

The University of Hong Kong

School of Computing and Data Science

COMP7705

Project Report

Multi-Modal RAG Based Learning and Research Assistance System

Submitted in partial fulfillment of the requirements for the admission to

the degree of Master of Science in Computer Science

By

|  |  |
| --- | --- |
| Yu Yongheng | 3036198516 |
| Qiu Jiehao | 3036380365 |
| Zhang Yuhao | 3036413019 |
| Liu Chen | 3036425646 |

Supervisor: Dr. T.W. Chim

School of Computing and Data Science, HKU

Date of submission: [dd/07/2025]

# **Abstract**

The rapid growth and heterogeneity of academic resources make it increasingly challenging for students and researchers to locate relevant information efficiently. We propose a web-based Learning and Research Assistance System powered by a multi-modal Retrieval-Augmented Generation (RAG) framework. Our approach combines the ColQwen2 visual-language model (based on Qwen2-VL-2B) with a ColBERT-style multi-vector retrieval mechanism to process diverse document formats including PDF, PPTX, DOCX, and images. The system extracts, embeds, and semantically indexes document content to deliver precise, context-aware answers to user queries. Beyond semantic search, it supports automated citation formatting (APA, MLA, Chicago), cross-document comparison, Neo4j-based knowledge graph visualization, and an exam preparation mode with fuzzy matching and highlight-based review. This unified platform addresses fragmented academic workflows and enhances productivity by integrating AI-powered retrieval with user-centric tools. (Result: In evaluation on real student workloads, ...)

# **Acknowledgement**

Thank

**Contents**

[1 Introduction 1](#_Toc410443884)

[1.1 Background & Motivation 1](#_Toc612884224)

[1.2 Problem Statement & Objectives 1](#_Toc1413581756)

[1.3 Report Organization 2](#_Toc456986331)

[2 Related Work 2](#_Toc1167743445)

[2.1 Academic Reference Managers and Tools 2](#_Toc411030182)

[2.2 Retrieval-Augmented Generation (RAG) Systems 3](#_Toc1876860122)

[2.3 Multi-Modal Learning and Retrieval Systems 3](#_Toc779671)

[3 Requirement Analysis 4](#_Toc219028615)

[3.1 Design-oriented user assumption 4](#_Toc426961347)

[3.2 Competitive SWOT Analysis 5](#_Toc1196494402)

[3.3 User Journey Mapping 6](#_Toc444543906)

[3.4 MoSCoW Prioritization 7](#_Toc353820229)

[4 Methodology 7](#_Toc274370260)

[4.1 System Architecture 7](#_Toc693569711)

[4.2 Front-End Design 8](#_Toc284896861)

[4.3 Back-End Implementation 9](#_Toc1520493664)

[5 Implementation 9](#_Toc2029095195)

[5.1 Document Ingestion and Preprocessing 9](#_Toc962628005)

[5.2 Multi-Modal Embedding and Retrieval 9](#_Toc1894567184)

[5.3 Testing 11](#_Toc1250627419)

[6 Evaluation 11](#_Toc1872577944)

[6.1 Experimental Setup 11](#_Toc1044658023)

[6.2 Quantitative Results 12](#_Toc1888578336)

[7 Conclusion & Future Work 13](#_Toc1527790492)

[Appendix 13](#_Toc112831865)

[References 13](#_Toc137094754)

# **Introduction**

## **Background & Motivation**

In modern academic and research environments, students and researchers are increasingly overwhelmed by the volume and diversity of digital resources, ranging from lecture slides and scholarly articles to scanned notes and multimedia content. Navigating this fragmented ecosystem and locating relevant information across heterogeneous formats is both time-consuming and error-prone.

Traditional reference managers such as Zotero and Mendeley assist with document organization and citation formatting, but they lack deeper semantic understanding and intelligent features such as contextual passage highlighting, semantic search, or knowledge graph visualization. These limitations hinder efficient learning, research synthesis, and exam preparation.

Recent advances in large language models (LLMs) and vector-based retrieval techniques like Retrieval-Augmented Generation (RAG), offer promising solutions by enabling semantic understanding and real-time document interaction. In particular, multi-modal RAG frameworks that incorporate visual-language models and multi-vector search mechanisms allow systems to ingest and interpret various document types—including slides, scanned images, and text—more intelligently and interactively.

This project aims to build such a system to streamline academic workflows by combining modern AI capabilities with practical document analysis and retrieval features.

## **Problem Statement & Objectives**

Despite the availability of various academic tools, there is currently no unified platform that supports semantic, cross-format information retrieval along with essential academic features such as citation generation, document comparison, and exam-focused review.

To address this gap, we propose the development of an intelligent, web-based Learning and Research Assistance System. The system's core objectives are as follows:

* **Document Ingestion & Parsing**: Ingest PDF, PPTX, and image files via a web interface and extract raw text using PyMuPDF, python‐pptx, and Tesseract OCR.
* **Multi‐Modal Embedding & Retrieval**: Use the ColQwen2 visual‐language model to generate separate text and image embeddings, fuse these representations, and perform top-N semantic retrieval via cosine similarity (and optional Euclidean distance) in a FAISS index.
* **Contextual Highlighting & Comparison**: Automatically highlight matched keywords or sentences in retrieved passages and enable side-by-side semantic comparison of different documents or document sections.
* **Citation Management**: Parse and format bibliographic references in APA, MLA, and Chicago styles, streamlining the process of preparing papers and reports.

This set of features directly addresses the common pain points in academic reading and research, offering a more efficient and AI-enhanced alternative to conventional tools.

## **Report Organization**

The remainder of this report is structured as follows:

* Section 2 reviews related work in academic tools, RAG systems, and multimodal models.
* Section 3 defines system requirements based on user assumptions and competitive analysis.
* Section 4 presents the methodology, including architecture, database design, and module interactions.
* Section 5 details implementation procedures, including embedding logic and result visualization.
* Section 6 evaluates retrieval performance using case studies and metrics.
* Section 7 concludes with future directions.

# **Related Work**

## **Academic Reference Managers and Tools**

A variety of academic software tools have been developed to assist with literature management, citation, and basic search. Zotero and Mendeley are among the most widely adopted reference managers, known for organizing citations and managing PDF libraries. However, they provide only basic keyword search, lacking semantic understanding, cross-format document handling, or real-time contextual retrieval.

Meanwhile, document utility platforms such as iLovePDF focus on file conversion, compression, and OCR functionalities. While useful for structural manipulation, these tools are not designed for deep content comprehension, semantic comparison, or learning support. None of the above integrate advanced AI-driven features such as multi-modal semantic search, citation automation, or knowledge graph exploration.

## **Retrieval-Augmented Generation (RAG) Systems**

RAG is a hybrid architecture that combines a retrieval component (often a dense vector index such as FAISS) with a generative language model to return contextually relevant responses. Pioneering models such as ColBERT (Emami et al., 2021) introduced late interaction mechanisms using token-level dual encoders, enabling high-precision multi-vector retrieval while preserving scalability.

More recent extensions have adapted RAG to multi-modal domains. Qwen2-VL (Zhu et al., 2024) is a visual-language model capable of jointly encoding text and image inputs. Its architecture serves as the foundation for our project’s ColQwen2 model, which extends Qwen2-VL with ColBERT-style multi-vector embedding to enable fine-grained retrieval over both text passages and visual slide/image content.

These advancements provide the technical foundation for building a RAG-based academic search system that supports precise passage-level retrieval across diverse media types.

## **Multi-Modal Learning and Retrieval Systems**

Research in multi-modal learning has explored models that jointly reason over images, text, and other inputs. Prior works include VilBERT, ViLT, and CLIP, which were originally developed for tasks like image captioning, cross-modal classification, and retrieval. These models demonstrated the potential of aligning different modalities in a shared embedding space.

However, their application in academic search remains limited. Existing academic platforms typically process only text or scanned PDFs without integrating visual learning elements, such as images from lecture slides or hand-annotated notes. Furthermore, cross-modal semantic alignment is rarely used in real-world learning tools.

Our system builds upon these foundations by embedding textual and visual content independently and fusing the representations to support unified semantic retrieval across document types.

# **Requirement Analysis**

## **Design-oriented user assumption**

Although we did not conduct formal user surveys or interviews, the system design is grounded in widely observed user behavior patterns among university students and researchers. We drew on academic literature, informal feedback from our own experience to hypothesize the needs and pain points of the target user group.

To anchor our design decisions, we developed two representative user archetypes that reflect the needs of our target audience:

Persona 1: David – Undergraduate CS Student

* Profile: 3rd-year Computer Science student
* Goal: Review lecture slides and textbooks for final exams
* Pain Points:
  + PDF files are long and hard to navigate
  + Forgetting where specific concepts (e.g., Bellman equation) were mentioned
  + Time-consuming to extract citations for project reports
* Needs:
  + Fast content search across course materials
  + Highlighted relevant answers
  + Citation suggestions in APA format

Persona 2: John– Postgraduate CS Student

* Profile: Graduate researcher working on NLP
* Goal: Compare concepts from multiple academic papers
* Pain Points:
  + PDF-to-PDF comparison is manual and slow
  + OCR errors in scanned datasets
  + No knowledge visualization across related topics
* Needs:
  + Visual graph of related terms
  + Semantic similarity-based document comparison
  + Image-supported search for figures and tables

These user profiles reflect three broader user categories:

* Undergraduate and postgraduate students working on assignments, thesis projects, or exam preparation.
* Early-career researchers and academic writers seeking to manage literature efficiently and generate accurate citations.
* Anyone interacting with large volumes of academic content, including scanned lecture notes, slides, and journal articles.

By designing for these personas, we aim to address challenges such as:

* Difficulty navigating lengthy, unstructured PDFs and mixed-format documents
* Time-consuming manual citation generation
* Fragmented workflows using multiple tools for reading, referencing, and review
* Lack of personalized learning features such as visual knowledge graphs or fuzzy search

This user-centered approach guided the prioritization of system features such as multi-format ingestion, semantic search with contextual highlighting, automated citation generation, and interactive knowledge graph construction.

## **Competitive SWOT Analysis**

To identify opportunities for innovation and differentiation, we conducted a comparative SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis on several widely used academic and document management tools. The goal is to identify gaps and improvement areas that our system can strategically address.

We selected three representative tools for analysis: Zotero, Mendeley, iLovePDF. These tools cover citation management, collaboration, and file manipulation respectively. However, despite their popularity, they exhibit several key limitations that highlight a clear opportunity for systems integrating AI-powered semantic search, multi-modal document understanding, and personalized learning support—all of which are core pillars of our system.

The table below summarizes the key comparative insights from the SWOT analysis.

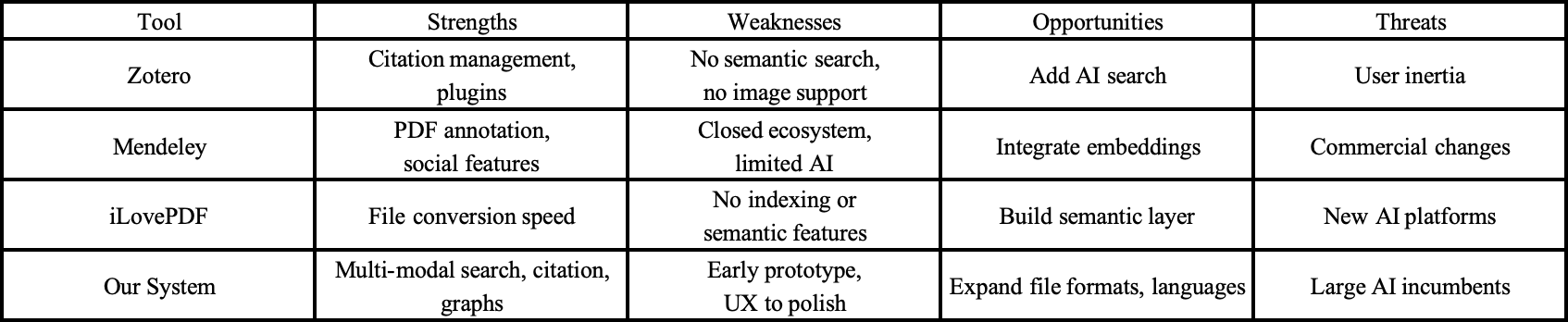


Table 1

The analysis reveals a key market gap: current tools are either utility-based or narrowly academic-focused, but none provide a unified, intelligent research and learning system. By leveraging RAG models, multi-modal input handling, and advanced backend design, our system aims to fill this unmet need, providing a unique and impactful academic assistant.

## **User Journey Mapping**

To validate the functional flow and ensure that the design supports real-world academic scenarios, we developed a hypothetical user journey map:

* A student uploads multiple resources—such as scanned lecture notes and a PDF of a research paper—on reinforcement learning.
* The student enters a natural-language query: “Explain the Bellman equation.”
* The system ingests the uploaded files, extracts and embeds content, then ranks passages based on semantic similarity using multi-vector embedding.
* The top passages are returned, with matched sentences highlighted for quick comprehension.
* The user is provided with:
  + Formatted citations (APA/MLA) for the source documents
  + An interactive knowledge graph centered on the Bellman equation
  + The option to bookmark relevant passages for later review

This journey highlights the system’s value proposition: rapid information retrieval across multiple formats, enhanced with citation and conceptual navigation features.

## **MoSCoW Prioritization**

To guide system development and feature delivery within a limited time frame, we adopted the MoSCoW prioritization framework, which categorizes requirements into four tiers:

* Must-have: Core functionalities without which the system would fail to deliver value
* Should-have: Important features that greatly improve usability but are not critical at launch
* Could-have: Useful enhancements that can be deferred if time or resources are limited
* Won’t-have: Out-of-scope ideas reserved for future development

Figure x demonstrate the prioritization:

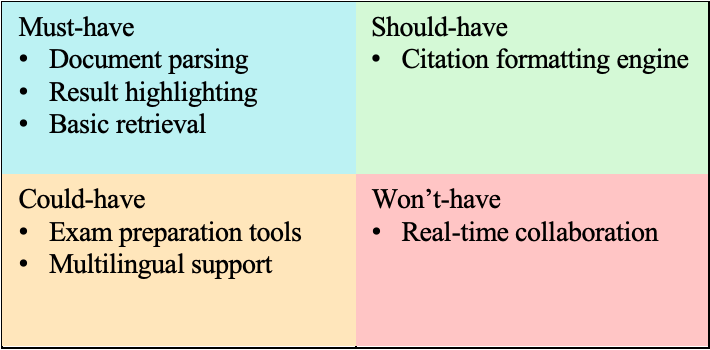


Figure 1

# **Methodology**

## **System Architecture**

Our system employs a microservices architecture to ensure modularity, scalability, and ease of maintainability. Each component is independently deployable and responsible for a single core function. This design enables parallel development, fault isolation, and flexible scaling. Figure1 illustrates the high-level architecture:

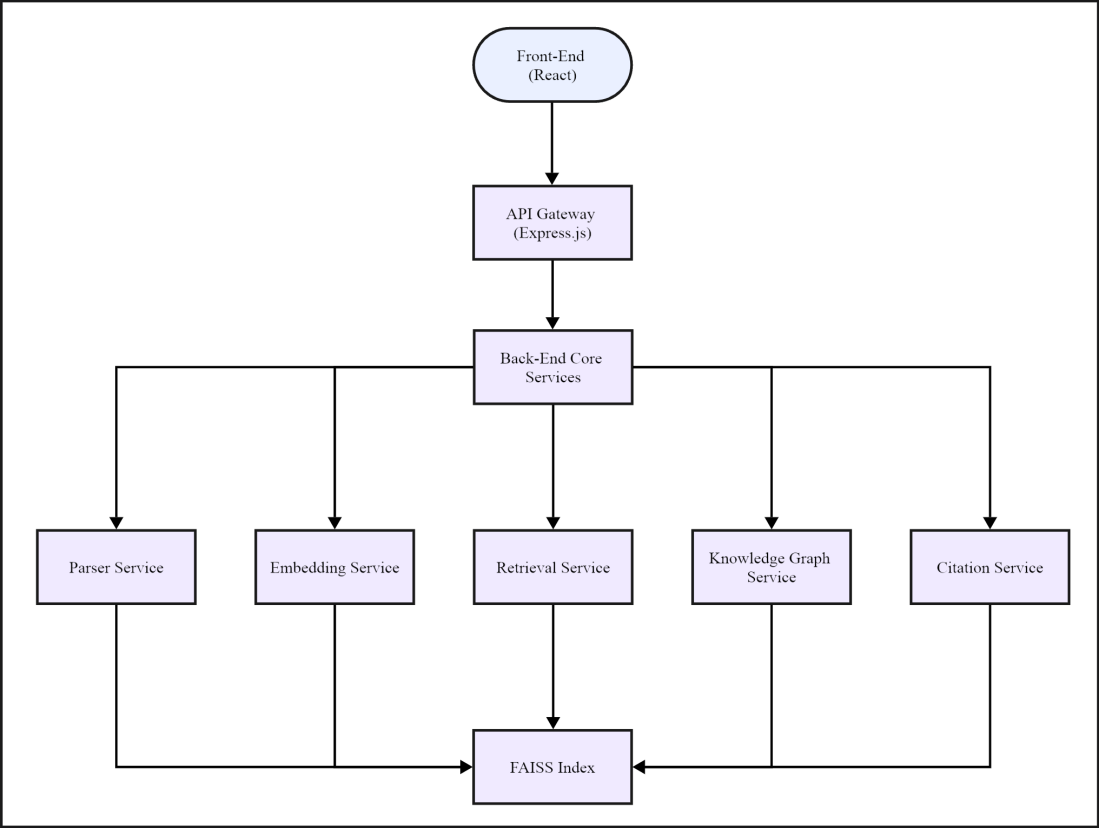


Figure 1

**Front-End (React)**: Provides the user interface for uploading files, entering queries, and viewing retrieval results with highlights.

**API Gateway (Express.js)**: Routes requests to specialized services, handles authentication, rate limiting, and logging.

**Parser Service**: Extracts plain text from PDF, PPTX (via PyMuPDF, python-pptx) and images (via Tesseract OCR).

**Embedding Service**: Cleans and tokenizes extracted text, then uses the ColQwen2 model to generate multi-vector embeddings, which are stored in a FAISS index.

**Retrieval Service**: Embeds user queries and computes cosine similarity against stored vectors to retrieve the Top-N relevant passages.

**Knowledge Graph Service**: Extracts named entities and relations and stores concept connections in Neo4j.

**Citation Service:** Identifies bibliographic references via regex and natural-language patterns, formats them in APA/MLA/Chicago.

## **Front-End Design**

To be continue

## **Back-End Implementation**

To be continue

# **Implementation**

This section presents the complete technical implementation of our multi-modal intelligent research assistance system, built around a RAG pipeline that integrates text and visual data. The system comprises a full-stack web application, microservices, and embedding models based on ColQwen2 and ColPali.

## **Document Ingestion and Preprocessing**

The system supports user uploads of academic materials in PDF, PPTX, and image (JPG/PNG) formats via a web-based interface. Once uploaded, files are parsed and transformed into text and/or image data for downstream processing.

**File Parsing**

* PDFs: Processed using PyMuPDF to extract page-level text.
* PPTX slides: Parsed using python-pptx, extracting content from slide shapes.
* Images: Processed with Tesseract OCR to recognize scanned content.

**Text Cleaning and Chunking**

Extracted text is cleaned by:

* Lowercasing
* Removing stopwords, punctuation, and numerals
* Segmenting into ~200-word "chunks" for efficient retrieval

Each chunk is stored with metadata including its source document, page number, and chunk index. Images are rendered into standardized PNG format for ColPali compatibility and layout preservation.

## **Multi-Modal Embedding and Retrieval**

To support semantic understanding across both textual and visual inputs, our system uses a hybrid retrieval architecture powered by ColQwen2 and ColPali.

*ColPali* is an efficient document retrieval with Vision Language Models, developed by [Manuel Faysse](https://arxiv.org/search/cs?searchtype=author&query=Faysse,+M) et al. (2024) and published on GitHub[[1]](#footnote-0) for all researchers to learn, and furtherly make contributions to it.

Compare with standard retrieval method, *ColPali* eliminates the need for potentially complex and fragile layout recognition and OCR pipelines, allowing only one model to simultaneously consider the text and visual content of documents, like layout, charts, etc.

In our program, the documents needed by the *ColPali* are posted through the website interface. Then, each page of the documents are transferred into a unified PNG image format and stored in the database, avoiding image loss and other potential errors.

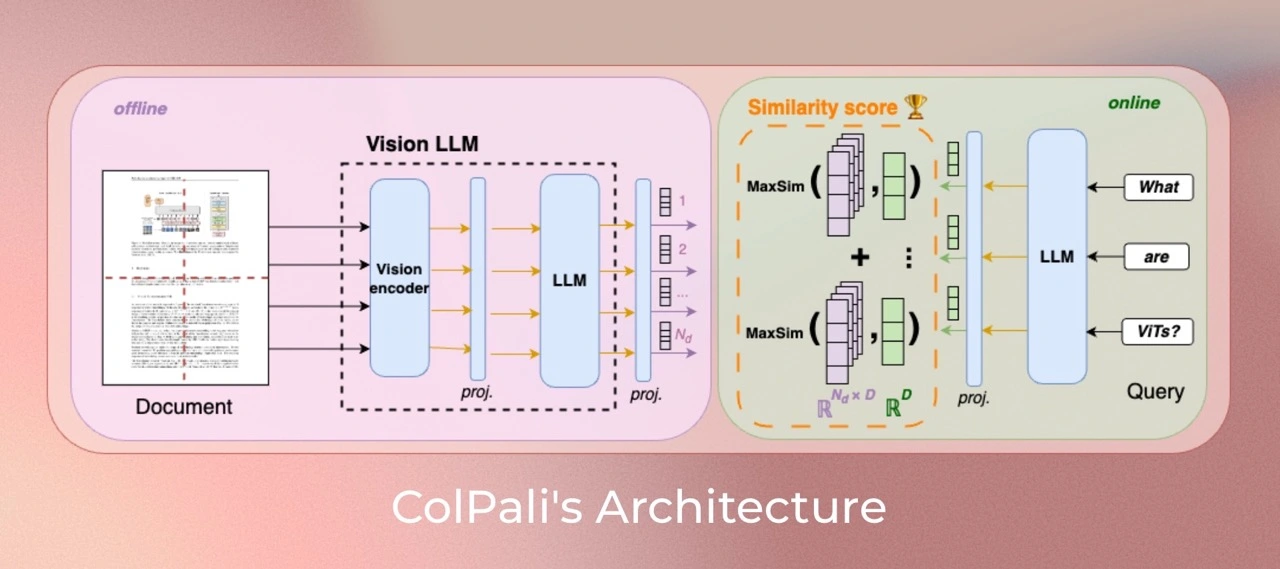


Figure 3

Once users input a query into the *ColPali*, both the queries and the PNG images of documents will be decoded by the model into vectors for further similarity comparison, witch is based on cosine similarity. Formula is showed below:

Where AB is the dot product of vectors A and B, ||A|| ||B|| are the magnitudes (or norms) of vectors A and B, respectively.

Finally, the model will output a list of scores, witch indicate the similarity between the query and the documents’ content. The images corresponding to high scores are more likely to be the original documents that the user is looking for, as their contents have a higher similarity to the query.

**Result Delivery & Visualization**

Since we got a list of similarity scores for each pages of documents compared with query, we need to decide witch pages can be the output result sent back to the website interface. Here, we use Top-k algorithm. Specifically, we use a min-heap (or max-heap, depending on the criterion) to store the top k elements as the dataset is processed. The heap allows efficient insertion and deletion, reducing the time complexity to O(n log k), which is especially useful for large datasets. In this system, we will set k into 3.

## **Testing**

To be continue

# **Evaluation**

This section presents the evaluation of our system’s retrieval component using the ColPali model in a simulated academic use case. The experiment aims to assess the model’s capability to distinguish semantically relevant visual content from irrelevant material and inform design choices for the Top-K retrieval strategy.

## **Experimental Setup**

To validate the semantic matching capability of the ColPali engine, we conducted a controlled test simulating a real-world use case in a course review scenario.

**Environment**

* Model: ColPali (installed via pip install colpali-engine)
* Runtime: Python 3.11.6, PyTorch 2.4, Ubuntu 22.04
* IDE: VSCode with Conda virtual environment (env1)
* Hardware: NVIDIA RTX 3060, 12GB VRAM

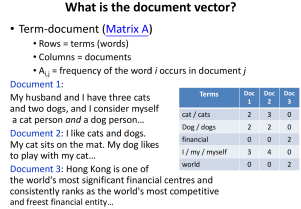
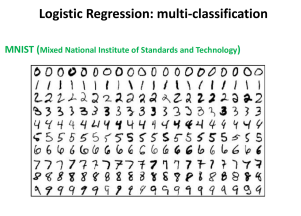
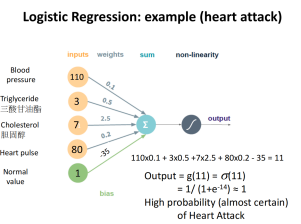
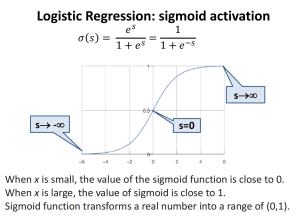
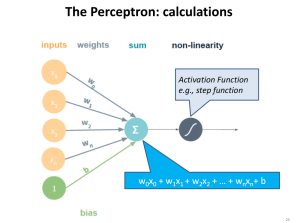
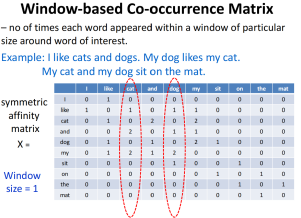
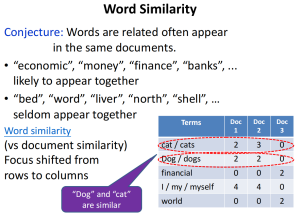
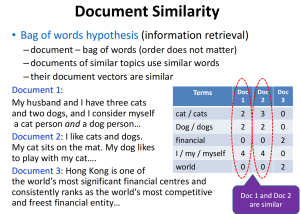


Figure 4 (From top left to bottom right are Img1 to Img8)

To test the model performance, we selected 8 lecture slides from actual coursework materials. These slides were converted to PNG images and placed into the input directory expected by the ColPali engine.

Then, we input 3 queries: “What is Bag of words hypothesis?”, “What is Term-document?”, and “I really hate Monday”. The key information words in the first two queries, such as Bag of words hypothesis and Term-document, appear in the courseware images, while the third question is unrelated to all courseware content.

After preparing all the necessary data for the experiment, run the file to obtain the similarity score as the result.

## **Quantitative Results**

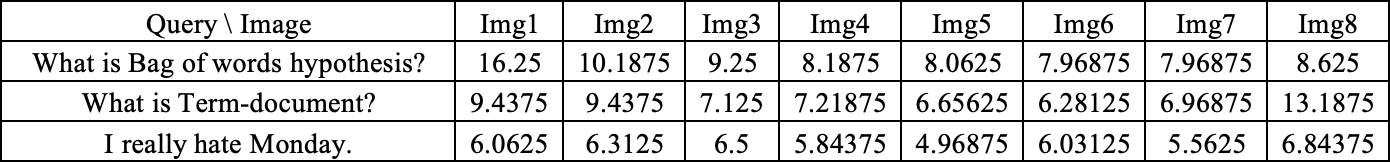


Table 3

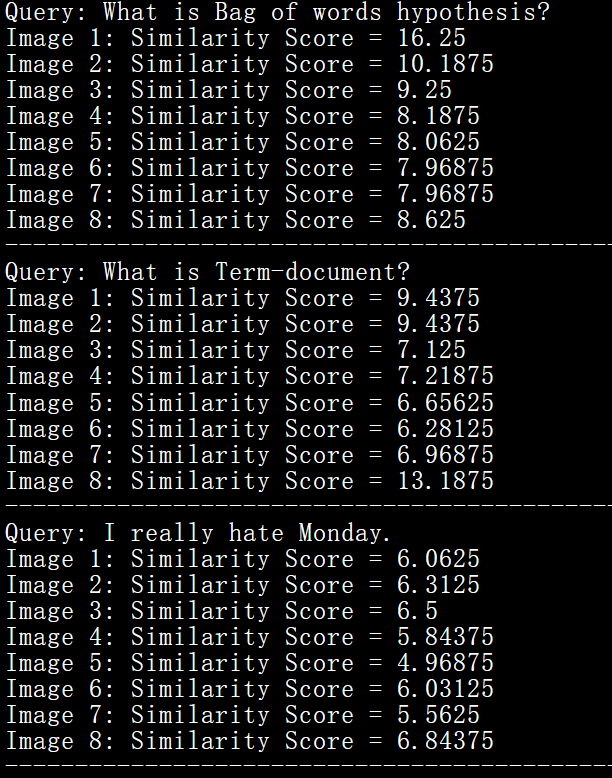


Figure 5 (Screenshot of Linux Terminal Experiment Results Display)

We can observe clearly in table 3 that for images containing same key information showed in queries, such as Img1 and Img8, similarity scores will be significantly higher than the scores of other images. Img1 get 6.25 points higher than the Img2, ranked second in terms of score. Also, Img8 is 3.75 points higher than the second one. For unrelated query, the score will be in a low score range, which in experiment is from 4.96875 to 6.84375.

The testing results will make an important role in functional design of Top-k algorithm in subsequent application.

# **Conclusion & Future Work**

To be continue

# **Appendix**

To be continue

# **References**

Faysse, M., Sibille, H., Wu, T., Omrani, B., Viaud, G., Hudelot, C., & Colombo, P. (2024, June). Colpali: Efficient document retrieval with vision language models. In The Thirteenth International Conference on Learning Representations. <https://arxiv.org/abs/2407.01449>

1. <https://github.com/illuin-tech/colpali.git> [↑](#footnote-ref-0)