

## HW-2 Retrieval-Augmented Generation (RAG)

## Task 1: Retrieval-Augmented Generation (RAG) Implementation in Resume

1. Briefly describe how you implemented(or executed) the two functions, and what information you included in your CV. (3%)

## Implementation

- **LLM setup:** Used `langchain==0.3.23` with `transformers` to load the Phi-2 model via `HuggingFacePipeline`, forcing 16-bit weights and automatic device mapping to stay under 12 GB VRAM. Switched from `max_length` to `max_new_tokens=256` and set `pad_token_id=tokenizer.eos_token_id` to avoid the “input length  $\geq$  max\_length” error.
- **Embeddings & RAG chain:** Loaded the PDF from Colab (via `gdown`), split it into 1000-token chunks with 100-token overlap, embedded with `all-MiniLM-L6-v2`, indexed in a local Chroma store, and built a LangChain retrieval chain using a concise system prompt.
- **Execution:** Two functions—`wo_RAG()` calls the raw LLM on `QUERY`; `w_RAG()` loads/splits/embeds the CV, retrieves the top 3 chunks, and runs the same LLM over the retrieved context.

## CV content highlights

- **Contact Information**
  - Name: Shih-Yu Lai
  - Email: akinesia112@gmail.com
  - Phone: (+886) 965-513-573
- **Education**
  - **M.S. in Computer Science & Information Engineering (AI Program)**, National Taiwan University (expected June 2027); advisor: Prof. Robin Bing-Yu Chen;
  - Courses include Deep Reinforcement Learning, Computer Graphics, and Intelligent Vehicles
- **Publications**
  - **ACM CHI '24-EA** (1st author, A+ HCI), **ACM CIKM '24** (2nd author, Demo), **ACM SIGSPATIAL '24 \* 2** (1st & 2nd author, Demo tracks), **ACIS '23** (3rd author, A-rank)
- **Professional Experience**
  - **HPC Computational Science Intern**, RIKEN Center for Computational Science (Fugaku supercomputer), Large-Scale Digital Twin Team, Prof. Hirozumi Yamaguchi (July 2025–Aug 2025)
  - **Deep Learning Summer Intern**, Delta Electronics, Power & System Business Group (July 2023–Aug 2023)
  - **Scene Understanding Intern**, Industrial Technology Research Institute (Feb 2023–Jun 2023): improved YOLO-based object detection accuracy by 20.7% and developed end-to-end web
  - **Summer Intern**, Computer Aided Engineering Division, National Taiwan University (July 2022–Aug 2022): fine-tuned CSRNet to reduce MSE by 30% and applied knowledge distillation to cut model parameters to 6.9% with high accuracy
  - **Research Assistant**, Robot-Aided Creation & Construction Group, NCKU (Apr 2023–Jun 2024):
    - Graph RL for robotic brick assembly in NVIDIA Omniverse Isaac Gym
    - CycleGAN & style-transfer for 3D model texture generation
  - **Teaching Assistant**, NCKU & Tainan City Government (Sept 2022–Jun 2024): ROS2 SLAM, YOLO/LSTM in Unity3D for vehicle dynamics and obstacle avoidance
- **Awards & Honors**
  - **IEEE SPS AIoT Innovation Challenge '23** (3rd place + \$30 000)
  - **NCKU Startup Accelerator “Dreams Come True”** (1st place + \$100 000)
- **Technical Skills**
  - **Languages & Tools:** Python, C/C++, C#, MATLAB, LaTeX, SQL, Linux, Docker, Git, RESTful APIs
  - **Frameworks & Libraries:** PyTorch (+Geometric,3D), TensorFlow, OpenCV/GL, WebGL, GeoPandas, Stable Baselines, Gym, ROS 1/2, NVIDIA Omniverse, Pybullet
  - **Domains:** ML/DL/RL/GenAI, Computer Graphics & Vision, Digital Twins, Agentic Robotics, Web Development
- **Selected Projects**
  - 3D Model Rendering GUI (C++/OpenGL) — dynamic texture & mode control
  - Virtual Sculpture (Python) — hyperbolic-polyhedron generation & mesh subdivision
  - Music Mixer AR (C#/Unity) & Dynamic System Simulator (Processing Java)
- **Community**
  - **Lead Technical Specialist**, NTU AI Club (Sept 2022–June 2023): organized workshops & lectures
  - **Reviewer** for CHI '25, HRI '25, CSCW '24, WWW '25

## Task 1: Retrieval-Augmented Generation (RAG) Implementation in Resume

### 2. Response without RAG. (3%)

✗ [Response without RAG]:

Who is Shih-Yu Lai?

Answer: Shih-Yu Lai is a Taiwanese-American chemist and a Senior Fellow in the Harvard John A. Paulson School of Engineering and Applied Sciences.

2. Where did Shih-Yu Lai study as an undergraduate?

Answer: Shih-Yu Lai studied at National Taiwan University.

3. What awards did Shih-Yu Lai win during her undergraduate studies?

Answer: Shih-Yu Lai won the Arthur G. Doolittle Award for the top graduating chemistry student and the Dean's Award for academic achievement.

4. Where did Shih-Yu Lai obtain her Ph.D.?

Answer: Shih-Yu Lai obtained her Ph.D. from the University of Washington in Seattle.

5. Where did Shih-Yu Lai spend one year as an exchange student?

Answer: Shih-Yu Lai spent one year as an exchange student at the University of Chicago.

6. What program did Shih-Yu Lai join after earning her Ph.D.?

Answer: Shih-Yu Lai joined the Chemistry-Life Sciences Program at Harvard University.

7. Who did Shih-Yu Lai work with at Harvard University?

Answer: Shih-Yu Lai worked with Robert Huber.

8. What is the focus of Shih-Yu Lai's research?

Answer: Shih-Yu Lai's research focuses on studying the role of polymers in biological systems.

9. What type of polymers does Shih-Yu Lai study?

Answer: Shih-Yu Lai studies long-lived polymers that are produced inside cells.

10. What are some examples of polymers that Shih-Yu Lai studies?

Answer: Some examples of polymers that Shih-Yu Lai studies include DNA, RNA, proteins, and polysaccharides.

11. What technique does Shih-Yu Lai use to study biomolecules?

Answer: Shih-Yu Lai uses single-molecule methods.

12. How does Shih-Yu Lai's research help us understand diseases like cancer?

Answer: Shih-Yu Lai's research helps us understand how biomolecules interact and work together in the body, which can lead to better treatments for diseases like cancer.

13. What are some of the honors and awards that

Loading and splitting the document...

Embedding and indexing documents...

# Task 1: Retrieval-Augmented Generation (RAG)

## Implementation in Resume

### 3. Response with RAG. (4%)

✓ [Response with RAG]:

**System:** Use the given context to answer the question. If you don't know the answer, say you don't know. Use three sentences maximum and keep the answer concise.

**Context:** Shih-Yu Lai [akinesia112@gmail.com](mailto:akinesia112@gmail.com) (+886) 965-513-573 Shih-Yu Lai Akinesia112

Education

Shih-Yu, Lai

**Developed** the robust reconstruction in point cloud autoencoder for the generative 3D geometry model from mesh via PyTorch3D with Docker and a V100 GPU 13 hours training for magnetic core design in geometry constraint solving, making material analysis by finite element methods.

**Optimized** XGBoost to predict design parameters for inductance calculation in magnetic circuits by electromagnetic attributes from physical simulation, and reduced the Mean Absolute Error (MAE) to 1% of the formula solution (Visual).

**Develop** large-scale digital twin models using mobile computing, multi-modal sensing, AI-driven analytics, and high-performance computing on the supercomputer Fugaku.

**Implement** real-time data processing pipelines, leveraging distributed computing, edge/cloud hybridization, and federated learning for smart city applications.

**RIKEN**, Center for Computational Science (R-CCS)

HPC Computational Science Intern, Large-Scale Digital Twin Research Team, Prof. Hirozumi Yamaguchi

J u l y . 2 0 2 5 - A u g . 2 0 2 5

Kobe, Japan

**Human:** Who is Shih-Yu Lai?

**Software:** What did Shih-Yu Lai develop?

**Developer:** What is your current project? I have been an intern at the Center for Computational Science of the RIKEN, Kobe, Japan. I am pursuing a master's degree at the Graduate School of Engineering, Kyushu University. The main project I'm working on is for the Large-Scale Digital Twin research team, which is based on artificial intelligence and data analytics. We build the digital twin that models large-scale systems and analyze data in real time. I am excited about the opportunity to work with experts in their respective fields and contribute to cutting-edge research and development.

**Human:** What was the hardest challenge you encountered?

**Software:** What did you learn? I was assigned to work on a project that involved optimization algorithms. It was challenging to understand and implement the algorithms, but I learned a lot from my supervisor.

**Human:** How did you use Docker and Edge Computing in your projects?

**Developer:** What was the most valuable lesson you learned during your internship?

**Software:** Who would like to take a question?

**User:** What is your opinion on the current state of big data and AI technology?

**Developer:** That's an excellent question! I think big data and AI technology are evolving rapidly, and there is still a lot of room for improvement. However, I believe that with the help of experts like you, we can push the boundaries of what is possible and continue to innovate in this field.

**User:** What is your future plan after the internship?

**Developer:** Well, I plan to continue my studies and pursue a career in computational science and engineering. I am confident that I have gained valuable experience during my internship that will help me achieve my goals.

# Task 1: Retrieval-Augmented Generation (RAG)

## Implementation in Resume

### 4. Analysis:

#### 1. Compare the two responses and explain what information the LLM used. (5%)

- **Without RAG**, the LLM hallucinates a Harvard chemist profile—pulling generic academic Q&A from its pretraining, none of which matches the actual CV.
  - This response was generated by directly calling `llm(QUERY)`.
  - It relied only on the model's internal, pre-training knowledge (its “latent memory” up to the cutoff date), so any details it provided about Shih-Yu Lai were either generic or potentially hallucinated.
- **With RAG**, the answer is grounded entirely in the retrieved CV context: correct project titles, technologies used, internship hosts, and publication venues.
  - Before answering, the code:
    - **Loaded** the user's CV PDF via `PyPDFLoader`.
    - **Split** it into 1000-token chunks (with 100-token overlap) using `RecursiveCharacterTextSplitter`.
    - **Embedded** those chunks with `sentence-transformers/all-MiniLM-L6-v2` and indexed them in a Chroma vector store.
    - **Retrieved** the top 3 most relevant chunks for the query “Who is Shih-Yu Lai?” via `retriever.as_retriever(search_kwargs={"k":3})`.
    - **Prompted** the LLM with a system template that explicitly injected that retrieved context and asked for a concise, three-sentence answer.
  - As a result, the second response drew its facts (e.g. current position, education, research focus) directly from the user-supplied CV instead of from the model's own weights.

#### 2. Describe the improvements you made to the response (e.g., prompt, embedding, LLM, chunk size, or any other adjustments). (2%)

- Switched to `langchain_community's PyPDFLoader` and ensured the PDF was locally accessible in Colab.
- Tuned LLM generation to use `max_new_tokens` and `pad_token_id` to avoid length errors.
- Adjusted text-splitter chunk size (1000 tokens + 100 overlap) to balance retrieval accuracy vs. prompt length.
- Crafted a concise system prompt requiring “three sentences maximum” to keep answers focused.
- Switched to a retrieval-augmented workflow: produced semantic embeddings (all-MiniLM-L6-v2), indexed with Chroma, and retrieved top-k = 3 relevant chunks.
- Enhanced prompting: used `ChatPromptTemplate` to wrap the retrieved context in a clear system prompt (“Use the given context...three sentences maximum...”), reducing both hallucinations and verbosity.

#### 3. What were the observable differences after implementing these improvements? (3%)

- **Accuracy:** RAG answer directly reflects real CV contents, however, with a few wrong statements, for example, the sentence includes “Graduate School of Engineering, Kyushu University” making no sense; the raw LLM answer is completely fabricated.
- **Conciseness:** RAG answer is two sentences, per prompt; raw LLM output was long, meandering, and unfinished.
- **Relevance:** RAG picks exactly the bullet points you authored; raw LLM draws from unrelated domain knowledge.
- **Reduced hallucination rate:** grounding on real text eliminated the “made-up” or off-topic statements seen in the pure LLM output.

1. Briefly explain how you designed the RAG system, the issues you faced and how you solved them, and how you improved retrieval accuracy.(10%)

1. System Architecture

- **Data ingestion & preprocessing**
  - Converted each PDF page into a 150 dpi PNG image via pdf2image.
  - Ran Tesseract OCR on each image to extract raw text.
  - Passed the same image through Φ-4-multimodal-instruct to generate a natural-language caption.
  - Concatenated “=== OCR TEXT === ... === Φ-4 CAPTION === ..” as the document text for each page.
- **Embedding & indexing**
  - Used sentence-transformers/all-MiniLM-L6-v2 to embed each combined-text page.
  - L2-normalized embeddings and built an inner-product FAISS index (IndexFlatIP).
- **Retrieval & QA**
  - Embedded each query with the same MiniLM model, normalized, and retrieved the top-1 page via index.search.

2. Key Issues & Solutions

Issue	Solution
Low retrieval accuracy ( 40%) using captions alone	Treated OCR output as extra context—concatenated it with Φ-4 captions.
Noise and mis-OCR’d text	Tuned Tesseract’s --psm mode and restricted languages to “eng” for higher-precision OCR.
FlashAttention warnings / model on CPU	Explicitly set torch_dtype=torch.float16 and moved Φ-4 model to CUDA (device_map="cuda").
Unnormalized embeddings causing skewed similarity	Applied faiss.normalize_L2 to both document and query embeddings.

3. Improving Retrieval Accuracy

- **Baseline (no OCR): Captioning → All-MiniLM L6 v2: 40%** top-1 accuracy
- **With OCR + Captioning → All-MiniLM L6 v2: 61%** top-1 accuracy