

# **Research on implementing customized local LLM based on speech recognition**

# Contents

---

- 01 **Introduction**
- 02 **Core Technology**
- 03 **System architecture**
- 04 **Performance Comparison**

# 01 Introduction: Reasons for starting research

---



- By using API, there are financial, security issues
- It is not good for obtaining specific information due to the hallucination phenomenon, which is an inherent problem of auto regressive LM.

# 01 Introduction: How to solve the problem & necessity

---

- **Security and financial problem**

→ This problem was solved using Llama3, a large language model provided by Meta AI under open source and free license.

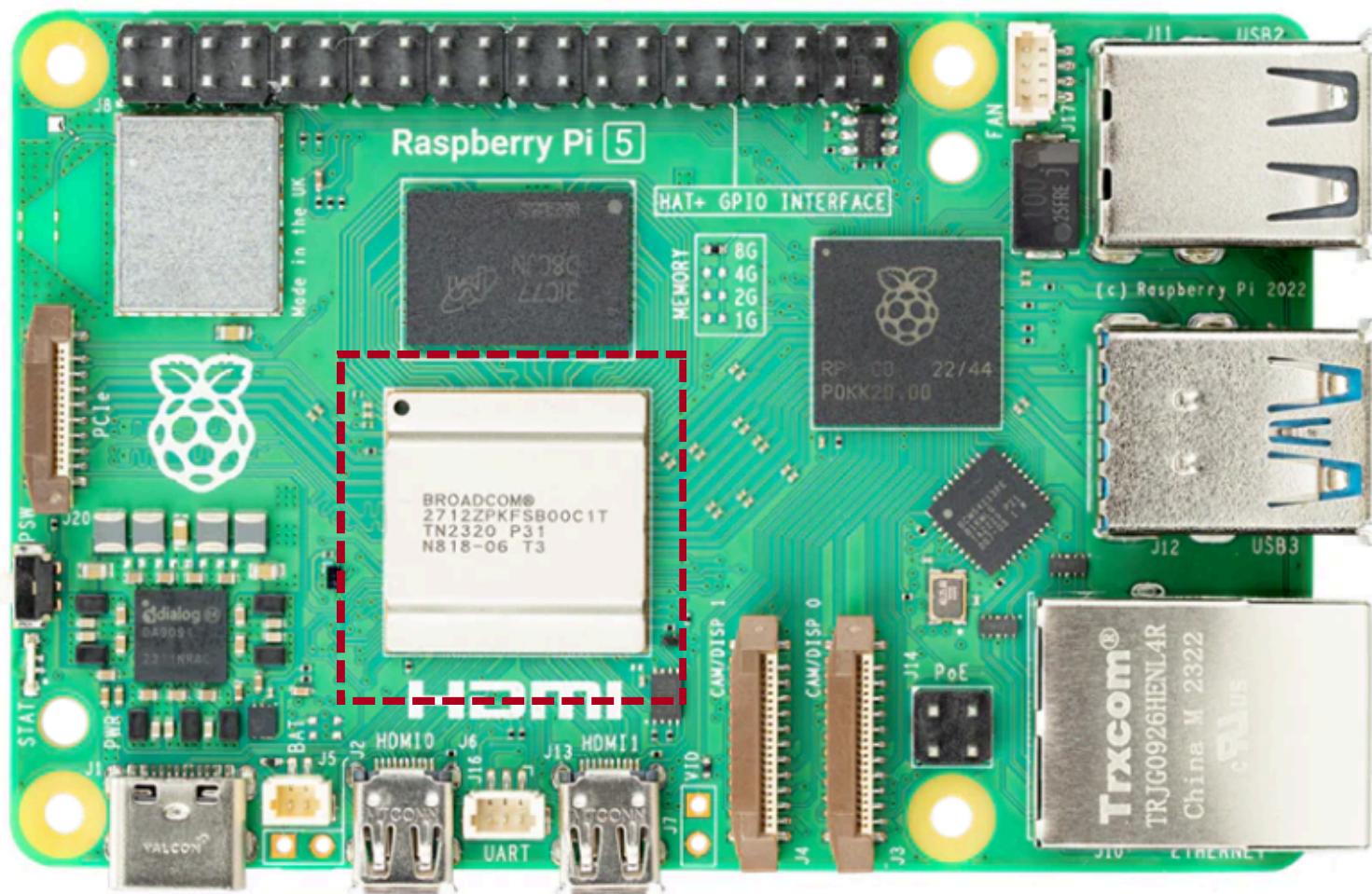
- **Hallucination problem**

→ Solve the problem by customizing it to the domain you want to apply it to using RAG and PEFT.

- **Mobile necessity**

→ Because the current trend is that AI can operate anytime, anywhere, it was implemented using Raspberry Pi 5, an educational board and additionally implemented voice recognition.

## 02 Core Technology: Raspberry Pi 5



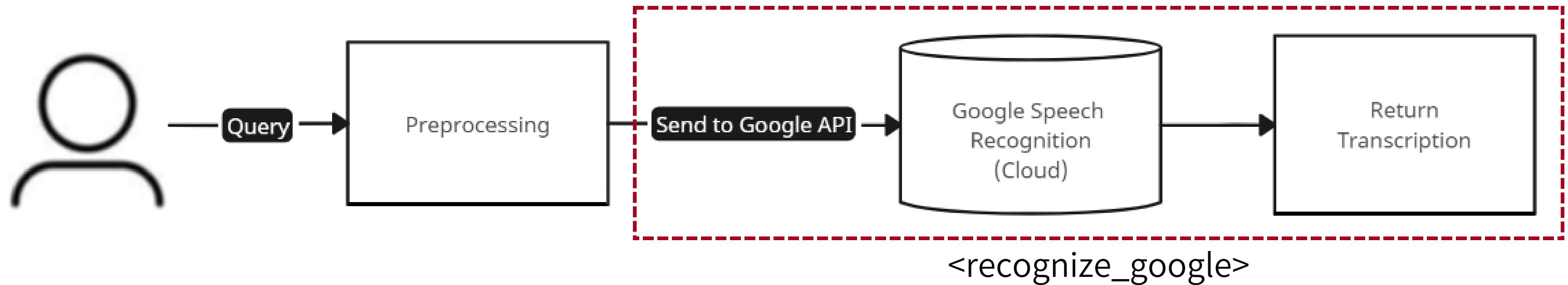
(Raspberry Pi 5)

항목		내용
H/W	CPU	BCM2712 (2.4GHz)
	GPU	VideoCore VII (800MHz)
	MEMORY	SDRAM 4267
	SD card	micro 카드 슬롯, SDR104 고속 모드 지원
S/W	O/S	Debian GNU/Linux 12
	Library	langchain_community = 0.2.7 speechrecognition = 3.10.7 langserve = 0.2.2, pyttsx3 = 2.90

(implementation environment)

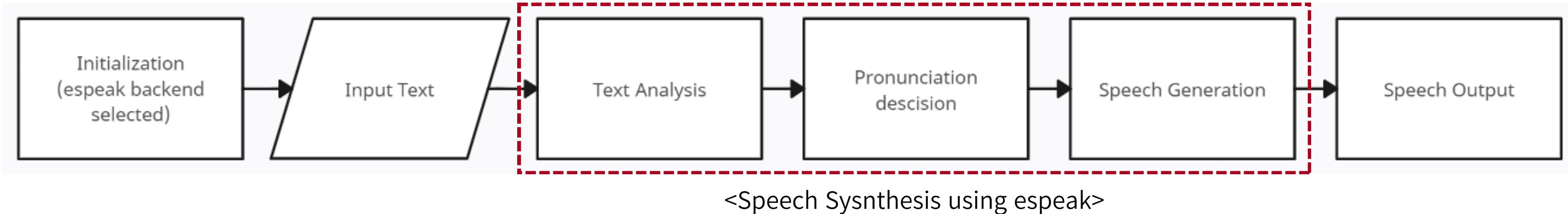
- Raspberry Pi 5 is a learning board equipped with a high-performance ARM Cortex-A76-based quad-core CPU.
- ARM Cortex-A76 implements CPU, GPU, and image processor in one chip

## 02 Core Technology: SpeachRecognition



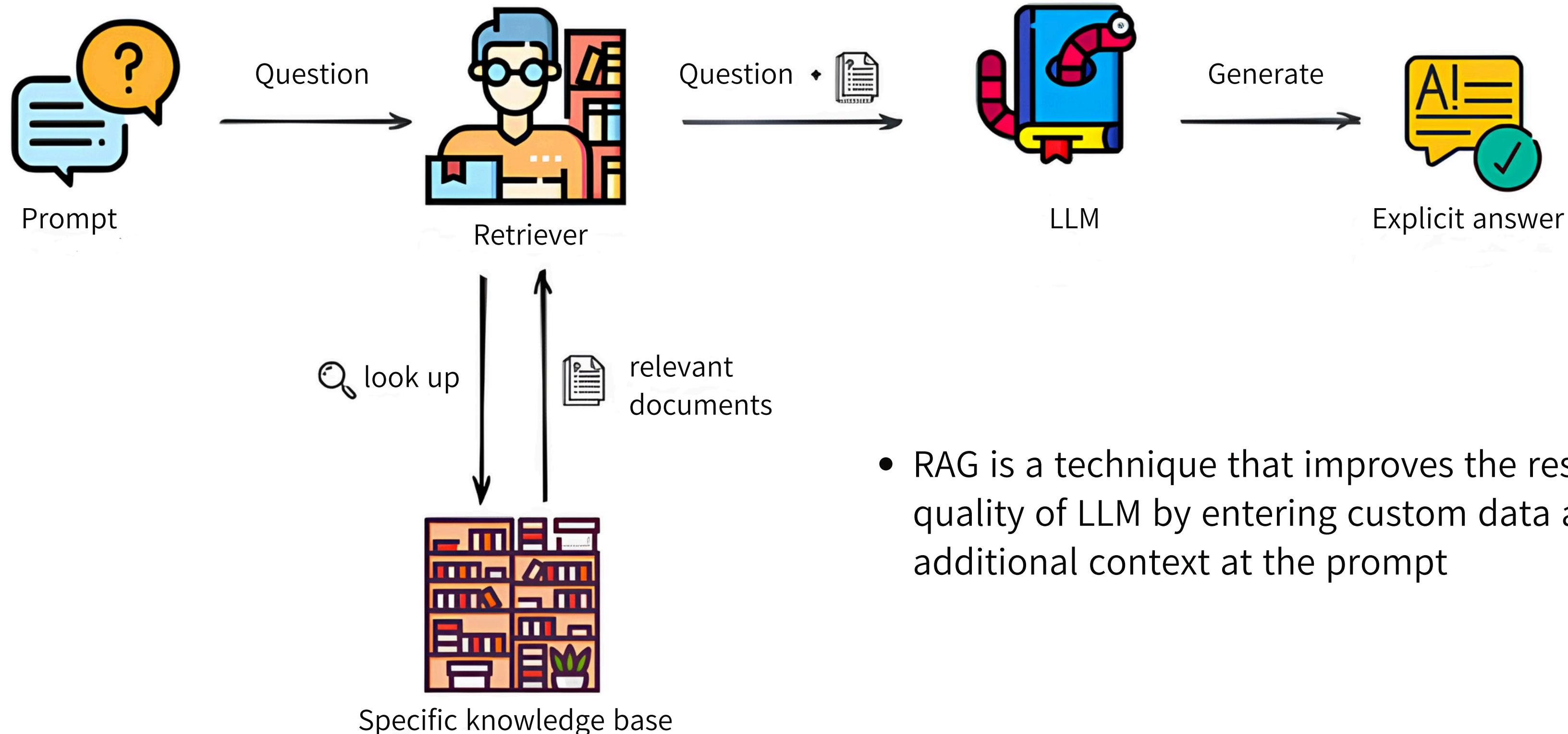
- This is the SpeachRecognition library flowchart, the STT algorithm used
- After preprocessing the query, voice recognition is performed with Google api using the recognize\_google function.
- Google API is an undisclosed algorithm now

## 02 Core Technology: pyttsx3

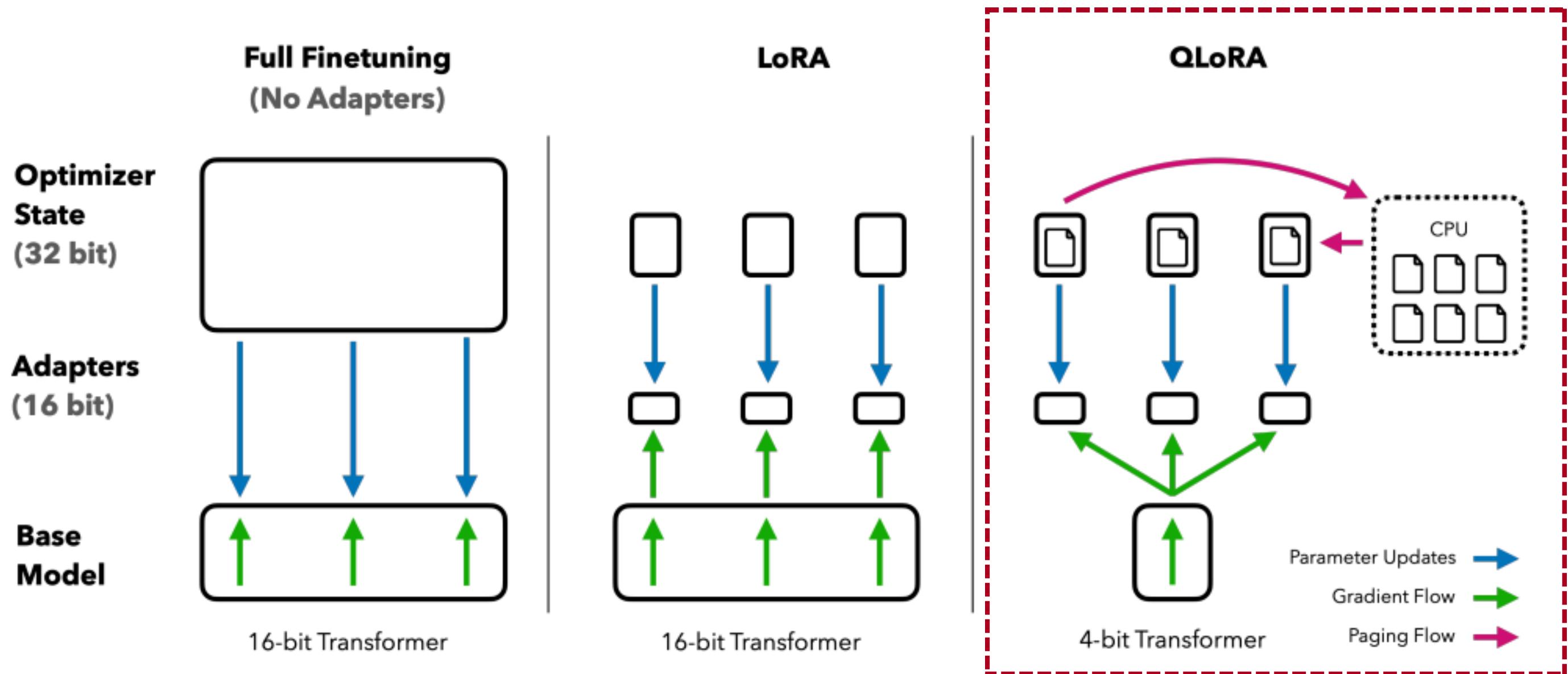


- This is a flowchart of pyttsx3, the TTS algorithm used
- The os of the Raspberry Pi 5 is Linux, so it uses the corresponding engine, espeak.
- espeak is a lightweight TTS open-source engine that is fast and uses less cpu and memory
- When text is inserted, the espeak proceeds with the TTS and outputs it to the speaker

## 02 Core Technology: RAG(Retrieval Augmented Generation)



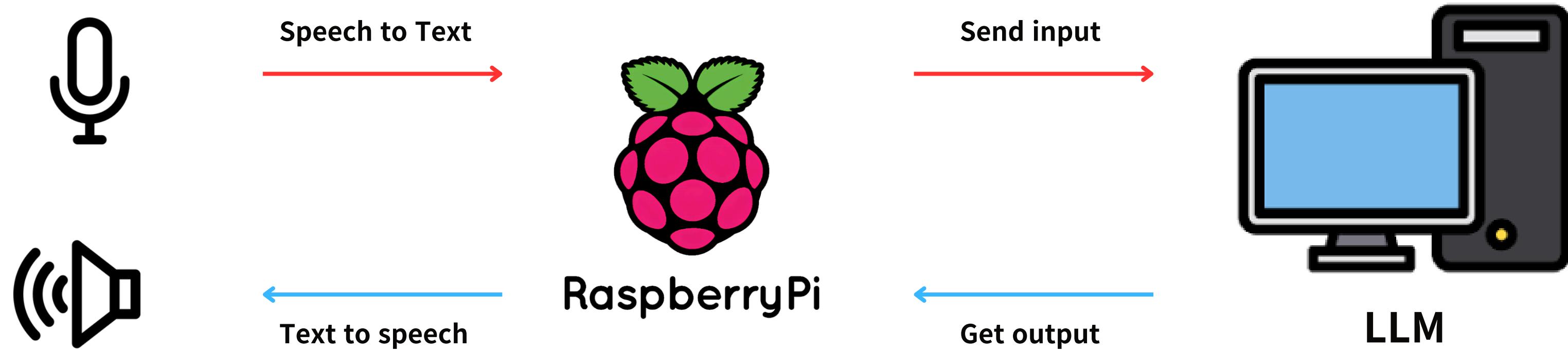
## 02 Core Technology: PEFT(Parameter Efficient Fine-tuning)



- PEFT is a technology that freezes most parameters and fine-tune only a few parameters.
- QLoRA (Quantized Low-Rank Adaptation) is a technology that adds quantization to LoRA, utilizing low-rank representations to efficiently learn LLM.

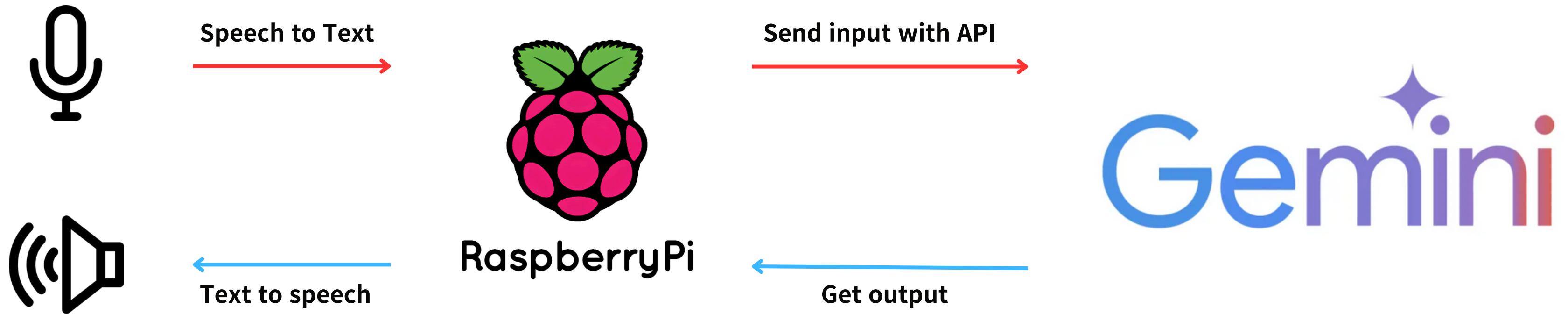
## 03 System architecture

---



- Implementing voice recognition using LLM and Raspberry Pi 5
  1. Use API(Gemini)
  2. Use Llama3
  3. Using Llama3 with RAG and PEFT

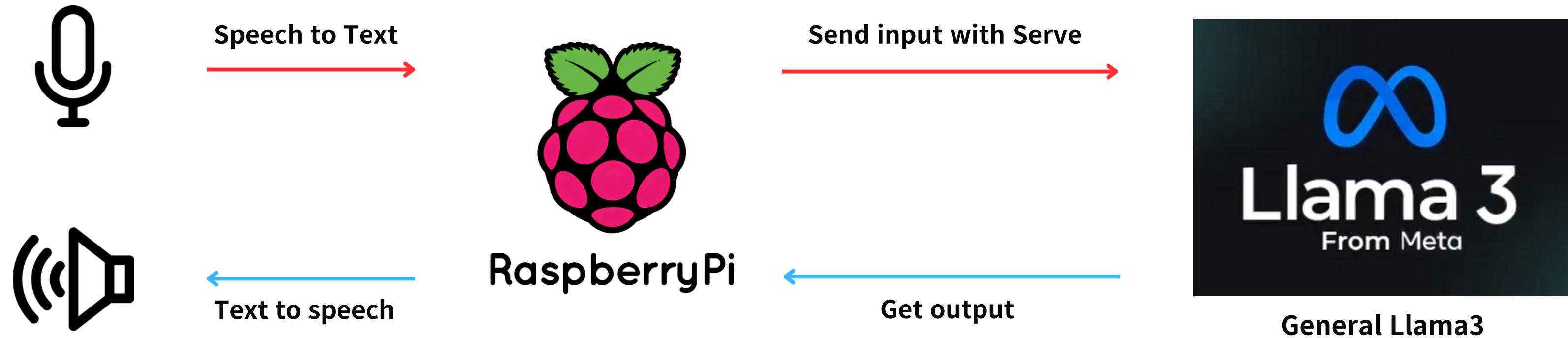
### 1. Use API(Gemini)



- We tested using Gemini's API to test whether speech recognition was working properly.
- For the Gemini version, we used Gemini-pro.

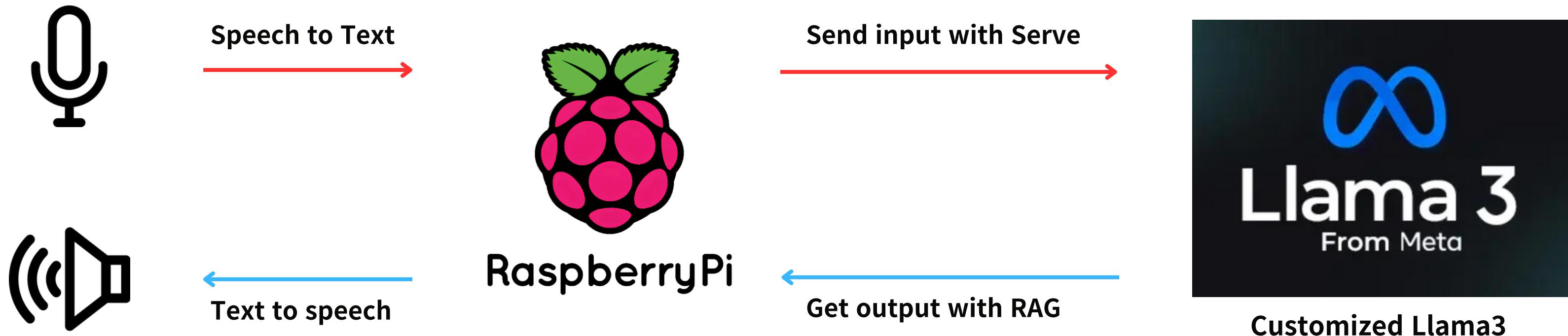
## 03 System architecture

### 2. Use Llama3



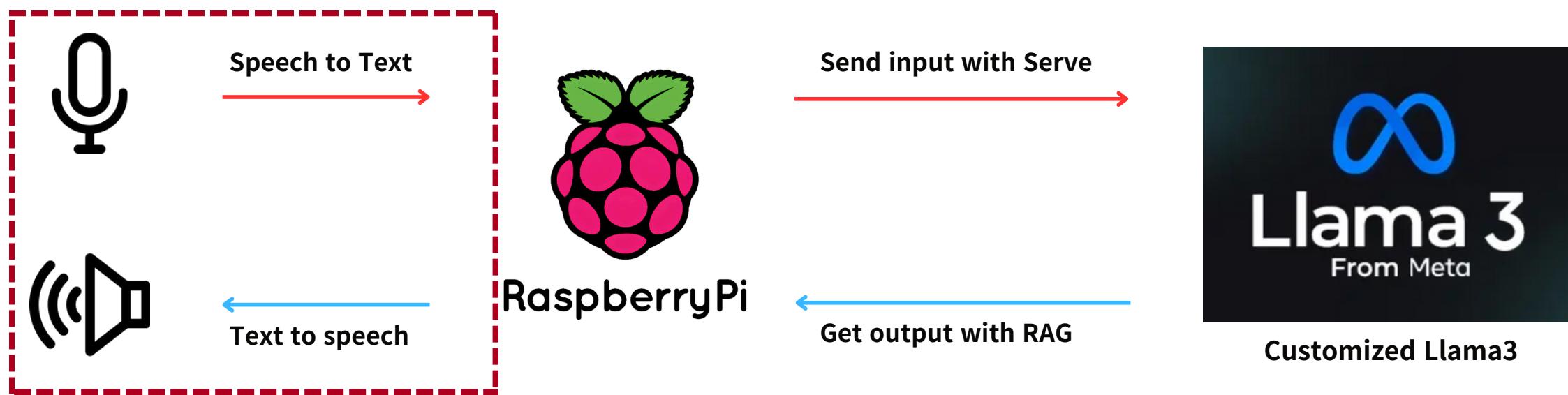
- We used Ollama, LangServe, Ngrok and Ollama to test the system that puts the LLM on the server.
- The LLM used is Llama3, fine-tuned in Korean.

### 3. Using Llama3 with RAG and PEFT



- Using information from the Department of Computer Software at Dongseoul University, PEFT and RAG is implemented
- The LLM used is Llama3, customized for Dongseoul University.

## 03 System architecture: Use Llama3 with RAG and PEFT



- Speech to text

```
with microphone as source:  
    recognizer.adjust_for_ambient_noise(source)  
    print("Say something!")  
    audio = recognizer.listen(source)  
  
response["transcription"] = recognizer.recognize_google(audio, language='ko-KR')
```

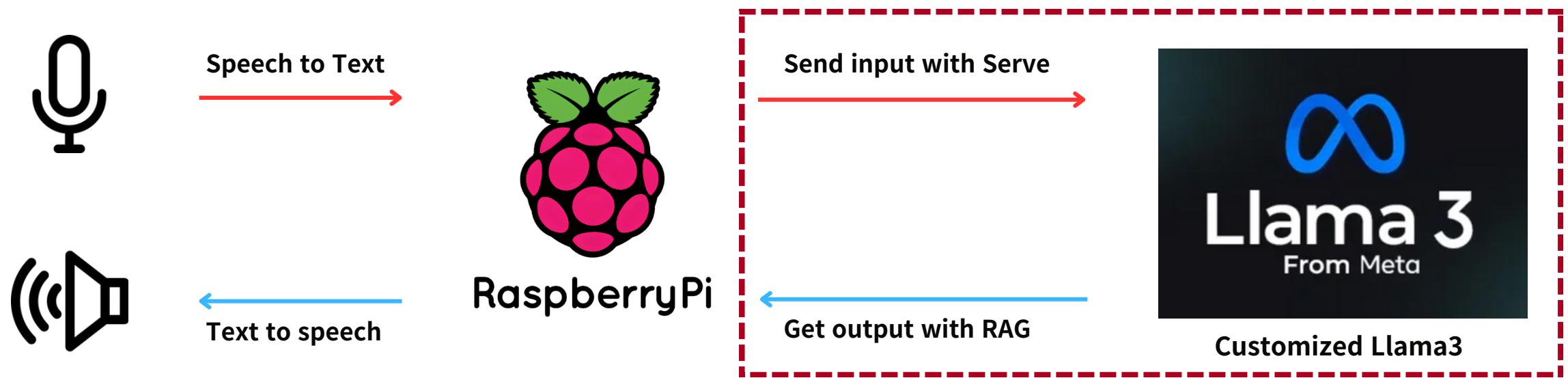
- Takes voice input and converts it into text in Korean.

- Text to speech

```
def speak_text(text):  
    """Convert text to speech and play it through speakers."""  
    engine = pyttsx3.init()  
    engine.say(text)  
    engine.runAndWait()
```

- Output the response of the LLM to the speaker.

## 03 System architecture: Use Llama3 with RAG and PEFT



- Importing customized LLM on the server

```
llm = RemoteRunnable("https://cute-easily-phoenix.ngrok-free.app/LLm/")
```

- Create Vector Store and RAG Chain for response with PDF File

```
pdf_path = "/home/Team98/audio/ComputerSoftware.pdf" # PDF 문서 디렉토리  
vector_store = create_vector_store(pdf_path)  
  
# RAG 체인 생성  
rag_chain = create_rag_chain(vector_store)
```

- Create a Vector Store by importing PDF files and create a RAG chain

- Input query and get response with RAG Chain by voice

```
result = recognize_speech_from_mic(recognizer, mic)  
if result["transcription"]:  
    print("You said: {}".format(result["transcription"]))  
    try:  
        rag_response = rag_chain.invoke(result["transcription"])  
        print("RAG response: {}".format(rag_response))  
        speak_text(rag_response)
```

- Use a RAG chain to generate a output

# 03 System architecture: Real implementation environment

```
(base) C:\Users\aa>conda activate llama_test

(llama_test) C:\Users\aa>ollama list
          NAME           ID      SIZE  MODIFIED
Llama3_dong_8b_q5:latest    abc8abf45346   5.7 GB  7 days ago
llama3.1:latest            62757c860e01   4.7 GB  10 days ago
Llama3_dong_200:latest     cd85ed765f4d   5.7 GB  10 days ago
Llama3_ko_8b_q5_ksjj:latest 9567109c5320   5.7 GB  4 weeks ago
ggml-model-Q5_K_M.gguf:latest eab0855063e8   7.7 GB  4 weeks ago
Llama3_ko_8b_q5:latest     eab0855063e8   7.7 GB  4 weeks ago
Llama3_unslot_08:latest     db36f1a3a9f9   8.5 GB  4 weeks ago
solar-instruct_q4:latest     3878fc8a09fb   6.5 GB  5 weeks ago
llama3:latest                365c0bd3c000   4.7 GB  5 weeks ago

(llama_test) C:\Users\aa>
```

- ollama

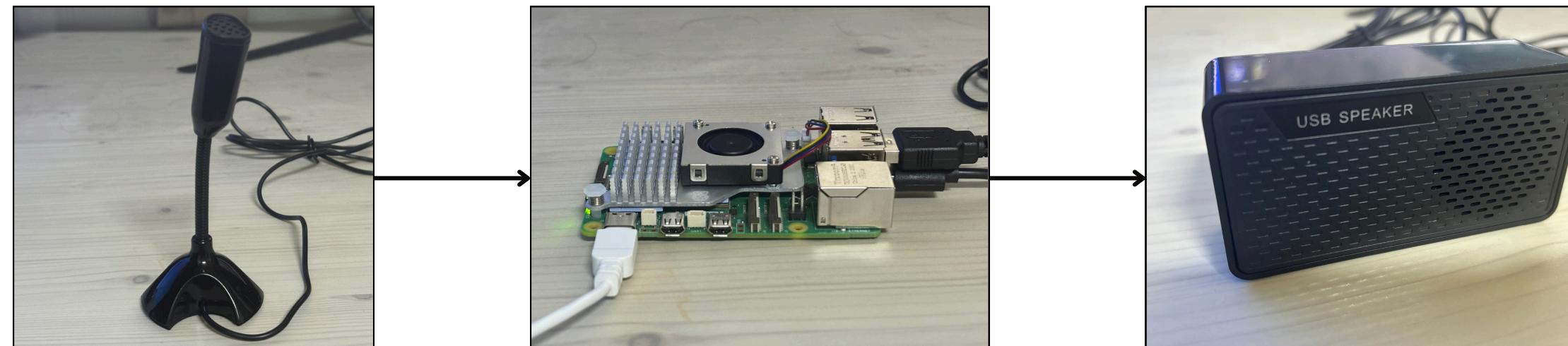
- Langserve

```
Anaconda Prompt - .\ngrok http --domain awake-werewolf-adjusted.ngrok-free.app 8000
ngrok
Help shape K8s Bindings https://ngrok.com/new-features-update?ref=k8s

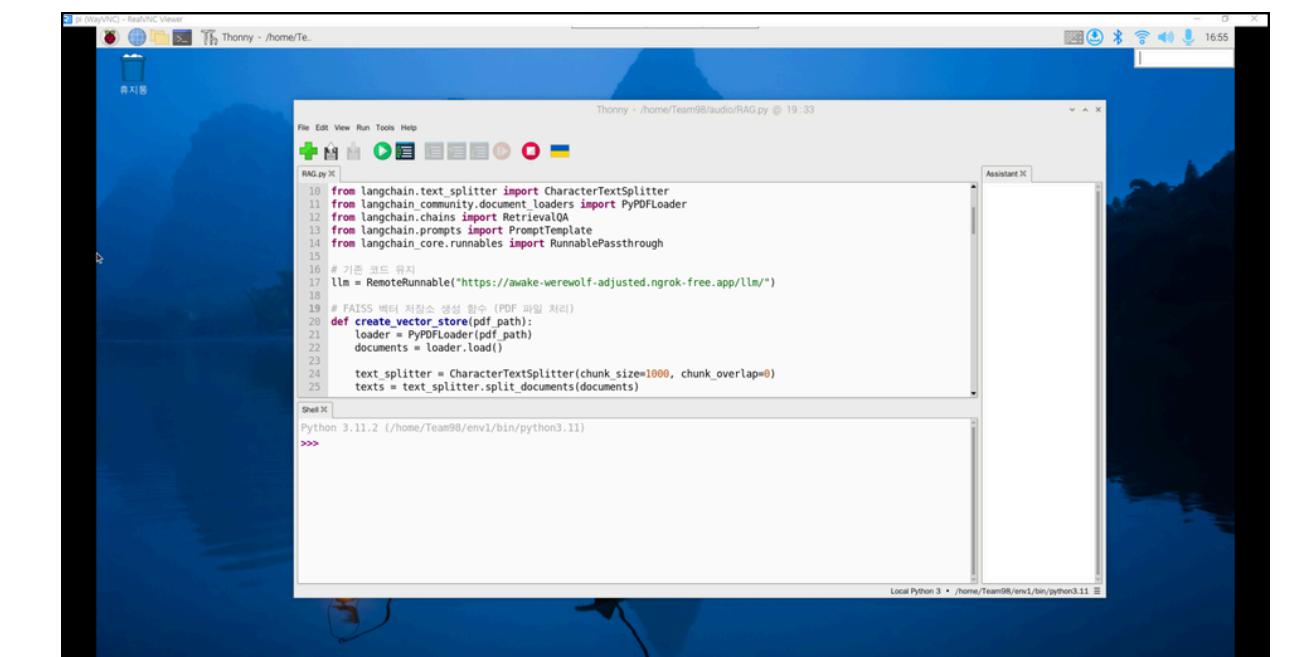
Session Status          online
Account                 d1acodn789@gmail.com (Plan: Free)
Version                3.14.0
Region                 Japan (jp)
Latency                36ms
Web Interface          http://127.0.0.1:4040
Forwarding             https://awake-werewolf-adjusted.ngrok-free.app -> http://localhost:8000

Connections            ttl     opn     rt1     rt5     p50     p90
                        0       0      0.00    0.00    0.00    0.00
```

- Ngrok



- Microphone, speaker and Raspberry Pi 5



- Code Completion Part

## 04 Performance Comparison: Dataset & methods

---

- To evaluate the implemented system, we compared three methods
  1. Using RAG with a general LLM,
  2. Using an LLM with PEFT applied
  3. Combining both PEFT and RAG in a single approach.
- The evaluation utilized a total of 18 datasets  
-> 6 sourced from the PDF provided by the RAG method, 6 from those used during the PEFT, and 6 entirely new datasets.
- Performance was evaluated by asking 5 questions for each query, for a total of 90 questions.

## 04 Performance Comparison: Dataset & methods

---

- Dataset obtained from PEFT

동서울대학교 컴퓨터소프트웨어과의 한 학년 반 수는?

동서울대학교에 가기 위한 방법은 무엇이 있나요?

동서울대학교 컴퓨터소프트웨어과의 특강에는 무엇이 있나요?

- Dataset obtained from RAG pdf

동서울대학교 컴퓨터소프트웨어과의 졸업생 취업 현황은 어떻게 되나요?

동서울대학교 컴퓨터소프트웨어과의 전공동아리에는 무엇이 있나요?

이재희 교수님의 연구실은 어디인가요?

- Dataset obtained from entirely new

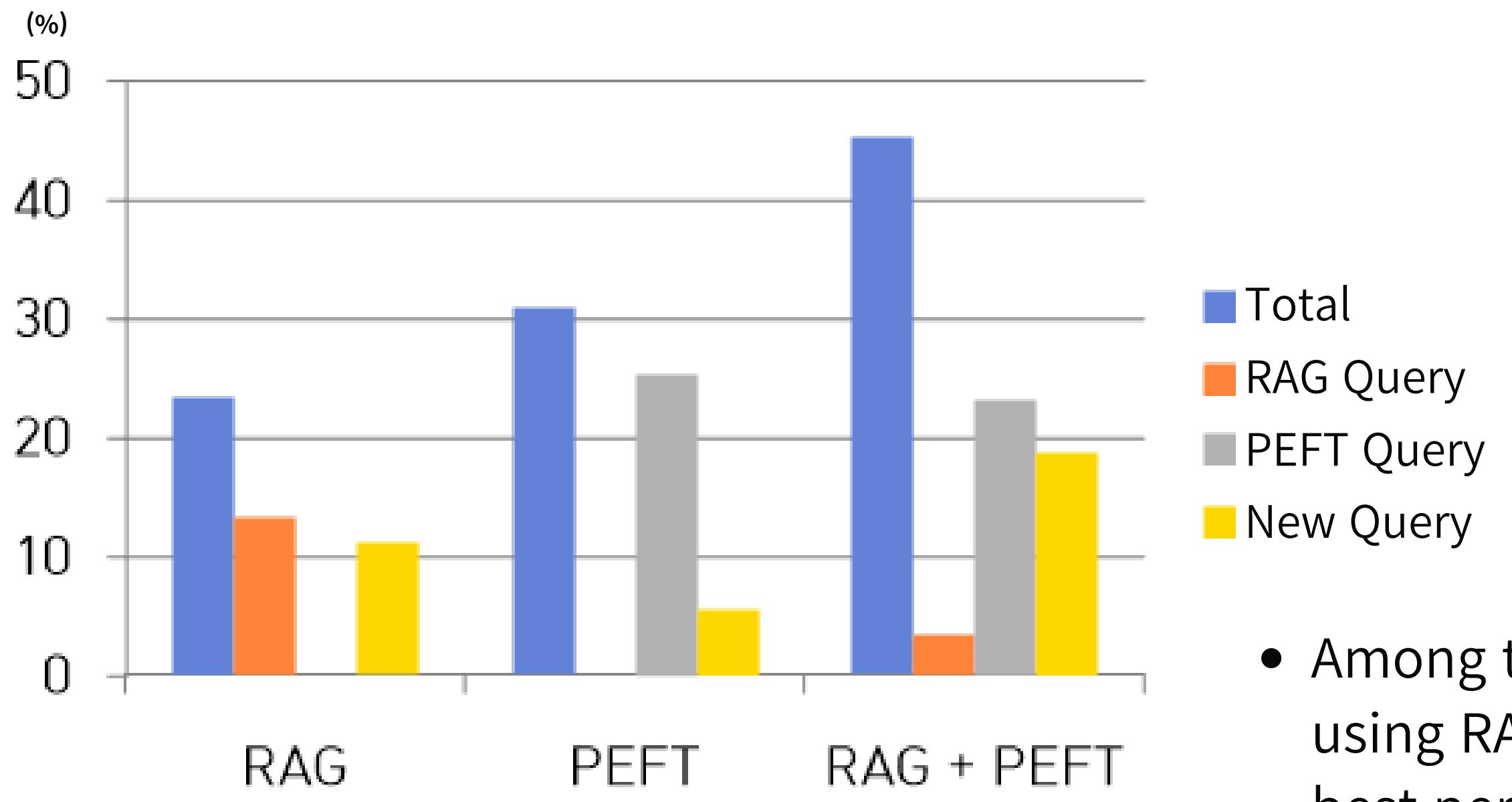
동서울대학교 컴퓨터소프트웨어과의 학과사무실 번호와 팩스 번호는 무엇인가요?

동서울대학교 컴퓨터소프트웨어과의 장학금은 얼마인가요?

동서울대학교 컴퓨터소프트웨어과의 학과사무실은 어디인가요?

## 04 Performance Comparison: Result

	RAG	PEFT	RAG+PEFT
Accuracy	23.3%	30.8%	45.1%



- Among the three methods, the method using RAG and PEFT together showed the best performance.

## 04 Performance Comparison: Analysis result

- RAG Query

	동서울대학교 컴퓨터소프트웨어과의 전공동아리에는 무엇이 있나요?
RAG	안녕하세요! 동서울대학교 컴퓨터 소프트웨어 학과의 전공 동아리들에 대해 궁금해하시는군요. 해당 학과는 SWAP, Logic, JTP, SM등 다양한 전공 동아리를 운영하고 있습니다.
PEFT	컴퓨터 소프트웨어과에 IT동아리가 있는데 주로 인공지능, 백엔드, 프론트엔드 등 각 분야에 담당자들이 매년 2월에 업부별 담당자들을 선정합니다.
RAG + PEFT	Logic, JIP, SM 등이 있으며 각각 논리 및 알고리즘, IT 분야의 창업 아이템을 개발하고 연구하는 동아리입니다.

- PEFT Query

	동서울대학교에 가기 위한 방법은 무엇이 있나요?
RAG	1. 대중교통 이용하기: 동서울대학교는 서울과 인접해 있어 다양한 교통수단을 이용하여 쉽게 접근 할 수 있습니다. a. 지하철: 동서울대학교는 2호선 상봉역에 위치하고 있습니다. b. 버스: 동서울대학교는 50번, 100번, 150번, 200번, 250번, 300번, 350번, 400번, 450번, 500번, 550번, 600번, 650번, 700번, 750번, 800번, 850번, 900번, 950번, 1000번, 1050번, 1100번, 1150번, 1200번, 1250번, 1300번, 1350번, 1400번, 1450번, 1500번, 1550번, 1600번, 1650번, 1700번, 1750번, 1800번, 1850번, 1900번, 1950번, 2000번, 2050번, 2100번, 2150번, 2200번, 2250번, 2300번, 2350번, 2400번, 2450번, 2500번, 2550번, 2600번, 2650번, 2700번, 2750번, 2800번, 2850번, 2900번, 2950번, 3000번, 3050번, 3100번, 3150번, 3200번, 3250번, 3300번, 3350번, 3400번, 3450번, 3500번, 3550번, 3600번, 3650번, 3700번, 3750번, 3800번, 3850번, 3900번, 3950번, 4000번, 4050번, 4100번, 4150번, 4200번, 4250번, 4300번, 4350번, 4400번, 4450번, 4500번, 4550번, 4600번, 4650번, 4700번, 4750번, 4800번, 4850번, 4900번, 4950번, 5000번, 5050번, 5100번, 5150번, 5200번, 5250번, 5300번, 5350번, 5400번, 5450번, 5500번, 5550번, 5600번, 5650번, 5700번, 5750번, 5800번, 5850번, 5900번, 5950번, 6000번, 6050번, 6100번, 6150번, 6200번, 6250번, 6300번, 6350번, 6400번, 6450번, 6500번, 6550번, 6600번, 6650번, 6700번, 6750번, 6800번, 6850번, 6900번, 6950번, 7000번, 7050번, 7100번, 7150번, 7200번, 7250번, 7300번, 7350번, 7400번, 7450번, 7500번, 7550번, 7600번, 7650번, 7700번, 7750번, 7800번, 7850번, 7900번, 7950번, 8000번, 8050번, 8100번, 8150번, 8200번, 8250번, 8300번, 8350번, 8400번, 8450번, 8500번, 8550번, 8600번, 8650번, 8700번, 8750번, 8800번, 8850번, 8900번, 8950번, 9000번, 9050번, 9100번, 9150번, 9200번, 9250번, 9300번, 9350번, 9400번, 9450번, 9500번, 9550번, 9600번, 9650번, 9700번, 9750번, 9800번, 9850번, 9900번, 9950번, 10000번, 10050번, 10100번, 10150번, 10200번, 10250번, 10300번, 10350번, 10400번, 10450번, 10500번, 10550번, 10600번, 10650번, 10700번, 10750번, 10800번, 10850번, 10900번, 10950번, 11000번, 11050번, 11100번, 11150번, 11200번, 11250번, 11300번, 11350번, 11400번, 11450번, 11500번, 11550번, 11600번, 11650번, 11700번, 11750번, 11800번, 11850번, 11900번, 11950번, 12000번, 12050번, 12100번, 12150번, 12200번, 12250번, 12300번, 12350번, 12400번, 12450번, 12500번, 12550번, 12600번, 12650번, 12700번, 12750번, 12800번, 12850번, 12900번, 12950번, 13000번, 13050번, 13100번, 13150번, 13200번, 13250번, 13300번, 13350번, 13400번, 13450번, 13500번, 13550번, 13600번, 13650번, 13700번, 13750번, 13800번, 13850번, 13900번, 13950번, 14000번, 14050번, 14100번, 14150번, 14200번, 14250번, 14300번, 14350번, 14400번, 14450번, 14500번, 14550번, 14600번, 14650번, 14700번, 14750번, 14800번, 14850번, 14900번, 14950번, 15000번, 15050번, 15100번, 15150번, 15200번, 15250번, 15300번, 15350번, 15400번, 15450번, 15500번, 15550번, 15600번, 15650번, 15700번, 15750번, 15800번, 15850번, 15900번, 15950번, 16000번, 16050번, 16100번, 16150번, 16200번, 16250번, 16300번, 16350번, 16400번, 16450번, 16500번, 16550번, 16600번, 16650번, 16700번, 16750번, 16800번, 16850번, 16900번, 16950번, 17000번, 17050번, 17100번, 17150번, 17200번, 17250번, 17300번, 17350번, 17400번, 17450번, 17500번, 17550번, 17600번, 17650번, 17700번, 17750번, 17800번, 17850번, 17900번, 17950번, 18000번, 18050번, 18100번, 18150번, 18200번, 18250번, 18300번, 18350번, 18400번, 18450번, 18500번, 18550번, 18600번, 18650번, 18700번, 18750번, 18800번, 18850번, 18900번, 18950번, 19000번, 19050번, 19100번, 19150번, 19200번, 19250번, 19300번, 19350번, 19400번, 19450번, 19500번, 19550번, 19600번, 19650번, 19700번, 19750번, 19800번, 19850번, 19900번, 19950번, 20000번, 20050번, 20100번, 20150번, 20200번, 20250번, 20300번, 20350번, 20400번, 20450번, 20500번, 20550번, 20600번, 20650번, 20700번, 20750번, 20800번, 20850번, 20900번, 20950번, 21000번, 21050번, 21100번, 21150번, 21200번, 21250번, 21300번, 21350번, 21400번, 21450번, 21500번, 21550번, 21600번, 21650번, 21700번, 21750번, 21800번, 21850번, 21900번, 21950번, 22000번, 22050번, 22100번, 22150번, 22200번, 22250번, 22300번, 22350번, 22400번, 22450번, 22500번, 22550번, 22600번, 22650번, 22700번, 22750번, 22800번, 22850번, 22900번, 22950번, 23000번, 23050번, 23100번, 23150번, 23200번, 23250번, 23300번, 23350번, 23400번, 23450번, 23500번, 23550번, 23600번, 23650번, 23700번, 23750번, 23800번, 23850번, 23900번, 23950번, 24000번, 24050번, 24100번, 24150번, 24200번, 24250번, 24300번, 24350번, 24400번, 24450번, 24500번, 24550번, 24600번, 24650번, 24700번, 24750번, 24800번, 24850번, 24900번, 24950번, 25000번, 25050번, 25100번, 25150번, 25200번, 25250번, 25300번, 25350번, 25400번, 25450번, 25500번, 25550번, 25600번, 25650번, 25700번, 25750번, 25800번, 25850번, 25900번, 25950번, 26000번, 26050번, 26100번, 26150번, 26200번, 26250번, 26300번, 26350번, 26400번, 26450번, 26500번, 26550번, 26600번, 26650번, 26700번, 26750번, 26800번, 26850번, 26900번, 26950번, 27000번, 27050번, 27100번, 27150번, 27200번, 27250번, 27300번, 27350번, 27400번, 27450번, 27500번, 27550번, 27600번, 27650번, 27700번, 27750번, 27800번, 27850번, 27900번, 27950번, 28000번, 28050번, 28100번, 28150번, 28200번, 28250번, 28300번, 28350번, 28400번, 28450번, 28500번, 28550번, 28600번, 28650번, 28700번, 28750번, 28800번, 28850번, 28900번, 28950번, 29000번, 29050번, 29100번, 29150번, 29200번, 29250번, 29300번, 29350번, 29400번, 29450번, 29500번, 29550번, 29600번, 29650번, 29700번, 29750번, 29800번, 29850번, 29900번, 29950번, 30000번, 30050번, 30100번, 30150번, 30200번, 30250번, 30300번, 30350번, 30400번, 30450번, 30500번, 30550번, 30600번, 30650번, 30700번, 30750번, 30800번, 30850번, 30900번, 30950번, 31000번, 31050번, 31100번, 31150번, 31200번, 31250번, 31300번, 31350번, 31400번, 31450번, 31500번, 31550번, 31600번, 31650번, 31700번, 31750번, 31800번, 31850번, 31900번, 31950번, 32000번, 32050번, 32100번, 32150번, 32200번, 32250번, 32300번, 32350번, 32400번, 32450번, 32500번, 32550번, 32600번, 32650번, 32700번, 32750번, 32800번, 32850번, 32900번, 32950번, 33000번, 33050번, 33100번, 33150번, 33200번, 33250번, 33300번, 33350번, 33400번, 33450번, 33500번, 33550번, 33600번, 33650번, 33700번, 33750번, 33800번, 33850번, 33900번, 33950번, 34000번, 34050번, 34100번, 34150번, 34200번, 34250번, 34300번, 34350번, 34400번, 34450번, 34500번, 34550번, 34600번, 34650번, 34700번, 34750번, 34800번, 34850번, 34900번, 34950번, 35000번, 35050번, 35100번, 35150번, 35200번, 35250번, 35300번, 35350번, 35400번, 35450번, 35500번, 35550번, 35600번, 35650번, 35700번, 35750번, 35800번, 35850번, 35900번, 35950번, 36000번, 36050번, 36100번, 36150번, 36200번, 36250번, 36300번, 36350번, 36400번, 36450번, 36500번, 36550번, 36600번, 36650번, 36700번, 36750번, 36800번, 36850번, 36900번, 36950번, 37000번, 37050번, 37100번, 37150번, 37200번, 37250번, 37300번, 37350번, 37400번, 37450번, 37500번, 37550번, 37600번, 37650번, 37700번, 37750번, 37800번, 37850번, 37900번, 37950번, 38000번, 38050번, 38100번, 38150번, 38200번, 38250번, 38300번, 38350번, 38400번, 38450번, 38500번, 38550번, 38600번, 38650번, 38700번, 38750번, 38800번, 38850번, 38900번, 38950번, 39000번, 39050번, 39100번, 39150번, 39200번, 39250번, 39300번, 39350번, 39400번, 39450번, 39500번, 39550번, 39600번, 39650번, 39700번, 39750번, 39800번, 39850번, 39900번, 39950번, 40000번, 40050번, 40100번, 40150번, 40200번, 40250번, 40300번, 40350번, 40400번, 40450번, 40500번, 40550번, 40600번, 40650번, 40700번, 40750번, 40800번, 40850번, 40900번, 40950번, 41000번, 41050번, 41100번, 41150번, 41200번, 41250번, 41300번, 41350번, 41400번, 41450번, 41500번, 41550번, 41600번, 41650번, 41700번, 41750번, 41800번, 41850번, 41900번, 41950번, 42000번, 42050번, 42100번, 42150번, 42200번, 42250번, 42300번, 42350번, 42400번, 42450번, 42500번, 42550번, 42600번, 42650번, 42700번, 42750번, 42800번, 42850번, 42900번, 42950번, 43000번, 43050번, 43100번, 43150번, 43200번, 43250번, 43300번, 43350번, 43400번, 43450번, 43500번, 43550번, 43600번, 43650번, 43700번, 43750번, 43800번, 43850번, 43900번, 43950번, 44000번, 44050번, 44100번, 44150번, 44200번, 44250번, 44300번, 44

## 04 Performance Comparison: Conclusion

---

- Using RAG and PEFT methods, we were able to create voice recognition chatbots that could deliver more accurate information in a specific field to users
- The performance of the RAG system was poor compared to other systems.
- To solve this problem, we plan to study in the future to improve the performance of our system using an optimization system such as AutoRAG.

# Q / A