

GenAI with Retrieval-Augmented Generation (RAG) Framework

REPORTER:

HSIGN-CHUAN HSIEH (謝幸娟)

DATA SCIENTIST @ ACER

A portrait of Jane Hsieh, a young woman with dark hair and glasses, wearing a red t-shirt and white earbuds. She is smiling slightly and looking towards the camera. The background is a blurred outdoor scene with a railing and greenery.

謝幸娟 Jane Hsieh

Data Scientist

❖ Experiences

Present - 2023/12	<u>Data Scientist</u> Acer (contracted through ConSem), TW
2023/02 - 2022/03	<u>Visiting Researcher (DAAD/NSTC Scholar)</u> Humboldt University of Berlin (HU Berlin), Germany (DE)
2016/01 - 2013/01	<u>Research Assistant</u> National Sun Yat-sen University (NSYSU), TW

❖ Education

2025/02 - 2016/09	<u>Ph. D. Candidate in Statistics</u> National Yang Ming Chiao Tung University (NYCU), TW
2012/06 - 2009/02	<u>MS in Psychology</u> National Chung Cheng University (CCU), TW
2008/06 - 2005/09	<u>BS in Psychology</u> National Chung Cheng University (CCU), TW

Outline

1. Preface
2. RAG Framework
3. RAG Hands-On Practice
4. More to Consider
5. References / Resources

Preface

1. OVERVIEW OF GenAI
2. LIMITATION OF GenAI
3. RISE OF RAG

Overview of GenAI

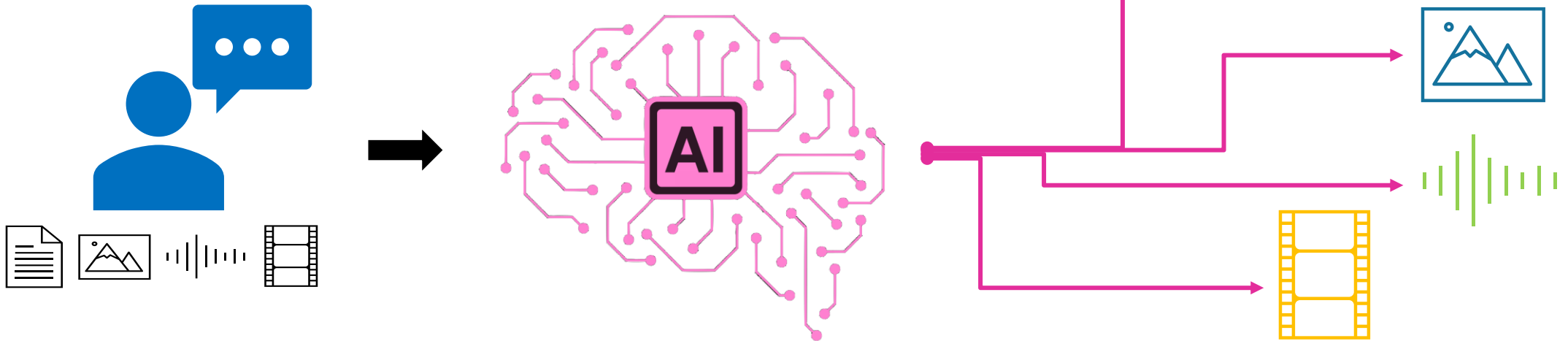
❖ **Generative AI:** describes models (**LLM**) that can generate new content—text, images, music—based on input data.

❖ **Large Language Models (LLMs):** act as Knowledge Bases (KB)

1. knowledge learned from training datasets
2. stored internally in the parameters (internal memory)
3. answers retrieved and generated from internal memory
 - E.g., GPT (OpenAI), Llama (Meta), Claude (Anthropic), Gemini (Google), ...

Note:

In this lecture only text input and output are considered



Limitation of GenAI / LLMs

1. **Lack of Domain-Specific Expertise:** difficulty in retrieving highly specific information
 - Training data source:
public websites (wiki, social media, news), open-source code depository (GitHub), eBooks, ...
2. **Outdated Knowledge:**
no access to new information occurring after the training period
3. **Knowledge Update Issues:**
 - data collection cost
 - data confidentiality
 - training cost

User

How to assemble a circuit board according to the FakeCompany's "New Employee Training Manual"

Sorry, but I don't have FakeCompany's data

User

How's the weather in Taichung today?

Sorry, but I don't have today's data

User left the chat

Limitation of GenAI / LLMs

4. Hallucination:

tendency to generate responses that are coherent and fluent but factually incorrect

ChatGPT ▾

Share



請介紹中興大學通識中心林建良教授的教學經歷



林建良教授是中興大學通識中心的一位資深教授，他在教學和學術研究方面具有豐富的經驗。以下是他的一些教學經歷：

1. **學術背景：**林教授擁有相關學科的博士學位，並在多個領域的學術研究中有著卓越的貢獻。
2. **教學職位：**他在中興大學通識中心任教，負責設計和教授多門通識課程，涵蓋人文社會科學、自然科學及跨領域研究等主題。
3. **課程設計：**林教授注重課程的多樣性和實用性，鼓勵學生從不同的視角思考問題，並培養批判性思維和創新能力。
4. **學生互動：**他積極與學生互動，鼓勵他們參與討論和實際案例分析，以提高學生的學習興趣和參與度。
5. **研究指導：**林教授也指導學生進行研究，特別是在通識教育相關的專題研究中，協助學生發展其研究能力和學術寫作技巧。
6. **學術貢獻：**除了教學外，林教授在學術期刊上發表過多篇論文，並參與各種學術會議，分享他的研究成果。

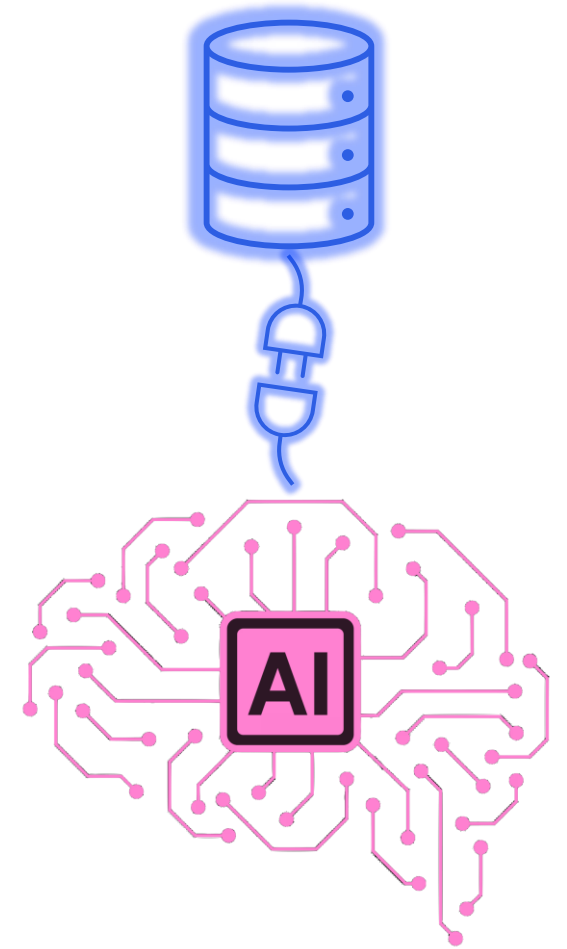
Rise of RAG

❖ Retrieval-Augmented Generation (RAG)

1. a framework incorporating **External Knowledge Bases (KB)** to augment LLMs
2. beyond the model's initial training

❖ RAG can address limitations of GenAI

Benefits	How
Domain-specific LLM	construct a domain-specific KB
Knowledge update issue addressed	update the external KB (with current data)
Hallucination alleviated	supply LLM with relevant factual information



RAG Framework

1. COMPONENTS
2. WORKFLOW

RAG Components

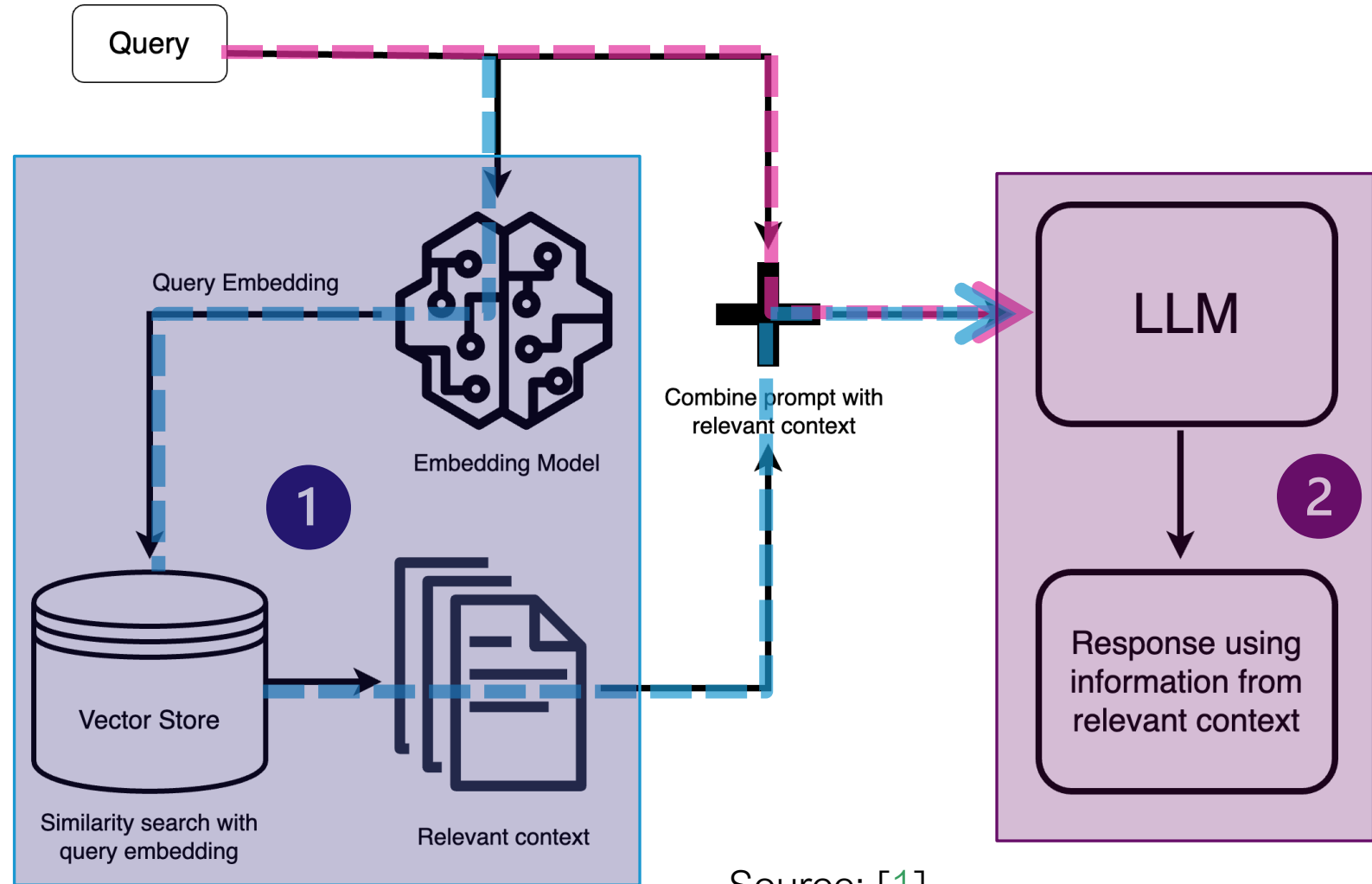
Framework Overview

❖ Components of RAG framework

1. Retriever (R)
2. Generator (G)

❖ Application

1. Conversational Agents (Chatbots)
 - Customer Service, virtual assistant
2. Advanced Question Answering (QA)

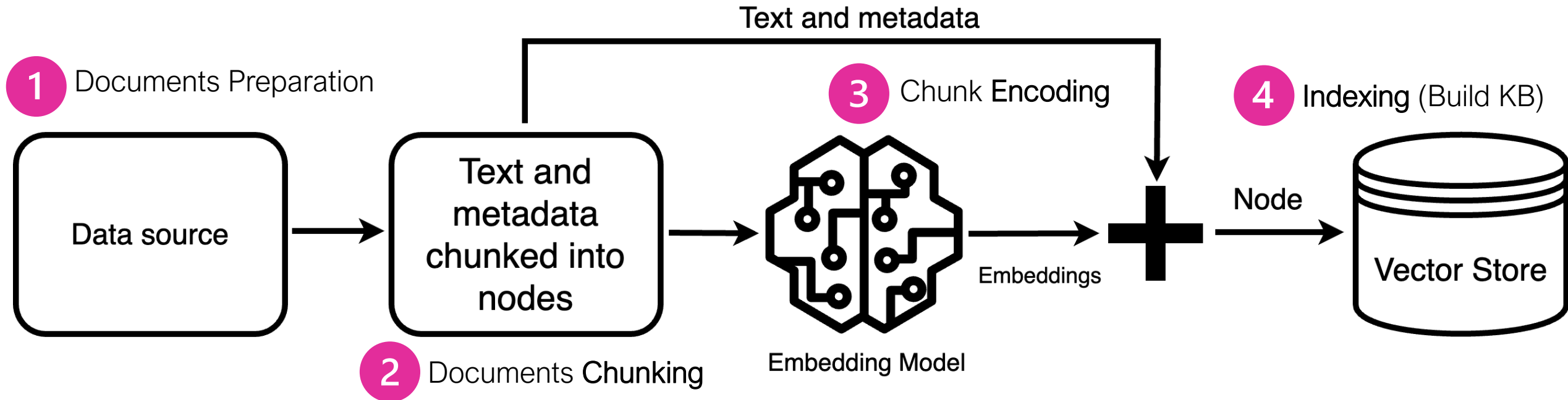


RAG Workflow: Retriever Built-up

Overview: Retriever Built-up

❖ Use Case: AI Customer Service Chatbot

- Build a **Knowledge Base (KB)** about a company's products
- User can ask question related to their products (e.g., repair, product info., product issues, ...)



Source: [1]

1. Documents Preparation

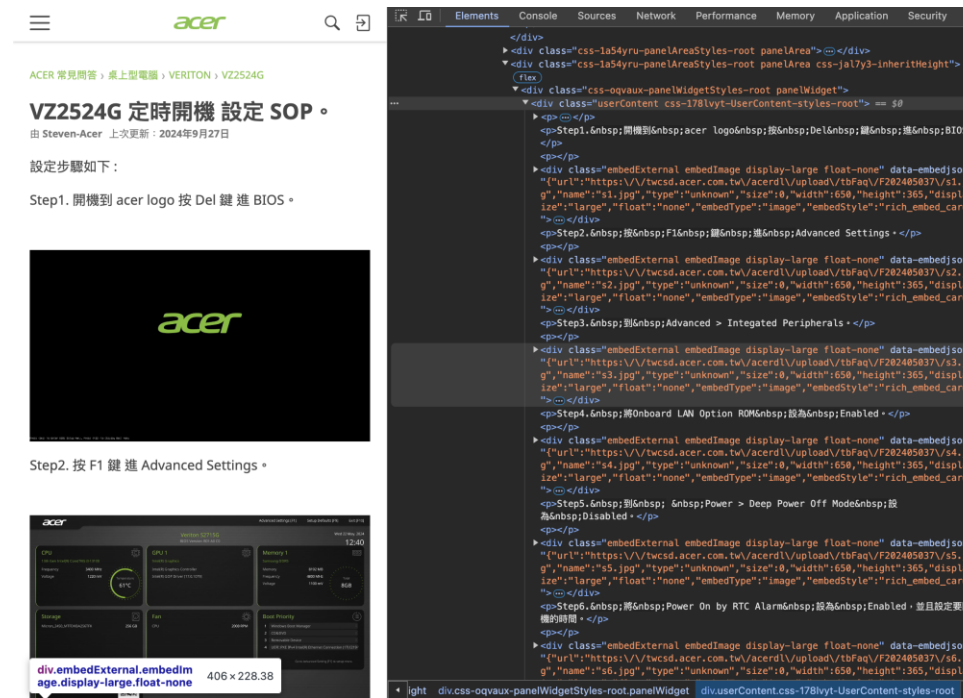
❖ Word documents should be converted to plain text

❖ Most common document formats:

- Documents (PDFs)
- Word Documents (DOCX)
- Text Files (TXT)
- Spreadsheets (CSV/XLSX)
- Web pages (HTML)
- etc.



HTML (e.g., Website)



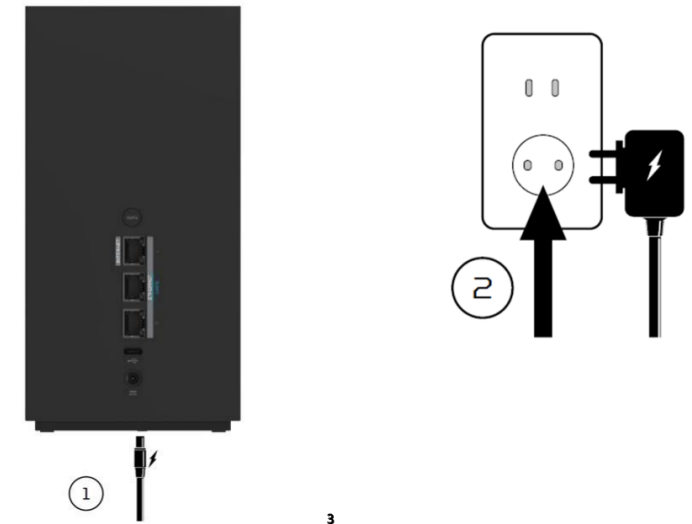
1. Overview

Acer Predator Connect Series X7 is a 5G CPE with a cutting-edge Wi-Fi 7 Tri-band (2.4GHz + 5GHz + 6GHz) simultaneously with BE11000 throughput, specifically optimized for gamers. It boasts dual WAN features and an easy 1-2-3 setup wizard for a hassle-free installation. Enjoy continuous connectivity with the seamless transition between your primary 5G NR and secondary Ethernet internet network. Our integrated load balancer and failover features ensure optimal network performance and robustness across your internet network. Unlock the ultimate gaming experience with Wi-Fi 7's groundbreaking technology, designed for peak data transmission and minimal latency. Wi-Fi 7's Multi-Link Operation (MLO) is a significant technical advancement, enhancing throughput, reducing latency, and improving network efficiency by allowing devices to connect to multiple links simultaneously. This CPE includes band steering, which ensures that each device uses the optimal frequency band for its conditions, resulting in a more efficient and reliable Wi-Fi experience. Trend Micro network security protection is built-in, with live updates keeping your network safe from malware and vulnerabilities around the clock. Automatic Channel Selection (ACS) dynamically selects the best channel to avoid interference from nearby networks. The X7 also includes port forwarding profiles for popular gaming consoles like PS5 and Xbox, facilitating seamless gameplay. Hybrid QoS prioritizes your gaming traffic and optimizes bandwidth utilization. Additionally, the VPN feature provides a secure connection for your device when browsing online.

2. Installation and Setup

2.1. Plug in the AC adapter

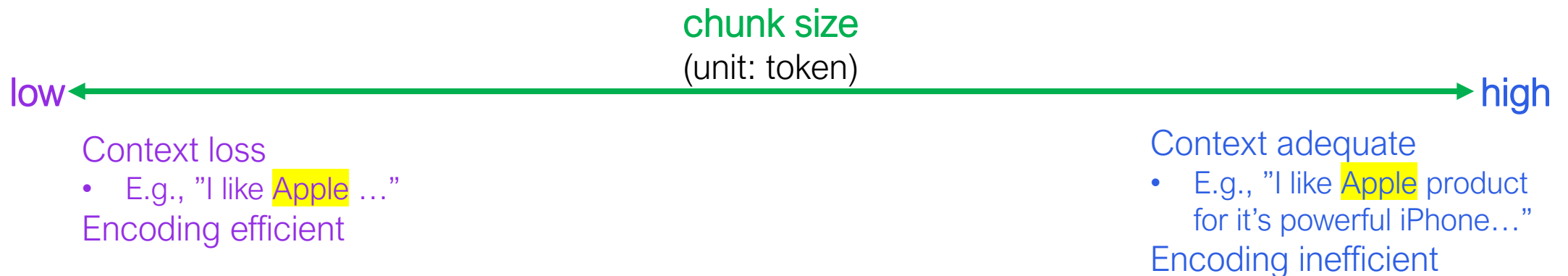
2.2. Plug into an outlet.



PDF (e.g., User Manual)

2. Documents Chunking

- ❖ **Chunking:** techniques to divide large documents into bite-sized text chunks
 - Each chunk should contain one core idea (e.g., one chunk for Product A issue, one for Product B info., ...)
- Types of chunking techniques:
 - 1) Chunking with fixed length: split documents sequentially by a **chunk size** (unit: token)
 - 2) Semantic chunking: cuts documents based on semantics that represents the end of the sentence (e.g., period character (.), newline character (/n)).
 - 3) Content-based chunking: segments documents according to the unique structural characteristics. (e.g., function blocks for codes, HTML tags for HTML contents)



2. Documents Chunking

1) Chunking with fixed length

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

Source: [2]

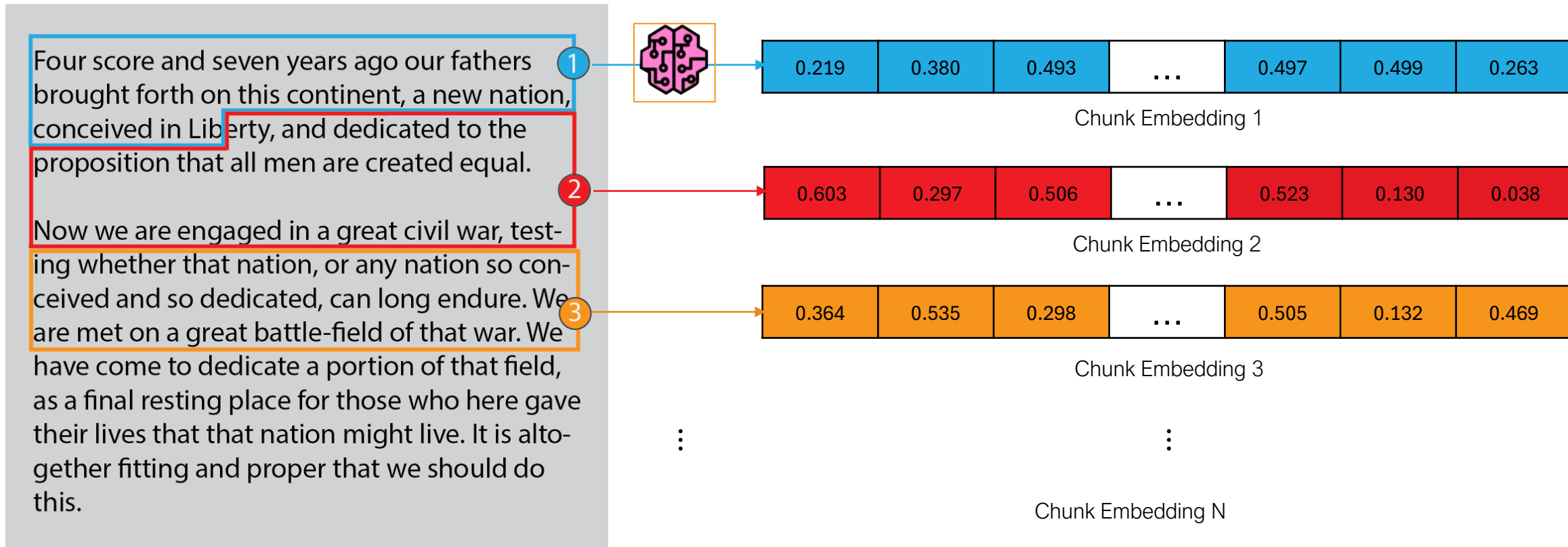
3) Content-based chunking

```
</div>
<div class="css-1a54yru-panelAreaStyles-root panelArea">
  <div class="css-1a54yru-panelAreaStyles-root panelArea css-jal7y3-inheritHeight">
    <div class="css-oqvaux-panelWidgetStyles-root panelWidget">
      <div class="userContent css-178lvyt-UserContent-styles-root">
        <p>Step1. 開機到 BIOS 設定 </p>
        <div class="embedExternal embedImage display-large float-none" data-embedjson="{"url":"https://twcsd.acer.com.tw/acerdl/upload/tbFaq/F202405037/s1.jpg","name":"s1.jpg","type":"unknown","size":"0","width":"650","height":"365","displaySize":"large","float":"none","embedType":"image","embedStyle":"rich_embed_card"}">
        <p>Step2. 按 F1 鍵進入 Advanced Settings </p>
        <div class="embedExternal embedImage display-large float-none" data-embedjson="{"url":"https://twcsd.acer.com.tw/acerdl/upload/tbFaq/F202405037/s2.jpg","name":"s2.jpg","type":"unknown","size":"0","width":"650","height":"365","displaySize":"large","float":"none","embedType":"image","embedStyle":"rich_embed_card"}">
        <p>Step3. 到 Advanced > Integrated Peripherals </p>
        <div class="embedExternal embedImage display-large float-none" data-embedjson="{"url":"https://twcsd.acer.com.tw/acerdl/upload/tbFaq/F202405037/s3.jpg","name":"s3.jpg","type":"unknown","size":"0","width":"650","height":"365","displaySize":"large","float":"none","embedType":"image","embedStyle":"rich_embed_card"}">
        <p>Step4. 將 Onboard LAN Option ROM 設為 Enabled </p>
        <div class="embedExternal embedImage display-large float-none" data-embedjson="{"url":"https://twcsd.acer.com.tw/acerdl/upload/tbFaq/F202405037/s4.jpg","name":"s4.jpg","type":"unknown","size":"0","width":"650","height":"365","displaySize":"large","float":"none","embedType":"image","embedStyle":"rich_embed_card"}">
        <p>Step5. 到 Power > Deep Power Off Mode 設為 Disabled </p>
        <div class="embedExternal embedImage display-large float-none" data-embedjson="{"url":"https://twcsd.acer.com.tw/acerdl/upload/tbFaq/F202405037/s5.jpg","name":"s5.jpg","type":"unknown","size":"0","width":"650","height":"365","displaySize":"large","float":"none","embedType":"image","embedStyle":"rich_embed_card"}">
        <p>Step6. 將 Power On by RTC Alarm 設為 Enabled, 並且設定要開機的時間 </p>
        <div class="embedExternal embedImage display-large float-none" data-embedjson="{"url":"https://twcsd.acer.com.tw/acerdl/upload/tbFaq/F202405037/s6.jpg","name":"s6.jpg","type":"unknown","size":"0","width":"650","height":"365","displaySize":"large","float":"none","embedType":"image","embedStyle":"rich_embed_card"}">
```

3. Chunk Encoding

❖ **Encoding**: techniques to transform input texts (user's query or document chunks) into **numerical vector representation** (aka. **embedding**) that capture semantic meaning of the texts

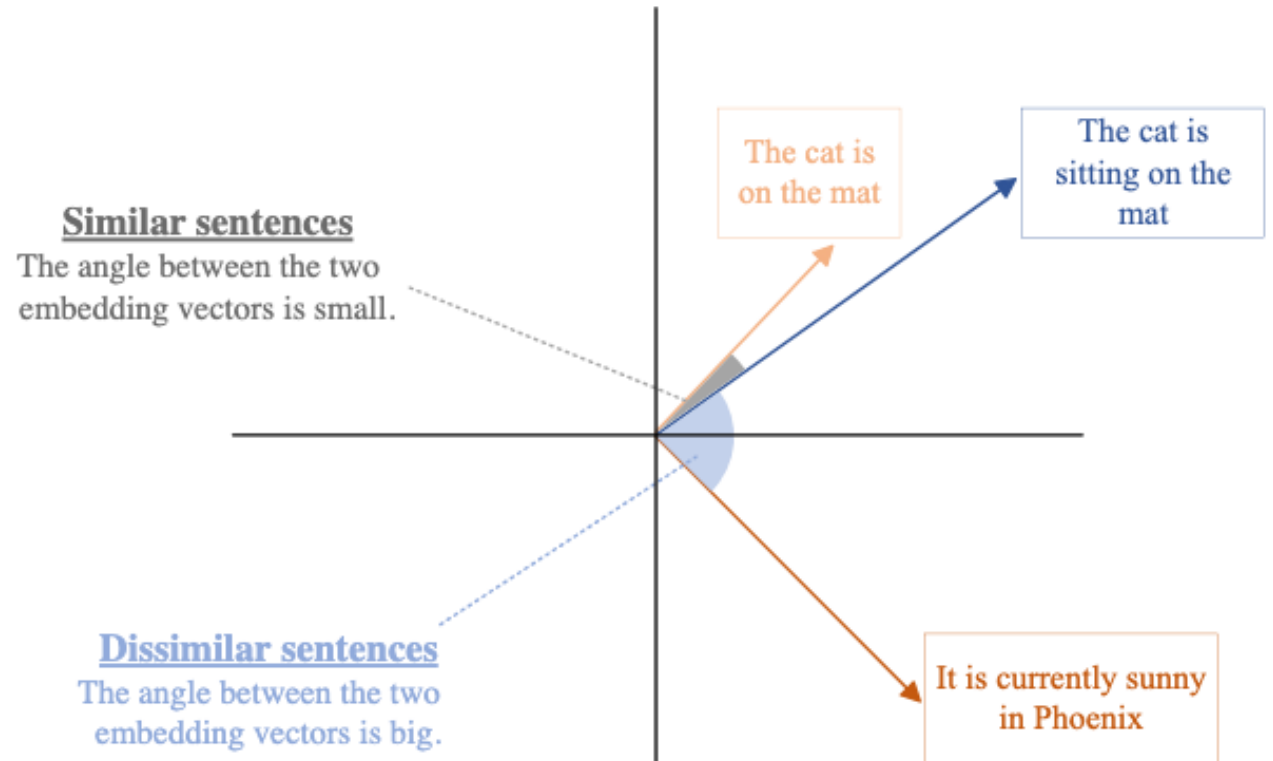
- LLMs used for encoding are '**Encoders**', e.g., "sentence-transformers/all-mpnet-base-v2"



3. Chunk Encoding

❖ Properties of Embeddings

- Words / phrases with similar meanings (i.e., semantics) have similar embeddings and hence are close in the high-dimensional space.
- Embeddings enables the retriever to search for similar information (chunks) based on query content.



Source: [3]

4. Indexing (Build a KB / an Index)

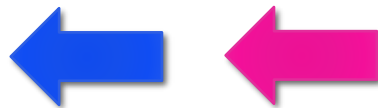
❖ **Indexing**: a process to transform data (embeddings and metadata) and store them in a **Vector Database (VectorDB)**/ Vector store, aka. **Knowledge Base (KB)**

- **Metadata**: data that provides information about other data, but not the content of the data itself (e.g., text, embeddings)

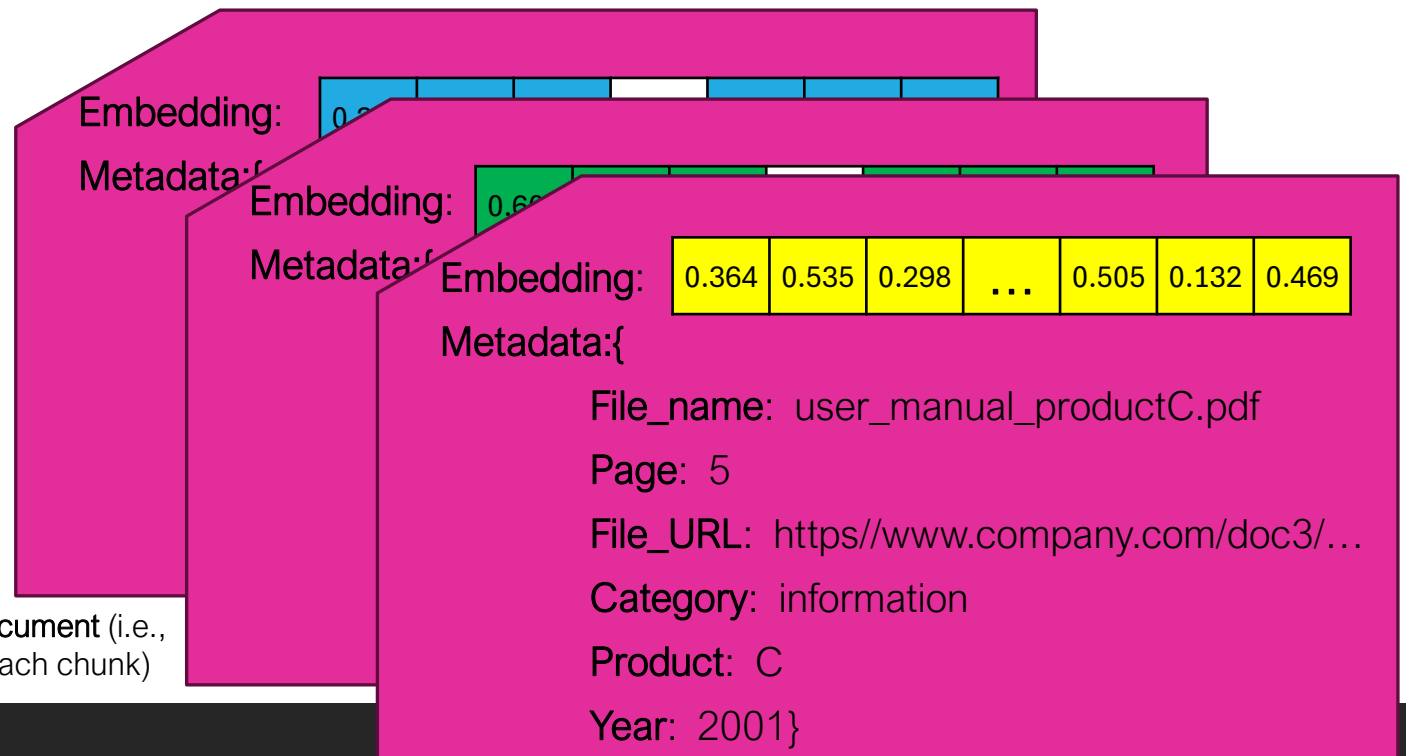
❖ **VectorDB** aims to accelerate the search process for data (e.g., chunk embeddings) similar to the query embedding.



1. Create a **container**
(aka. Collection)

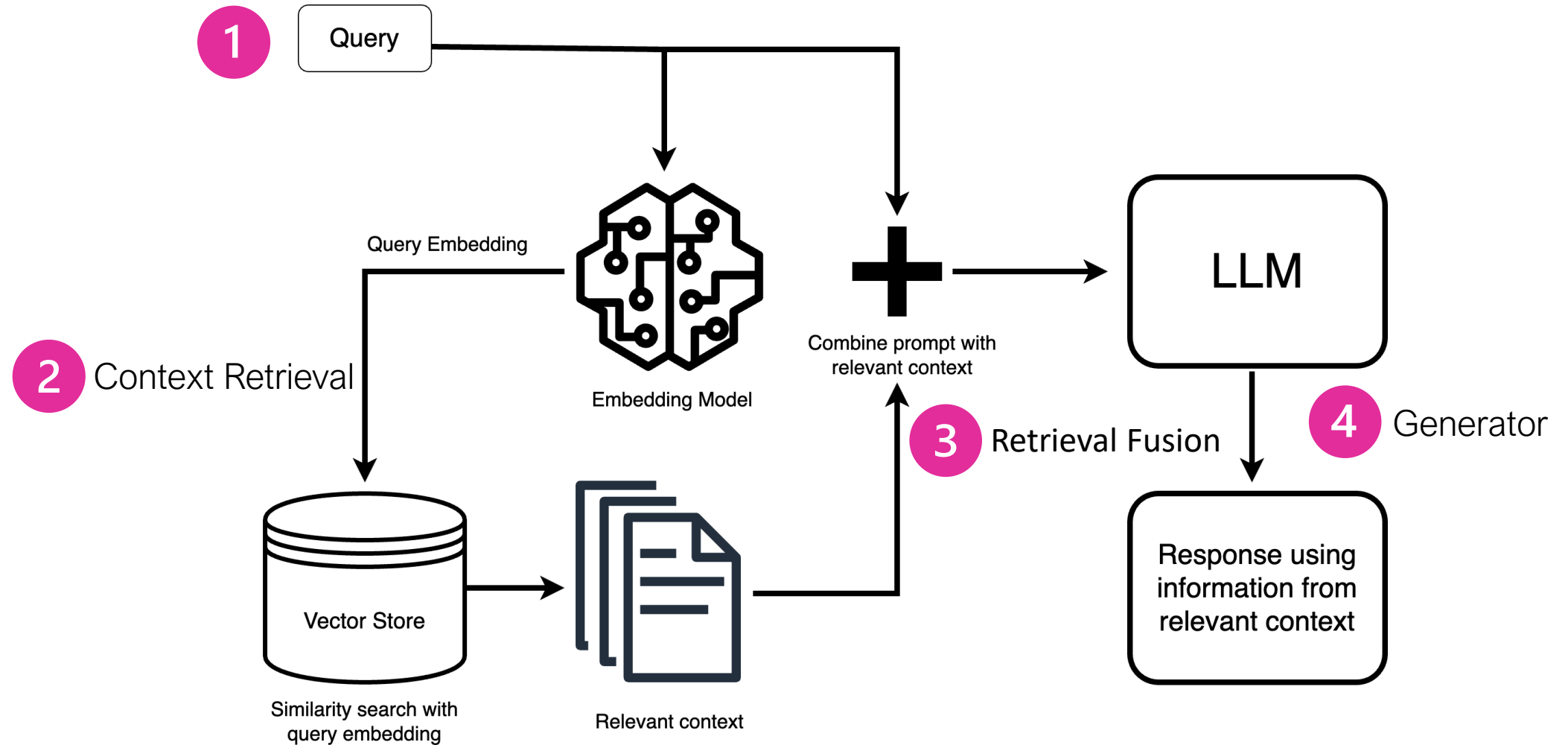


2. Prepare **document** (i.e.,
content for each chunk)



RAG Workflow: Augmented Generation

Overview: Augmented Generation

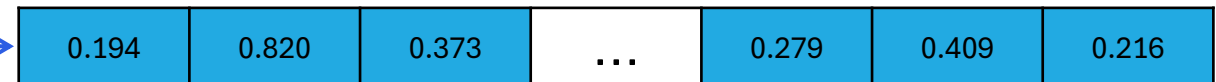
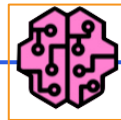


Source: [1]

1. Query

- ❖ Input: **Query**
- ❖ Input Processing: **Query Encoding**
 - To align with the pre-built embedding space, the retriever **MUST** use the same encoder to encode queries

"Hi, I just purchased a **smartphone** from your store, but it's not holding a charge. I've tried different chargers, but the battery still drains quickly. Can you help me figure out what's wrong or how to get a replacement?"



Query Embedding

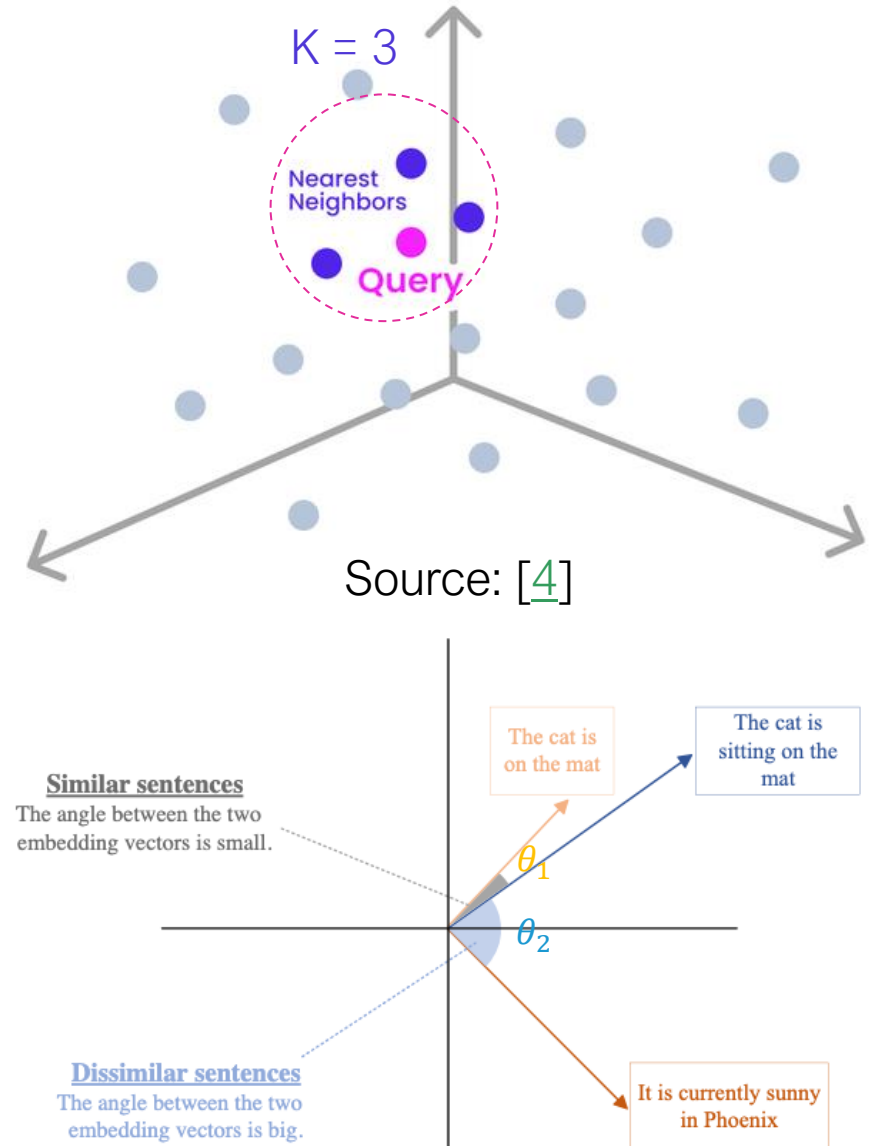
2. Context Retrieval

❖ **Context Retrieval:** retrieve relevant information from KB based on query embedding

- Search method: **Vector / Semantic / Similarity Search**
 - Algorithm: **Approximate Nearest Neighbor (ANN)**
(Efficient version of KNN search)
 - Return the top-k nearest neighbors (chunk embeddings) to the query embedding

❖ **Distance Metric:** estimate closeness (similarity)

- ✓ cosine distance: $1 - \text{cosine similarity} = 1 - \frac{\cos(\theta_{ab})}{\|a\| \|b\|}$ ($\in [0, 2]$)
- Euclidean distance = $\|a - b\|$ ($\in [0, \infty]$)
- More ...



3. Retrieval Fusion

❖ **Retrieval Fusion:** technique to leverage the retrieved knowledge (context), in order to improve the Generator's performance.

- Simplest way: Query-based Fusion with text concatenation
 - put the context and query together as an input (i.e., **prompt**) to the Generator (LLM)

❖ **Prompt:** to instruct the model on what kind of response to generate

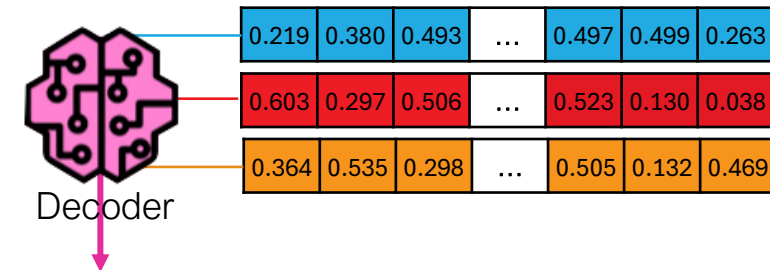
“”“Your’e a professional customer service agents. You can (not) do ...(omitted)

You need to answer the following question based on the knowledge:

Knowledge:

{**context**}

Question: {**query**}



[1] You can check the battery health of your smartphone by going to Settings > Battery > Battery Health....

[2] Our store offers a warranty that covers battery issues for the first year....

[3] Please ensure that you’re using the original charger provided with the device, as third-party chargers can sometimes cause battery issues.

"Hi, I just purchased a **smartphone** from your store, but it's not holding a charge. I've tried different chargers, but the battery still drains quickly. Can you help me figure out what's wrong or how to get a replacement?"

4. Generator

- ❖ **Generator:** pretrained LLM which takes a prompt in order to produce the final response
 - Most LLMs adopt Transformer-based architecture; e.g., Llama, GPT, Gemini, Claude.
- ❖ Elements of Prompt (Input)
 1. **Instruction (System Prompt):** guidance on how the model should behave or respond to the user's query
 - E.g., role, task, style, examples...
 2. **Query (User prompt):** user's question / request
 3. **Retrieved Context (RAG only):** additional information or knowledge retrieved from an external source
- ❖ Importance
 - ❖ Watch out of the “**token limitation**” – restrain the length of the prompt (as well as the generated output) within the limit of each LLM



RAG Hands-On Practice

1. PROJECT:
BUILD AI CAREER
CONSULTANT
2. PREPARATION
3. CODES

Project: Build AI Career Consultant

❖ Goal: Find your dream job

➤ Context: Job Search Preparation

❖ Strategy

- Build an **AI Career Consultant** who helps job seekers to make strategic plans according to the current job trend
 - Who recommends the **suitable trending jobs** based on your current experiences
E.g., education, abilities, projects, jobs, etc.
 - Get **further advice** according to the job recommendation and your experiences

autobiography
I recently graduated with a Bachelor degree in Computer Science, I use Python and have good grades in machine learning and deep learning. I had various projects that allowed me to apply these skills, from building predictive models to analyzing large datasets. I am now seeking an entry-level data scientist or data analyst role.



Job Seeker

Based on your experience and current job trend, my advice is ...



AI Career Consultant

Preparation

1. Application for API Keys

To build our AI assistant, we needed to apply for several keys from following service providers

➤ Note: these keys are sensitive information, do not expose them other than yourself.

1. Kaggle API: to download data
2. Hugging Face API: to call embedding model
3. Gemini API: to call generation model


2. Self Description

To provide your brief autobiography for your consultant's reference

1. Imagine you're preparing your resume, what information should you put? (E.g., education, experience, abilities, personalities, job position you're looking for, etc.)
2. The words (texts only) need not be too long (< 500 words)

Preparation: Kaggle API

❖ Purpose: to download data directly via Kaggle API (without manual downloading)


1. Sign in or Register a Kaggle account (if you haven't any): <https://www.kaggle.com/account/login>
2. After login, click on your **profile icon**  (top right corner) > click **Settings**
3. In Settings, scroll down to the **API** section > click the **Create New Token** button.
 - This will automatically download a JSON file (kaggle.json) containing your API credentials.
 - Save it in a secure location on your computer.
4. Open **kaggle.json** file (with text editor) > Get Kaggle API information:
 - `{"username": "YourUserName", "key": "Your_32_digit_token"}`
5. Remember this information as you need to store it as Colab's environment variables (in Secrets section):

Name	Value
KAGGLE_USERNAME	<i>YourUserName</i>
KAGGLE_KEY	<i>Your_32_digit_token</i>

❖ Full tutorial: [How to Obtain a Kaggle API Key](#)

Preparation: Hugging Face API

❖ Purpose: to call embedding model for text encoding

1. Sign in or Register a Hugging Face account (if you haven't any): <https://huggingface.co/login>
2. After login, click on your **profile icon**  (top right corner) > click **Settings**
3. In Settings, click the **Access Tokens** button (left sidebar) > click **Create next token** (right)
 - Select token type as 'Read', and name yourself the **Token Name**, click **create token** once you're done!
4. Copy and save your Access Token: `{YourAccessToken}`
 - Keep it secure as it provides access to your Hugging Face resources.
5. Remember this information as you need to store it as Colab's environment variables (in Secrets section):

Name	Value
HF_TOKEN	<code>YourAccessToken</code>

❖ Full tutorial: [How to Access HuggingFace API key?](#)

Preparation: Gemini API


❖ Purpose: to call generation model for query-based content creation

1. Sign in **Google AI Studio** with your Google account: <https://aistudio.google.com/app/welcome>
2. After login, click **Get API Key** (on top of left sidebar) > click **Create API key** (right)
 - In the pop-up window, randomly select an existing Google Google Cloud projects (e.g., Gemini API) and create API key
3. Copy and save your Access Token: *{YourAccessToken}*
 - Keep it secure as it provides access to your Hugging Face resources.
4. Remember this information as you need to store it as Colab's environment variables (in Secrets section):

Name	Value
GEMINI_API_KEY	<i>YourAccessToken</i>

❖ Full tutorial: [Create a Gemini AI key](#)

Codes Fetching

1. Go to **Colab** : Welcome To Colab – Colab
2. In the “Open notebook” window, click **GitHub** (left sidebar)
 - If you don't see the window, then click File > Open notebook, then the window will pop up
3. Copy and paste the GitHub URL: <https://github.com/DreamBird-Jane/GenAI-Application>
4. Select the path: LinkedIn Job Posting Recommendation and Advice Using GenAI/Build_....ipynb
5. After opening the notebook, set your keys in **Secrets** (left sidebar with  symbol):

Name	Value
KAGGLE_USERNAME	<i>YourUserName</i>
KAGGLE_KEY	<i>Your_32_digit_token</i>
HF_TOKEN	<i>YourAccessToken</i>
GEMINI_API_KEY	<i>YourAccessToken</i>

6. Congrats, you can start to run the code!!

More to Consider

1. RAG
ENHANCEMENTS
2. WRAP-UP

RAG Enhancements

Stage	Issues	Strategies
Document Preprocessing	<ul style="list-style-type: none">• Specific data: e.g., tables, images in PDF	<ul style="list-style-type: none">• Conversion (via packages): table parser, image caption, ...
Chunking	<ul style="list-style-type: none">• Context loss	<ul style="list-style-type: none">• Tuning on chunk sizes, overlap size• Chunking technique selection
Encoding & Indexing	<ul style="list-style-type: none">• Encoder trade-off: info richness vs computation expense• Technical language unfamiliar to embedding model	<ul style="list-style-type: none">• Selection of suitable encoder (LLM)• Encoder (LLM) fine-tuning
Querying	<ul style="list-style-type: none">• Small wording variations lead to different outcomes	<ul style="list-style-type: none">• Query rewriting (by another trained LLM)• Subquery generation (by LLM)
Retrieval	<ul style="list-style-type: none">• Irrelevant retrieved info (chunks)• Retrieval efficiency (speed)	<ul style="list-style-type: none">• Trade-off: top-k document vs token limit• Different search algorithms, reranking• Useful metadata for filtering before search
Generation	<ul style="list-style-type: none">• Output quality & length	<ul style="list-style-type: none">• Selection of LLM generators• Prompt engineering

Wrap-up

Build your own RAG application!

1. Define your query scope
2. Relevant data preparation & preprocessing
3. Build Vector DB
4. Prompt engineering for better outcomes
5. Start querying!

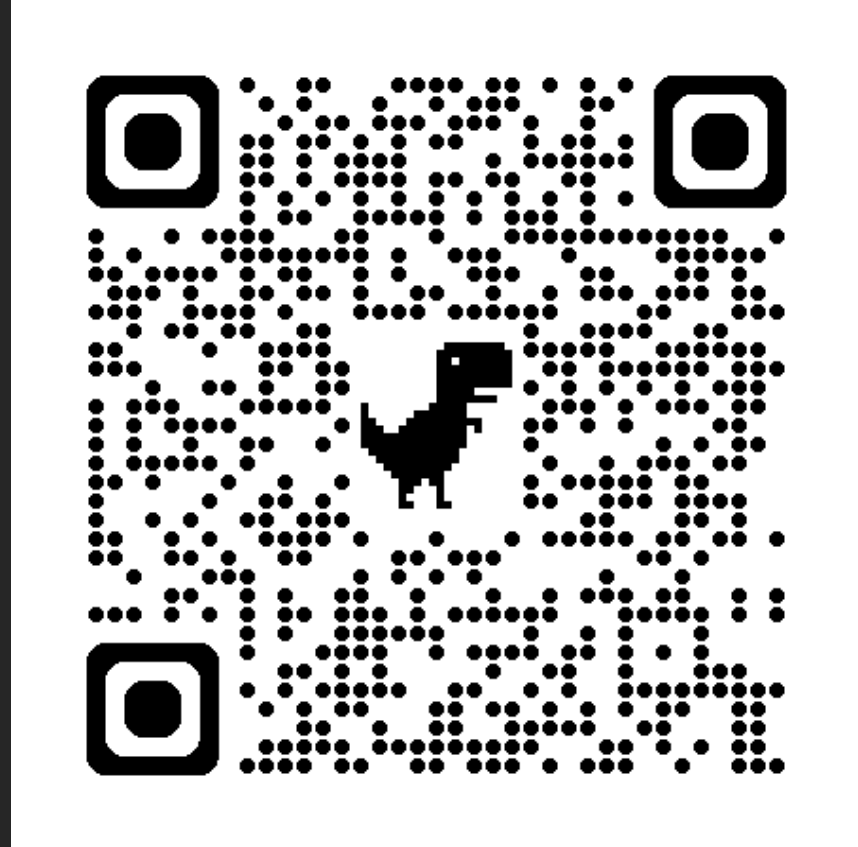
References / Sources

References

1. Wu, S., Xiong, Y., Cui, Y., Wu, H., Chen, C., Yuan, Y., ... & Xue, C. J. (2024). Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*.
2. Kamath, U., Keenan, K., Somers, G., & Sorenson, S. (2024). Retrieval-Augmented Generation. In *Large Language Models: A Deep Dive: Bridging Theory and Practice* (pp. 275-313). Cham: Springer Nature Switzerland.
3. 避免AI幻覺RAG漸興起 防外洩「私有」當道 | 網管人. (2024). Netadmin.com.tw.
<https://www.netadmin.com.tw/netadmin/zh-tw/trend/5DA70E16C2A747C9BF391AB4E2C04383>

Thanks for Your Attention!

Welcome to Connect:



Jane (Hsing-Chuan) Hsieh 謝幸娟
www.linkedin.com/in/jane-hsing-chuan-hsieh