conference on Machine Learn ing and Knowledge Discovery in Databases, pages 585–601. Springer, 2023.

[12] Gemini Team. Gemini 1.5: Unlocking multimodal under standing across millions of tokens of context, 2024.

[13] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-WeiChang. REALM:Retrieval-AugmentedLanguage Model Pre-Training. In ICML, 2020.

[14] A novel hybrid methodology of measuring sentence similarity Yongmin Yoo°, Tak-Sung Heo°, Yeongjoon Park° and Kyungsun Kim*

[15] Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks Nils Reimers and Iryna Gurevych Ubiquitous Knowledge Processing Lab (UKP-TUDA) Department of Computer Science, Technische Universit¨at Darmstadt www.ukp.tu-darmstadt.de.

[16] A YOLOv5-Based Model for Real-Time Mask Detection in Challenging Environments Saya Sapakovaa,*, Askar Sapakovb, Yelidana Yilibulec

[17] Alexis Conneau and DouweKiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Represen tations. arXiv preprint arXiv:1803.05449.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Un derstanding.