

LLM is all you need

LLM의 모든 것

Contents

01

소개

Team

02

기획

Research & Services

03

시스템 아키텍처

Front end & Back end

04

분석 및 평가

Analysis and Evaluation

05

결론

Conclusion

06

QnA

궁금한 것이 있다면
물어보세요.

01

소개

Team

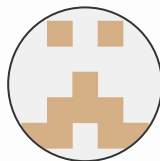
소개



신동원

- Backend
- Technology Director
- AI

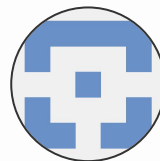
- ✓ 로컬 및 클라우드 MLOps
- ✓ LangChain + Model 빌드
- ✓ Backend 기능 구현 및 단위 테스트
- ✓ 통합 테스트



전진환

- Frontend
- Project Manager
- AI

- ✓ 총괄 기획
- ✓ Frontend 기능 구현 및 단위 테스트
- ✓ UI / UX 제작



전대엽

- UI UX design
- AI

- ✓ UI 제작
- ✓ Prompt Engineering
- ✓ 평가지표 제작

공통

- ✓ 코드리뷰 및 논문리뷰
- ✓ 결과 분석
- ✓ 함수 기능 및 성능 테스트
- ✓ 자료조사 및 정리

02


기획

Research & Services

Goal : Gemini / PaLM2를 활용한 RAG + In-Context Learning 실험 및 적용

- How to tune for LLM?
 - LLM을 학습하는 방법 찾기
- HuggingFace PEFT(Parameter Efficient Fine Tuning)
 - PEFT란?
- PEFT applicability range for PaLM2 and Gemini Pro
 - 프로젝트 기간동안 구현 가능한 범위 설정
- RAG(Retrieval Augmented Generation) + In-Context Learning
 - LangChain을 활용한 PDF 인식 및 Prompt Tuning
- Research & Services
 - 평가 지표 생성
 - 구현한 기술을 바탕으로 한 실험 및 서비스

Project Schedule



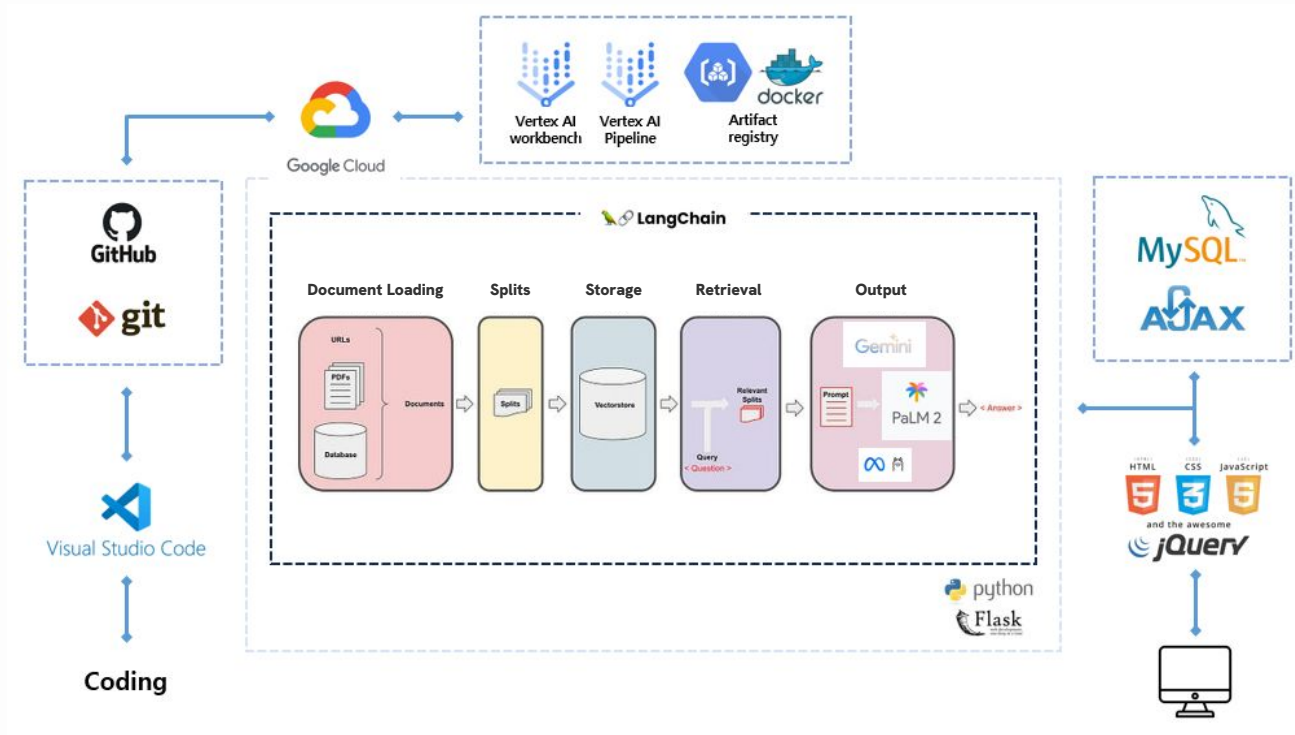
Task	Description	Date	J 15	J 17	J 26	F 2	F 5	F 8	Status
Local Docker setting	Local 환경 Docker container 설치 및 테스트 준비	Jan 15 (1일)							Completed
Prompt Tuning	모델 및 용도별 Prompt tuning	Jan 15 - Jan 17 (2일)							Completed
Frontend	HTML, CSS, bootstrap, Ajax를 활용한 UI/UX 및 기능 구현	Jan 15 - Jan 26 (12일)							Completed
LangChain	PaLM2 / Gemini / KoLlama 2 RAG 관련 세부 기능 구현 및 테스트	Jan 16 - Feb 2 (14일)							Completed
Backend	Flask + LangChain + docker 등을 활용한 기능 구현	Jan 16 - Feb 2 (14일)							Completed
Evaluation & Analysis & Clean up	평가지표 생성 & 분석 & 정리	Jan 31 - Feb 5 (6일)							Completed
Cloud Deploy	Docker Hub 배포 Google Cloud Platform 배포	Feb 2 - Feb 5 (3일)							Completed

03

시스템 아키텍처

Frontend & Backend

System Architecture



Tree

```
├── PDF_DN_FOLDER
├── __pycache__
├── static
│   ├── chat.js
│   └── style.css
├── templates
│   ├── admin.html
│   ├── diffusion.html
│   ├── gemini.html
│   ├── index.html
│   ├── law.html
│   ├── llama.html
│   └── museum.html
├── text_Similarity.py
├── .env
├── .gitignore
├── app_Gemini.py
├── app_Llama2.py
├── app_PaLM2.py
├── model_Gemini.py
├── model_Llama2.py
├── model_PaLM2.py
├── app.py
├── README.md
├── requirements.txt
├── llama-2-13b-chat.Q5_K_M.gguf
├── ver1_app.py
└── ver1_requirements.txt
```

PDF_DN_FOLDER

Pdf Upload Check

Static & templates

Frontend

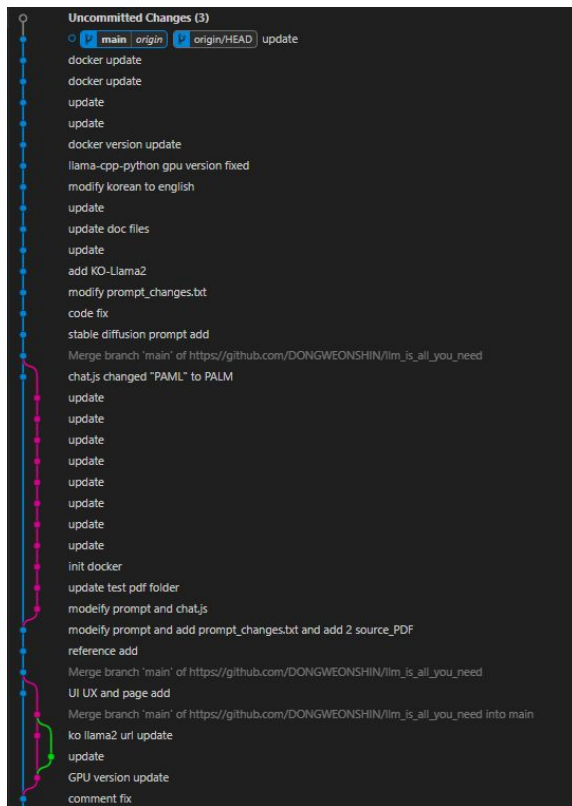
App.py & model.py

Backend

.env

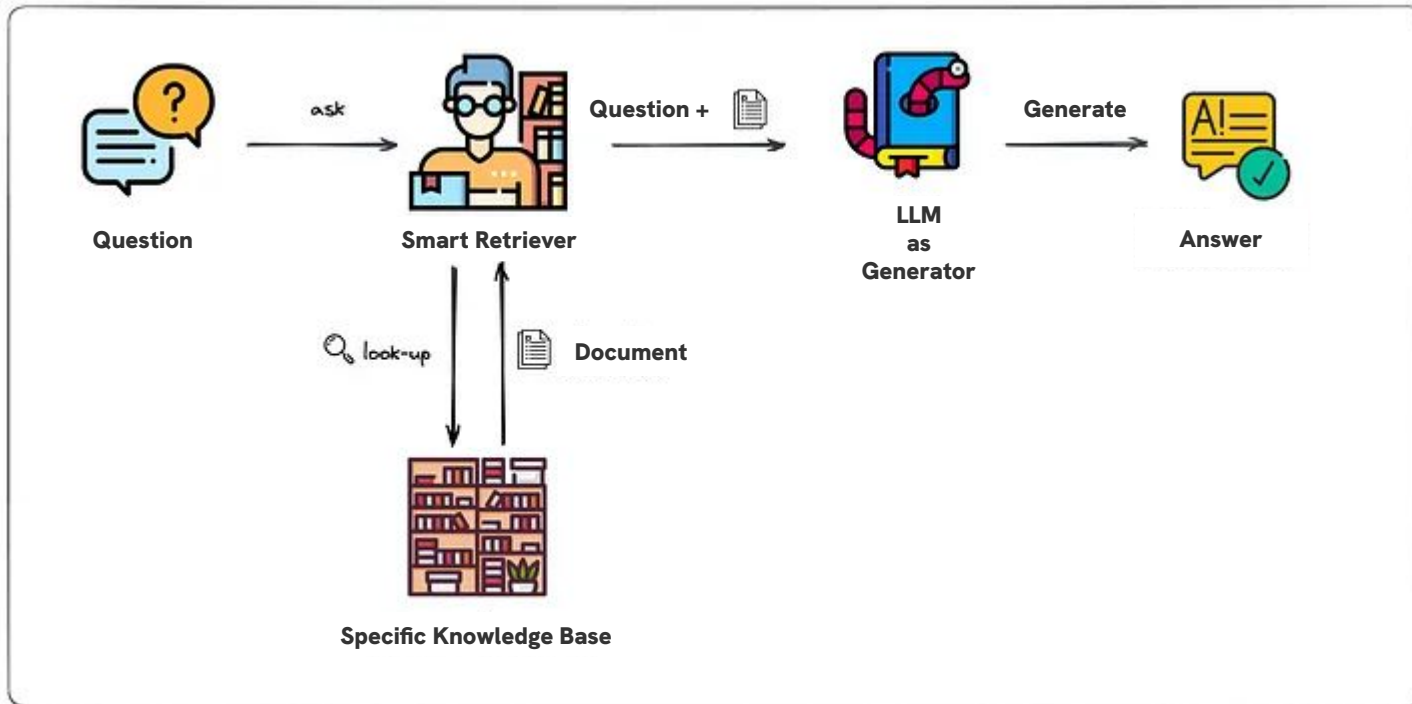
API key

Git Graph



RAG(Retrieval-Augmented Generation)

Backend



RAG

```

01
# Ingest PDF files
fileFullPath = os.path.join(PDF_DIR_FOLDER, fullFilename)
loader = PyPDFLoader(fileFullPath)
documents = loader.load_and_split()

```

```

04
# Test search
query = embeddings.embed_query(msg)
docs = db.similarity_search_by_vector(query)
print(docs[0].page_content)

# Retrieval
retriever = db.as_retriever()

```

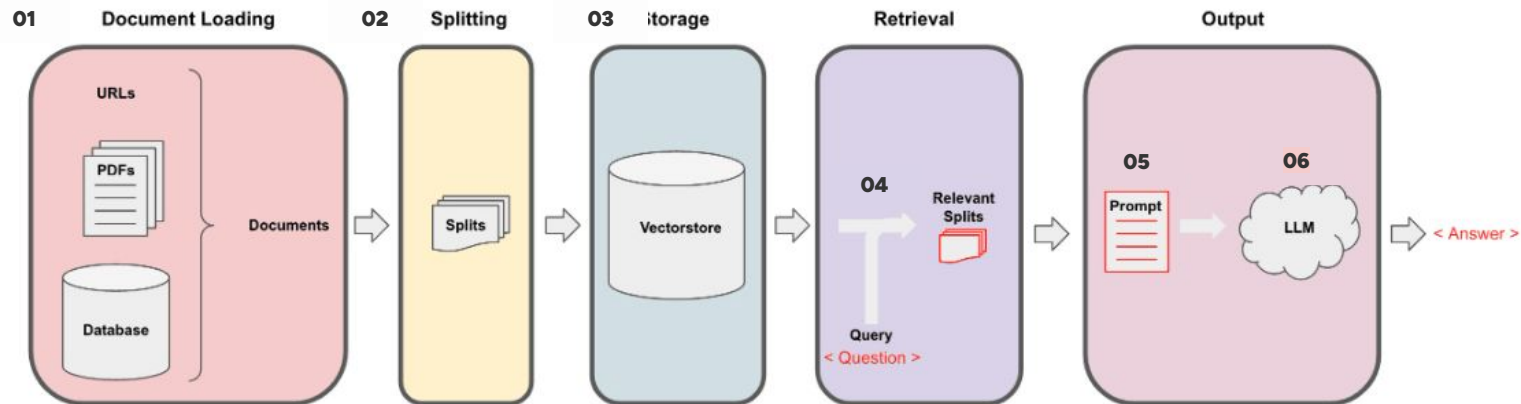
```

05
# Customize the default retrieval prompt template
template = """
{context}

Question: {question}
"""

prompt = ChatPromptTemplate.from_template(template)

```



```

02
# Chunk documents
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=1000,
    chunk_overlap=50,
    separators=["\n\n", "\n", ".", "!", "?", ",", " ", ""],
)
doc_splits = text_splitter.split_documents(documents)

```

```

03
# Embeddings
embeddings = VertexAIEmbeddings(model_name="textembedding-gecko@001")

# Vector Store Indexing
db = FAISS.from_documents(doc_splits, embeddings)

```

```

06
# Configure RetrievalQA chain
retrieval_chain = (
    {"context": retriever, "question": RunnablePassthrough()}
    | prompt
    | self.llm
    | StrOutputParser()
)

response = retrieval_chain.invoke(msg)

```

Backend

Issue

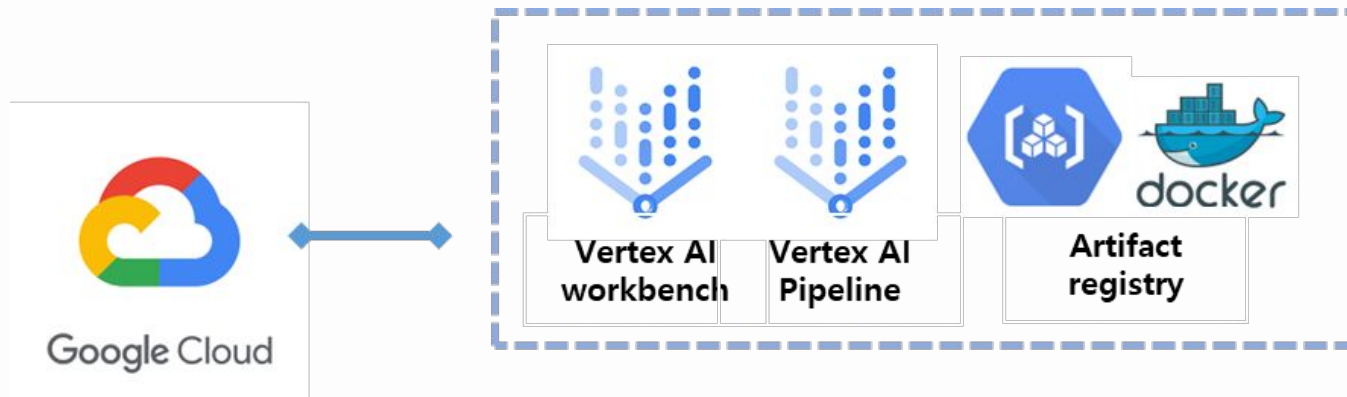
문제점

- 2023년 12월에 출시된 Gemini의 버전별 코드가 다르고 샘플코드가 X
 - Google AI Gemini
 - Vertex AI Gemini
- LangChain도 Gemini를 대응하기 위한 샘플코드가 많지 않음
- RAG 구현 후 첫번째 테스트에서 10번의 request 중 1번의 response만 받는 경우가 발생

해결

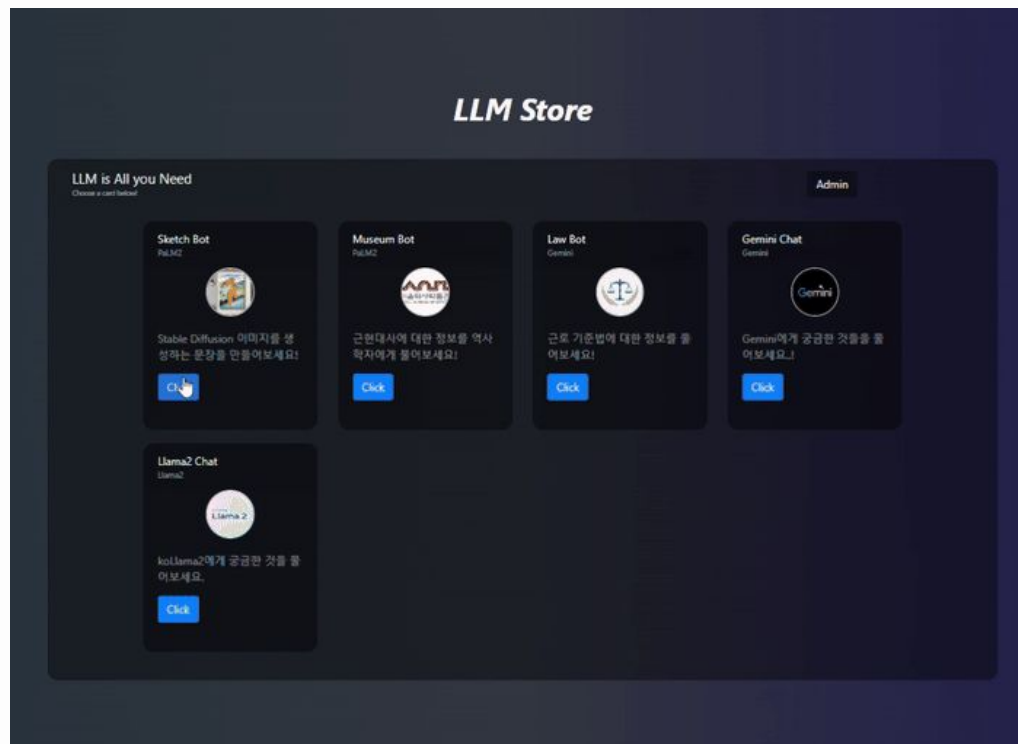
- Gemini 샘플코드 분류 후 분석
임베딩과 질의, 벡터 스토어 개념 재학습
LangChain 버전 안정화 확인하고
LangChain 구조를 변경하여 RAG 성공
- 두 버전 중 Google AI Gemini를 사용
- Vector store를 FAISS로 변경

Deploy



Prototype

시연 이미지 및 영상



04

분석 및 평가

Analysis and Evaluation

Issue

Use Cases

Get comparable performance to full finetuning by adapting LLMs to downstream tasks using consumer hardware

GPU memory required for adapting LLMs on the few-shot dataset [ought/raft/twitter_complaints](#). Here, settings considered are full finetuning, PEFT-LoRA using plain PyTorch and PEFT-LoRA using DeepSpeed with CPU Offloading.

Hardware: Single A100 80GB GPU with CPU RAM above 64GB

Model	Full Finetuning	PEFT-LoRA PyTorch	PEFT-LoRA DeepSpeed with CPU Offloading
bigscience/T0_3B (3B params)	47.14GB GPU / 2.96GB CPU	14.4GB GPU / 2.96GB CPU	9.8GB GPU / 17.8GB CPU
bigscience/mt0-xxl (12B params)	OOM GPU	56GB GPU / 3GB CPU	22GB GPU / 52GB CPU
bigscience/bloomz-7b1 (7B params)	OOM GPU	32GB GPU / 3.8GB CPU	18.1GB GPU / 35GB CPU

Vram이 너무 크다!

GCP로 학습을 하려는데 하드웨어 자원이 너무 비싸다!

1개월 내에 여러개의 논문을 구현 할 시간이 부족하다!

PEFT vs In-Context Learning

In-Context Learning

- 학습 없음
- New task 수행 가능
- Mixed-task batches

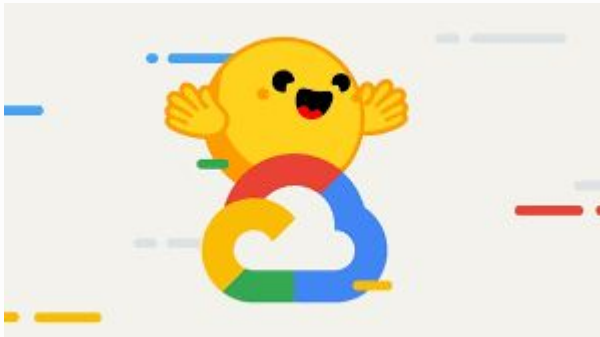


Parameter-efficient Fine-tuning

- 학습 시 적은 파라미터 사용
- New task에 대해 높은 성능
- Mixed-task batches 가능

PEFT

Parameter Efficient Fine-Tuning



State-of-the-art Parameter-Efficient Fine-Tuning (PEFT) methods

Parameter-Efficient Fine-Tuning (PEFT) methods enable efficient adaptation of pre-trained language models (PLMs) to various downstream applications without fine-tuning all the model's parameters. Fine-tuning large-scale PLMs is often prohibitively costly. In this regard, PEFT methods only fine-tune a small number of (extra) model parameters, thereby greatly decreasing the computational and storage costs. Recent State-of-the-Art PEFT techniques achieve performance comparable to that of full fine-tuning.

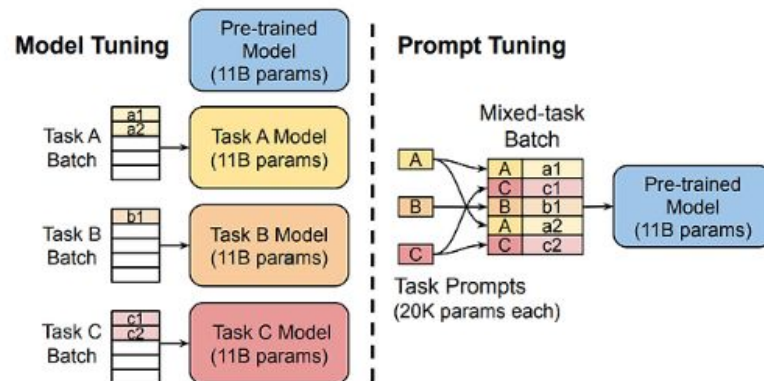
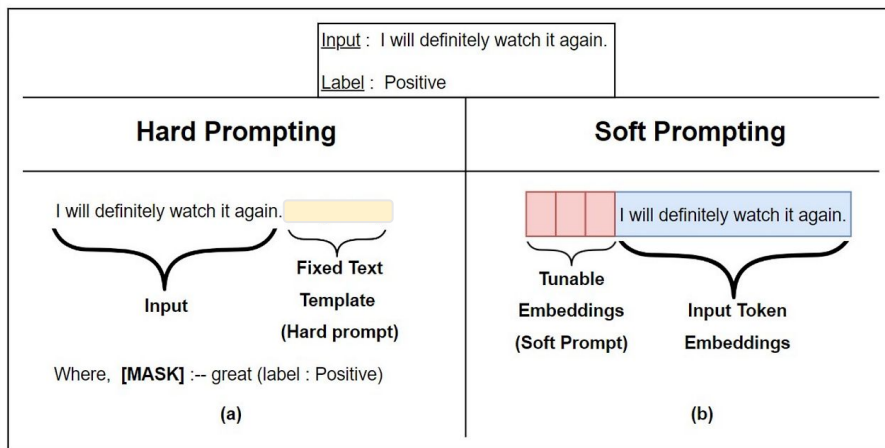
Seamlessly integrated with 🤖 Accelerate for large scale models leveraging DeepSpeed and Big Model Inference.

Supported methods:

1. LoRA: [LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS](#)
2. Prefix Tuning: [Prefix-Tuning: Optimizing Continuous Prompts for Generation](#), [P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks](#)
3. P-Tuning: [GPT Understands Too](#)
4. Prompt Tuning: [The Power of Scale for Parameter-Efficient Prompt Tuning](#)

Prompt Tuning

Hard prompt vs Soft prompt

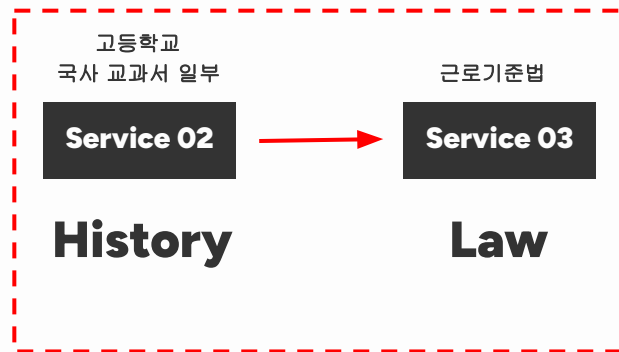


(A) Model Tuning vs. Prompt Tuning

Prompt Tuning Test

공통

1. 모델(PaLM2, Gemini, GPT4, Kollama 2) 기준
2. 각 모델에 PDF 내용에 대한 질문 30문항씩
3. 한 질문 당 5점 만점 기준으로 각 문항별 점수 산출 후 총점 합산 후 정답률 산출



PDF의 질문(Answer)에 "n개의 keyword"가 있다면

chat model의 응답(Response)을 받을 때 몇 개의 keyword가 포함되어 있는가. ($0 \leq m \leq n$)

0점 : keyword가 전혀 포함되지 않음

중간점수 : $\frac{\text{response에 포함된 keyword 수}}{\text{총 keyword}} \times 5$

5점 : keyword가 모두 포함되어 있음

$$\text{항목 당 점수}(S_i) = \frac{m}{n} \times 5 \quad \text{총점}(S_t) = \sum S_i$$

$$\text{정답률} = \frac{\text{총점}}{(30\text{문항} \times 5)} * 100$$

Prompt Tuning Test

History / Law

순서

1. PaLM2 한글프롬프트 vs 영문프롬프트 (History)
2. PaLM2 영문프롬프트 vs 영문번역프롬프트 (History)
 - 2-1. PaLM2 vs Gemini (History)
3. PaLM2 (History vs Law)
4. PaLM2 vs Gemini vs Kollama2 (Law)

1	한글 프롬프트	영문 프롬프트
총점(score)	36.00(150.00)	48.17(150.00)
정답률(%)	24.0	32.1

8% 상승

2	영문 프롬프트	영문번역 프롬프트
총점(score)	48.17(150.00)	49.50(150.00)
정답률(%)	32.1	33.0

0.9% 상승

2-1	PaLM2	Gemini
총점(score)	49.50(150.00)	6.00(150.00)
정답률(%)	33.0	4.0

※ pdf 적합성 의심

3	History	Law
총점(score)	49.50(150.00)	119.50(150.00)
정답률(%)	33.0	79.7

46.7% 상승

4	PaLM2	Gemini	Kollama 2
총점(score)	119.50(150.00)	144.00(150.00)	57.83(150.00)
정답률(%)	79.7	96.0	38.6

약 16% 상승!

한글로만 작성된 프롬프트(한글 프롬프트)

한글을 영어로 단순 번역한 프롬프트 (영문 프롬프트)

프롬프트를 **step**별로 작성하고,
PDF와 질의를 영어로 인식하도록 영어로 프롬프트 작성 (영문 번역 프롬프트)

Prompt Tuning Test

Stable Diffusion

추가

1. 만든문장을 바탕으로 프롬프트를 생성할 때 영어 문장처럼 자연스럽게 나올 수 있도록 프롬프트
2. TF-IDF + cosine similarity

$$TFIDF(t, d, D) = TF(t, d) * IDF(t, D)$$

$$TF(t, d) = \frac{\text{문서 } d \text{에서 단어 } t \text{가 등장한 횟수}}{\text{문서 } d \text{에 등장한 모든단어수}}$$

$$df(t) = \text{특정 단어 } t \text{가 등장한 문서의 수}$$

$$IDF(t, D) = \log \frac{\text{총 문서의 개수}}{\text{단어 } t \text{를 포함하는 문서의 수}}$$



$$similarity = \cos(\Theta) = \frac{A \cdot B}{||A|| ||B||}$$

Prompt Tuning Test

Stable Diffusion prompt

```
template = """Answer the question based only on the following context:

I am a student in Seoul, South Korea. \
I want you to help me make requests (prompts) for the Stable Diffusion neural network. \
What I want from you is that when I ask a question, you will answer it slowly and according to the procedure. \
Additionally, if you answer well, we will use the tip to promote you to people around you and give you lots of praise. \
Please answer basically in Korean. \
If there is content that is not in the pdf, please reply, "I don't know. Please only ask questions about what is in the pdf." \

Be as specific as possible in your requests. Stable diffusion handles concrete prompts better than abstract or ambiguous ones. For example, instead of "portrait of a woman" it is better to write "portrait of a woman with brown eyes and red hair in Renaissance style". \
Specify specific art styles or materials. If you want to get an image in a certain style or with a certain texture, then specify this in your request. For example, instead of "landscape" it is better to write "watercolor landscape with mountains and lake". \
Specify specific artists for reference. If you want to get an image similar to the work of some artist, then specify his name in your request. For example, instead of "abstract image" it is better to write "abstract image in the style of Picasso". \
Weigh your keywords. You can use token:1.3 to specify the weight of keywords in your query. The greater the weight of the keyword, the more it will affect the result. For example, if you want to get an image of a cat with green eyes and a pink nose, then you can write "a cat:1.5, green eyes:1.3,pink nose:1". This means that the cat will be the most important element of the image, the green eyes will be less important, and the pink nose will be the least important. \
Another way to adjust the strength of a keyword is to use () and []. (keyword) increases the strength of the keyword by 1.1 times and is equivalent to (keyword:1.1). [keyword] reduces the strength of the keyword by 0.9 times and corresponds to (keyword:0.9). \
My query may be in other languages. In that case, translate it into English. Your answer is exclusively in English (IMPORTANT!!!), since the model only understands it. \
Also, you should not copy my request directly in your response, you should compose a new one, observing the format given in the examples. \
Don't add your comments, but answer right away. \

You can use several of them, as in algebra... The effect is multiplicative. \
(keyword): 1.1 \
((keyword)): 1.21 \
(((keyword))) : 1.33 \

Similarly, the effects of using multiple [] are as follows \
[keyword]: 0.9 \
[[keyword]]: 0.81 \
[[[keyword]]]: 0.73 \

This is negative prompting. \
paintings, sketches, (worst quality:2), (low quality:2), (normal quality:2), (normal, normal quality), (monochrome)), (grayscale), skin spots, \
acnes, skin blemishes, age spot, extra fingers, fewer fingers, strange fingers, bad hand, bad anatomy, fused fingers, missing leg, mutated hand, \
malformed limbs, missing feet, Closed eye, TXT, fewer digits, missing arms, bad hands, extra digit, morearms,(more hands), text, title, logo, \

1. 단절차단용 \
2. 알맞은 값으로 stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
3. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
4. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
5. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
6. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
7. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
8. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
9. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
10. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
11. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
12. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
13. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
14. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
15. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
16. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
17. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
18. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
19. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
20. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
21. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
22. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
23. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
24. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
25. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
26. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
27. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
28. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
29. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
30. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
31. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
32. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
33. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
34. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
35. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
36. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
37. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
38. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
39. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
40. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
41. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
42. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
43. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
44. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
45. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
46. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
47. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
48. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
49. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
50. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
51. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
52. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
53. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
54. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
55. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
56. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
57. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
58. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
59. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
60. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
61. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
62. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
63. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
64. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
65. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
66. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
67. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
68. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
69. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
70. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
71. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
72. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
73. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
74. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
75. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
76. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
77. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
78. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
79. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
80. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
81. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
82. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
83. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
84. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
85. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
86. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
87. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
88. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
89. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
90. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
91. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
92. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
93. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
94. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
95. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
96. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
97. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
98. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
99. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \
100. stable diffusion prompt을 만들어 달라고 부탁하는 예시들 \

Question:
Answer:
...

```

Stable diffusion 성능을 올리는 여러가지 방법

Zero-shot prompt

Few-shot prompt(3-shot)

Prompt Tuning Test

Stable Diffusion

입력 프롬프트

니콘카메라로 촬영한 (흑백)의 모나리자 배경 안에
엠마스톤이 정면을 바라보고 있습니다.

정답 프롬프트

Emma Stone is looking straight ahead in
the background of the Mona Lisa (black
and white) taken with a Nikon camera.

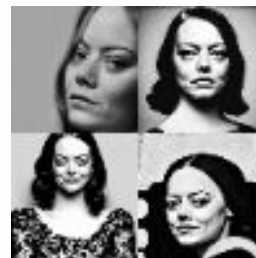
Model	생성 프롬프트	점수	유사도
PaLM2	Emma Stone looking straight ahead against a background of the Mona Lisa (in black and white) shot with a Nikon camera.	4.5	0.858
Gemini Pro	Emma Stone looking at the camera in front of the Mona Lisa background in black and white, taken with a Nikon camera.	5	0.829
GPT4	Emma Stone, shot with a Nikon camera, facing forward within a black and white background of the Mona Lisa.	4.8	0.566
Llama 2	비교 대상 제외		

Prompt Tuning Test

Stable Diffusion

Model	총점(score)	정답률(%)	문장 유사도
PaLM2	132.20 (150)	88.13	0.78
Gemini Pro	128.80	85.87	0.710
GPT4	120.30	80.20	0.527
Llama 2	비교 대상 제외	-	-

정답률이 상승함에 따른 문장 유사도 상승
양의 상관관계!



05

결론

Conclusion

Conclusion

- PaLM2 / Gemini / Llama2 RAG + Prompt Tuning 구현
 - Law / Stable Diffusion / Chat 모델 구현
 - GCP를 통한 서비스 배포
- RAG + Prompt Tuning을 통한 효율적인 Tuning
 - 비용절감
- 지표 생성을 바탕으로 한 객관적인 평가
 - Prompt 평가를 위한 공통 지표 (평균, 정답률)
 - TF-IDF + cosine similarity (문장 유사도)
 - 정답률과 문장 유사도간의 상관관계 확인
- Prompt Tuning을 통한 성능 향상
 - 한글과 영문프롬프트 실험 결과 **8%** 상승
 - 영문 번역프롬프트 실험 결과 **0.9%** 상승
 - Pdf 파일의 형식에 따른 성능 차이 확인. 이에 따른 프롬프팅 개선 및 향상 방법 발견
 - 프롬프팅을 통한 효율적인 성능 향상 확인 **16%** 상승

향후 계획



PEFT 추가 적용

- P-tuning
- Prefix tuning



Prompt Tuning

- Cognitive prompting
- CoT / ToT ...



다른 **Tuning** 방법 적용

- PPO(Proximal Policy Optimization)
- DPO(Direct Preference Optimization)
- RLHF(Reinforcement Learning Human Feedback)

06

QnA

궁금한 것이 있다면 물어보세요.

Sources and Results

Github / Notion

- [LLM is all you need \(Github\)](#)
- [LLM is all you need \(Notion\)](#)

Experiment result

- [History Evaluation](#)
- [Total Evaluation](#)