

# 캡스톤디자인

최종 발표 - 6/10/24 (월)

송효섭, 이정호, 최우성

# 목차

- 제품 개요 - 취업준비생을 위한 페인 포인트 도출
- 제품 MVP 도출 - 기술면접 답변 정량평가 모델 기반 웹 서비스
- 제작 과정 - 기능별 구현 및 오류 해결
  - Dataset Generation, Augmentation
  - Tree-of-thought prompt engineering
  - Classification Fine-tuning
  - Contrastive Learning
  - Chat Vector
- 웹서비스 데모 계획 및 구현
  - MVP 기능 API 명세서 작성
  - 프로젝트 아키텍처 및 적용 패턴
  - 클라우드 기반 배포 환경 구축
  - 개발 고도화 계획 및 회고

# 제품 개요 : Usay

## 생성형 AI 모의 면접 Chatbot

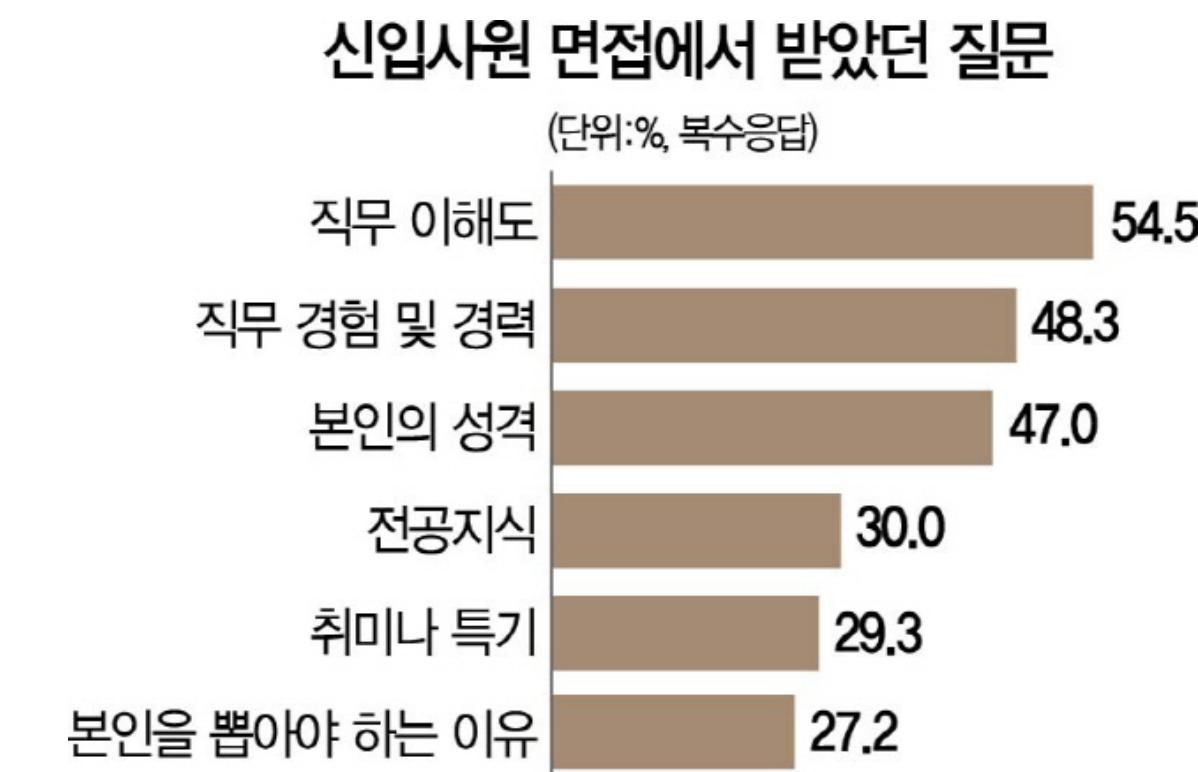
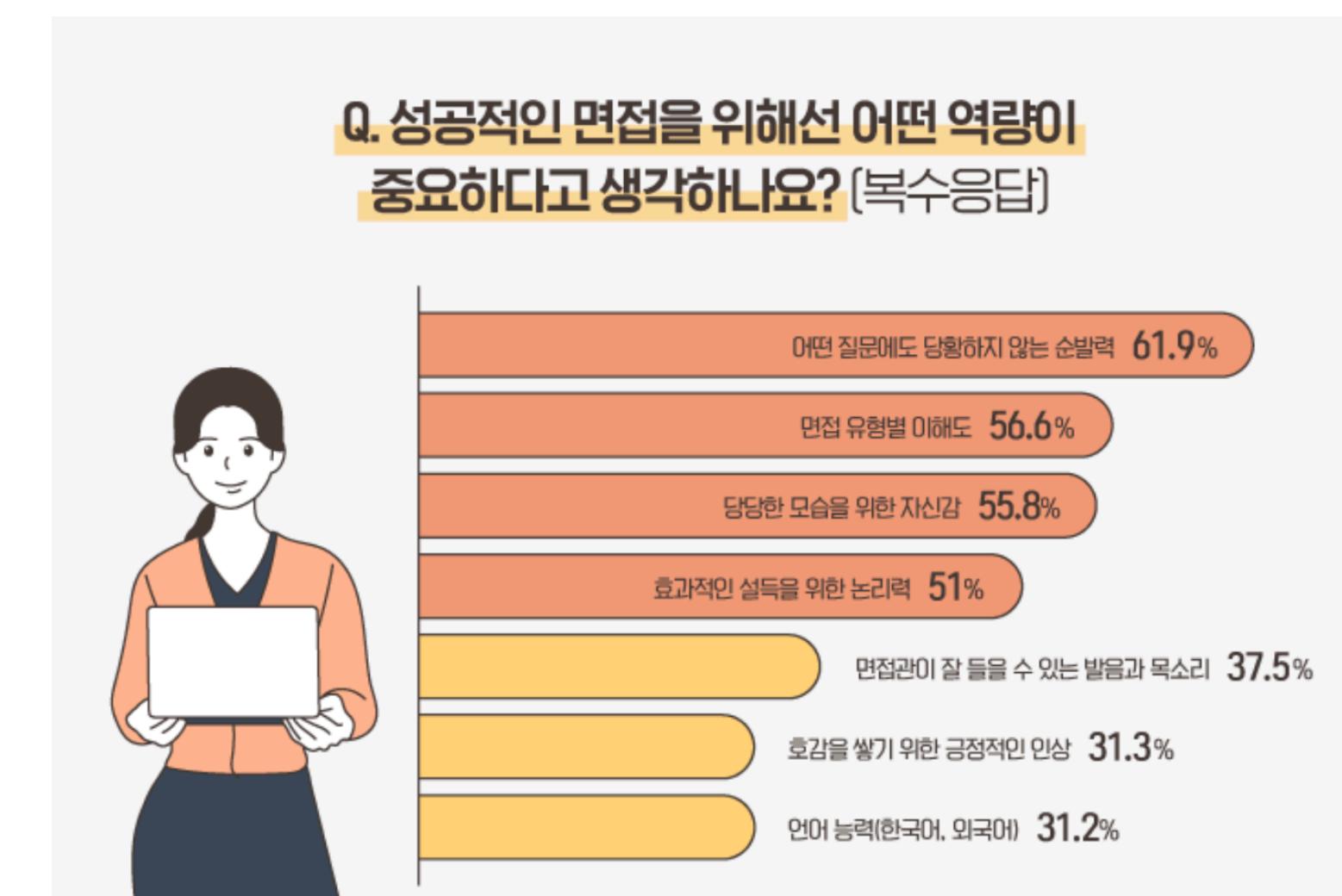
- CS 분야의 면접을 준비하고 있는 개발자 지망생들을 위한 생성형 AI 기반의 모의 면접 Chatbot.
- CS 분야의 정보들을 면접 상황에 맞게 설명할 수 있도록 사용자의 면접을 평가하고 개선해주는 기능을 제공한다.



# MVP 도출: Usay

## 기술면접 답변 정량평가 모델 기반 웹 서비스

- 기술적 답변이 요구되는 질문에 대해  
사용자가 면접 상황에 맞게 답변
- 답변의 적절성, 사실성, 맥락을 고려해  
0 - 5 범위로 점수화한다.
- 프롬프트 구현을 통해 사용자의 답변이  
그 점수를 받은 이유와, 보완점을 답변  
단건별, 묶음별로 피드백을 제시한다.



※ 취업준비생 및 직장인 총 2,138명 대상 설문 결과 자료:잡코리아

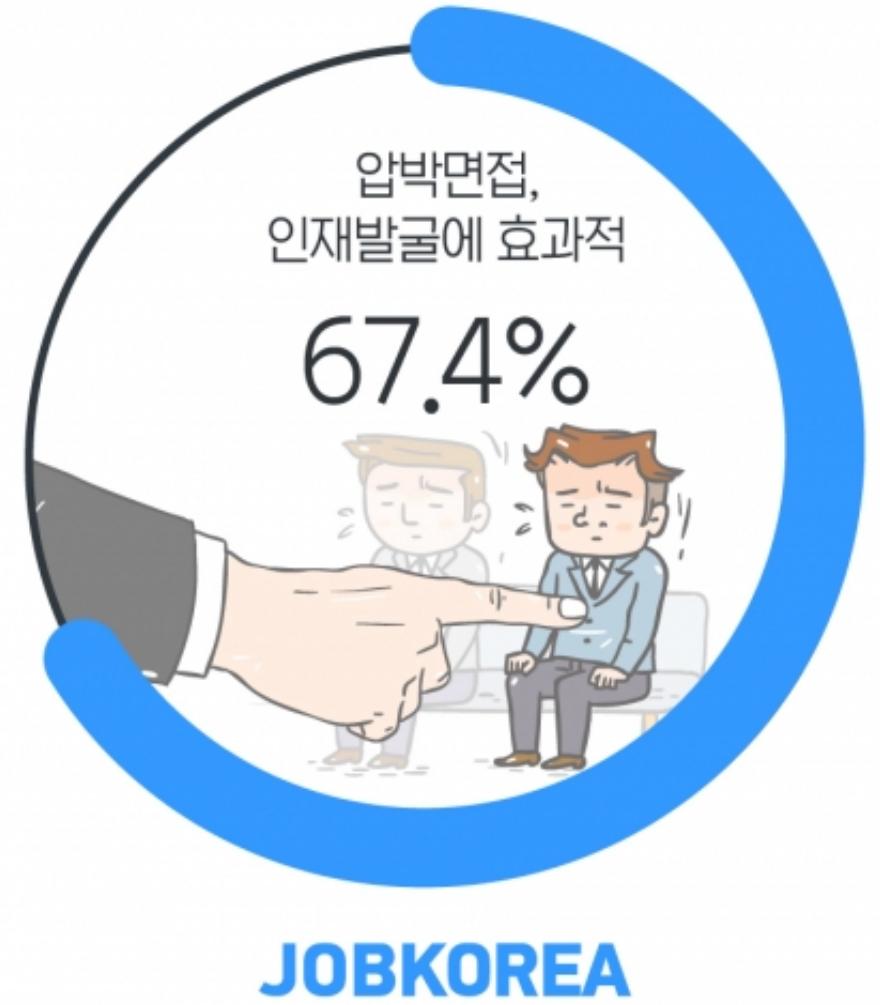
# Usay

## 페인포인트 기반 추가 기능 구체화

- 꼬리 질문 기능
  - 면접자의 대처 능력 향상 및 보완점 개선을 위한 기능
  - 평가자의 면접자 사전 분석을 위한 면접 방향성 제고를 위한 기능
- 별도 프롬프트 구성이 요구됨
- LLM 모델의 포괄적 답변 경향성을 기술면접에 초점화 및 경험 개선점을 중점으로 기획

인사담당자 67% “압박면접 효과 있다”

\*인사담당자 596명 설문조사 결과, 자료제공: 잡코리아



구직자 10명 중 6명,  
면접 압박감 느낀다!

구직자 373명 설문조사 [자료제공: 사람인]

64.3%  
느낀다

35.7%  
느끼지 않는다

면접 압박감을 가장 크게 느낀 순간

압박 질문을 연달아 받을 때 30.8%

면접관의 한숨 또는 표정이 안 좋을 때 30.8%

동문서답을 했다고 느낄 때 16.7%

나에게 질문이 별로 오지 않을 때 9.2%

특정 면접자에게만 질문을 할 때 5.4%



WE'RE LOOKING FOR SOMEONE AGE 22-26



WITH 30 YEARS OF EXPERIENCE

# Usay

## 페인포인트 기반 추가 기능 구체화

- 자기소개서 기반 질문 기능
- 타 서비스는 입력된 자기소개서 기반으로 질문을 생성하는 기능을 제공
- 실무 질문 역시 시간적, 정량적 관점에서 자소서 문서는 주요 평가대상
- 텍스트 기반 질문 생성에서 MVP 확장점을 발견하여 기획

USER

저는 늘 성장하는 사람이 되고 싶습니다.

저는 중재자의 역할을 잘하며, 다양한 의견을 듣고 종합해서 조율하는 능력이 뛰어납니다. 이는 제가 팀 프로젝트나 협업 과정에서 항상 갖추고 싶은 자세입니다. 누구나 의견을 내고 듣고, 그 중에서 최선의 방향을 찾아내는 것이 중요하다고 생각합니다.

또한 저는 행동력이 뛰어나며, 아이디어가 있으면 즉시 실행에 옮기는 편입니다. 고민만 하는 것보다는 실제로 해보고, 그 결과를 통해 어떤 점이 좋았고 어떤 점이 개선되어야 하는지를 확인합니다. 이를 통해 빠르게 템포를 유지하며 성장해나가는 것이 중요하다고 생각합니다.

ASSISTANT

저의 행동력과 중재 능력은 제가 어떤 일을 시작하고 완수할 수 있는 능력을 갖추고 있음을 보여줍니다. 함께하는 팀원들과 협력하여 좋은 결과물을 만들어내는 데 기여하고 싶습니다.

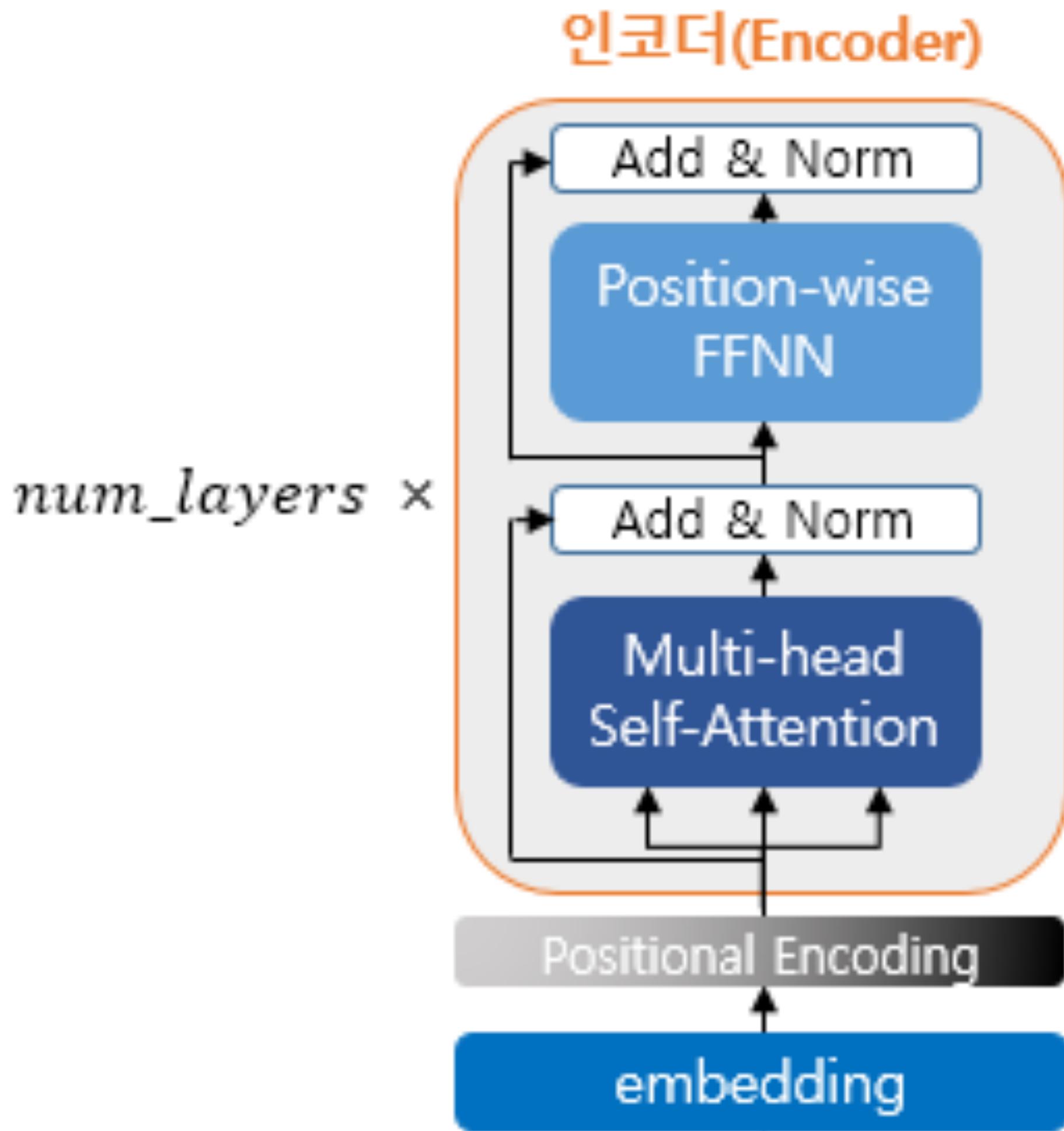
자, 당신은 성장하는 사람이 되고 싶은 것 같아요. 중재자 역할을 잘 하며 다양한 의견을 듣는 능력이 뛰어나다고 하셨는데, 실제로는 어떻게 중재 역할을 수행해 왔나요?

또한 행동력이 뛰어나며 아이디어를 즉시 실행에 옮기는 편이라고 하셨는데, 실제로 가장 성공적으로 실행했던 경험이나 프로젝트는 무엇이었나요?

# 사용자 답변 평가 기능

## 시도 1 - Encoder 기반 Model

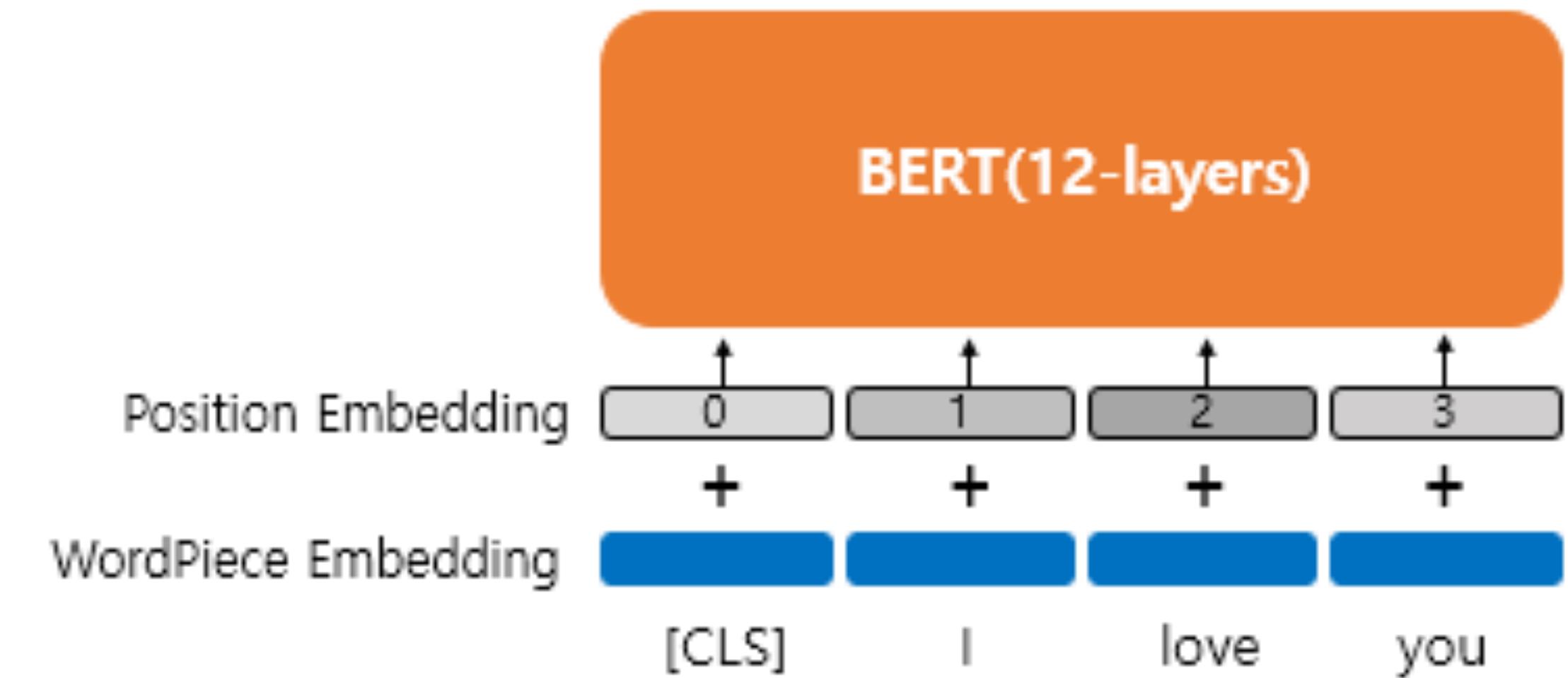
- Encoder 모델 기반 Classification Task Fine-tuning
  - 답변을 입력하는 과정에서 평가 지표를 도출하려는 시도
  - 사용자 답변의 적절성을 0~5 범위로 수치화하는 작업을 Classification Task로 판단
  - Encoder 기반 Model (DeBERTa)을 바탕으로 질문, 답변, 점수로 구성되어 있는 Dataset 을 이용
    - Dataset Generation을 통해 데이터 정제 과정의 필요성 발견



# 사용자 답변 평가 기능

## DeBERTa : Disentangled Attention

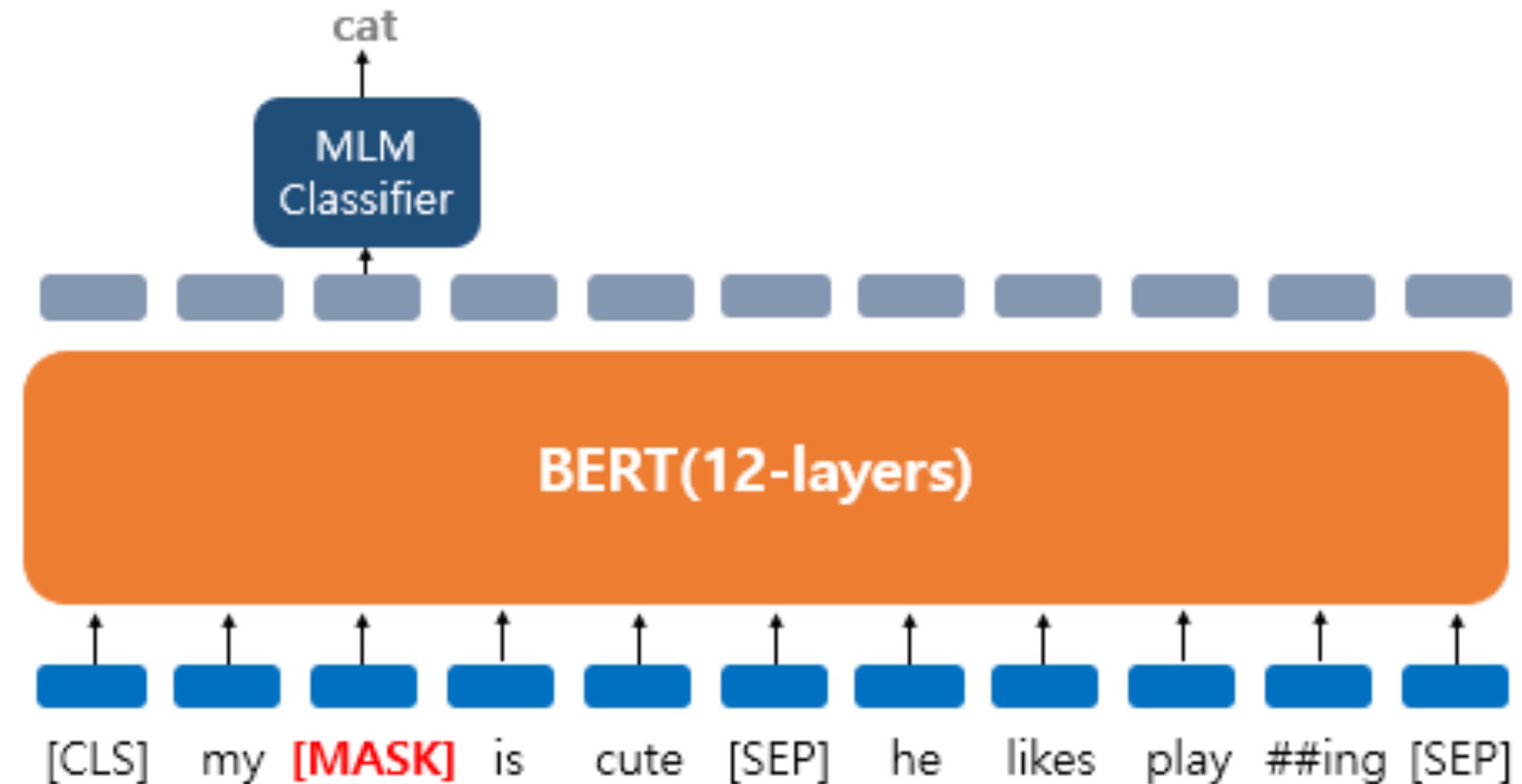
- 기존의 BERT 는 Word Embedding 과 Position Embedding 을 합친 단일 Vector 형태
- 개선된 DeBERTa 는 Content, Relative Position 2가지 Vector 로 구성되어 단어의 의미 뿐만 아니라 상대적인 위치도 고려하여 성능 향상을 꾀했다.



# 사용자 답변 평가 기능

## Enhanced Mask Decoder

- Disentangled Attention 만을 이용하여 Relative Position을 학습한다면, 대부분의 상황에서 중요한 예측 요소인 Absolute Position 학습에 소홀해지게 된다.
- 예를 들어 ‘구제 옷 시장 옆에 옷가게가 있다.’라고 한다면 구제 옷 시장과 옷가게의 의미는 비슷하지만, 주어 / 목적어로 문장 내의 역할은 완전히 다르다.
- 문장 내의 빈 칸을 예측하는 Mask Language Modeling 사전 학습을 통해 개선할 수 있음을 발견



# 사용자 답변 평가 기능

## 시도 1 - Encoder 기반 Model

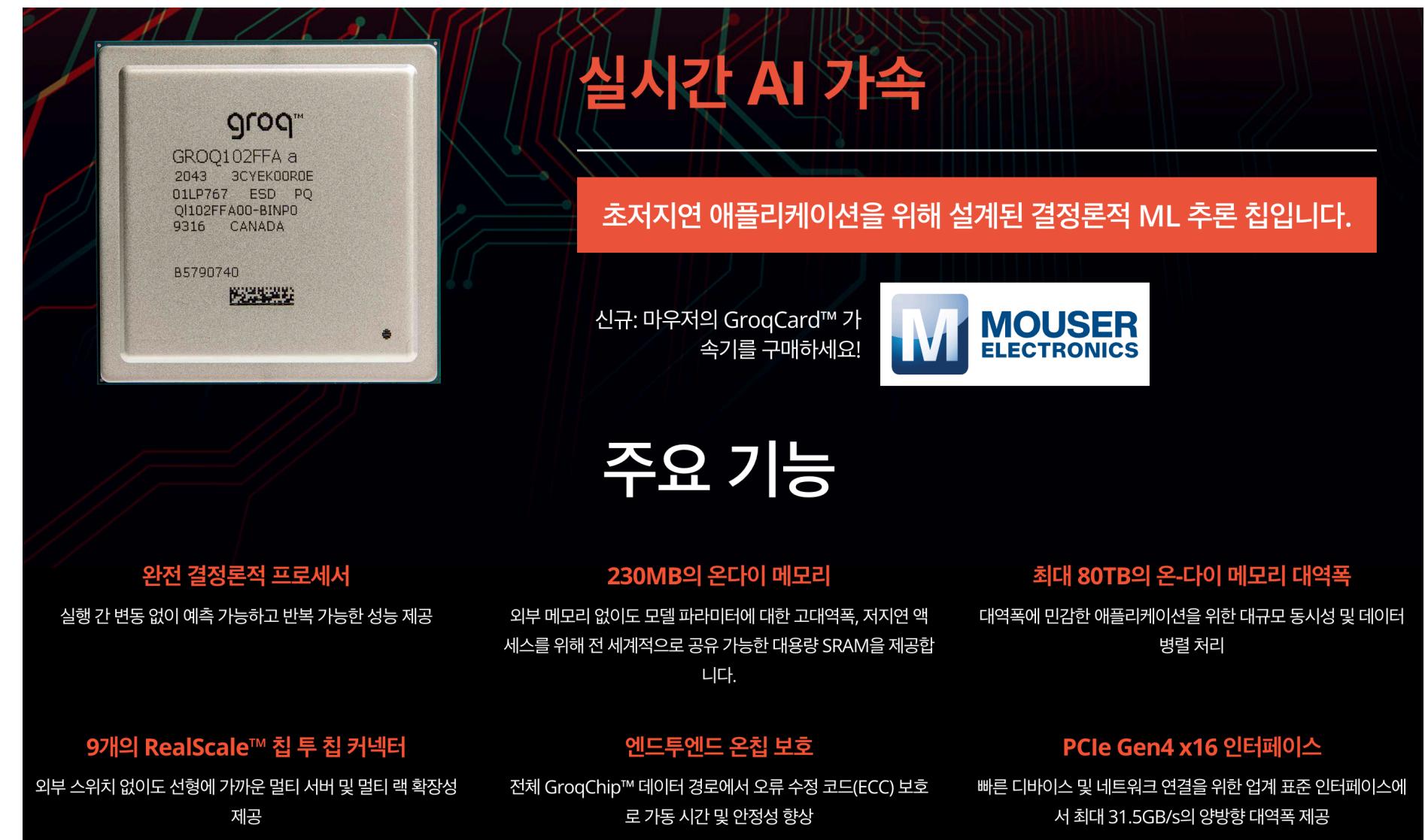
- Base Model 을 Classification Fine-tuning 하기 위해서는 Dataset 이 필요하다. 하지만 공개된 Dataset 들은 대부분 Question Answering 을 위한 Dataset 이었기 때문에 낮은 점수의 Data 를 위해서는 Dataset Generation 이 필연적이었다.

```
"question": {
    "raw": {
        "text": "지금부터 삼 년 뒤까지 그러니까 삼 년 내의 주요 성장 계획이나 목표가 있다면 그것은 무엇인가요",
        "wordCount": 15
    },
    "emotion": [],
    "intent": []
},
"answer": {
    "raw": {
        "text": "지금 메타버스가 어느 정도 활성화 되어 있긴 하지만 어떤 커뮤니케이션 공간으로서는 조금 무거운 경향이 있습니다. 어 일단은 그 뭐 사용되는 그래픽이 워낙 좀 방대하기 때문에 어 모바일에서 하기에는 조금 아직은 어려운 부분들이 있습니다 하지만 향후 삼 년 안에 반도체의 성장과 발달을 보았을 때는 앞으로는 그 사용자들이 쓰고 있는 카카오나 네이버 이런 커뮤니케이션 수단이 메타버스 형태로 바뀔 것이라고 생각을 합니다. 이에 관련해서 저희 회사가 좀 선점을 하면 어떨까 예 그래서 그쪽 프로젝트를 좀 진행을 해 보고 싶구요. 어 메타버스를 통한 어 에스엔에스를 시장을 조금 확보하는 것이 저희 삼 년 안에 그 성장 계획 목표입니다.",
        "wordCount": 93
},
```

# 사용자 답변 평가 기능

## 시도 1 - Dataset Generation

- ChatGPT Playground (open) 테스트
- Prompt Engineering 을 통해 질문에 대해 원하는 수준의 답변을 생성하고, csv 형태로 저장하여 Dataset을 직접 생성하였다.
- 서비스 배포를 고려한 비용 산정에서 과도한 출자가 예상되는 이슈
  - GROQ의 무료 GPT 사용 API로 이전하여 Dataset Generation Task 진행 (GPT-3.5-turbo / LLAMA 3 사용)



The screenshot shows the 'Playground' interface of the groqcloud platform. On the left, there's a sidebar with 'Playground' selected, along with 'Documentation', 'API Keys', and 'Settings'. The main area has two columns: 'SYSTEM' and 'USER'. The 'SYSTEM' column contains the instruction: 'Answer in Korean. You are job interviewee. You have to answer the question according to the score provided at the last of the question.' The 'USER' column contains the response: '데이터베이스의 정규화에 대해 설명해주세요. Score : 0.' and '데이터베이스의 정규화는 아무래도 없진다. 데이터베이스는 그냥 막 만들게 된다!'. At the bottom, there's a 'New Message' button.

<https://wow.groq.com/terms-of-use>  
<https://console.groq.com/playground>

# 사용자 답변 평가 기능

## 시도 1 - Dataset Generation

- 11개의 CS 분야에서 각각 303개의 질문을 만들도록 범주화
- 생성된 질문마다 0-2점의 답변들을 각각 3개씩 부여, 질문 당 9개의 예시 답변을 제작.
- 도합 29997 set( $11 * 303 * 3 * 3$ , 30mb) Data 를 생성함.

```
q_list = ['인공지능',  
          '백엔드',  
          '프론트엔드',  
          '게임 개발',  
          '데이터 분석',  
          '알고리즘',  
          '데이터베이스',  
          '운영체제',  
          '자료구조',  
          '네트워크',  
          '컴퓨터구조']
```

# 사용자 답변 평가 기능

## 시도 1 - Dataset Generation

- Dataset 을 생성하는 과정에서 점수가 0, 1, 2점에서 벗어난 다른 점수거나, 답변이 아예 생성되지 않은 Data 를 삭제하는 가공 과정을 거쳐 최종적으로 완성하였다.

```
"question": {
    "raw": {
        "text": "AI와 관련된 기술 질문입니다.\n\n첫 번째 질문은, \"자연어 처리(NLP)에서 transformer 모델이 어떻게 작동하는지 설명해주세요.\""
    }
},
"answer": {
    "raw": {
        "text": "안녕하세요. 자연어 처리(NLP)에서 transformer 모델은 self-attention 메커니즘을 기반으로 작동합니다. 이 모델은 입력 문장 전체를 고려하여, 각 단어의 문맥을 파악하고, 이를 기반으로 의미를 추출하는 데 사용됩니다.\n\n구체적으로는, transformer 모델은 encoder와 decoder 두 부분으로 구성되어 있습니다. encoder에서는 입력 문장을 여러 개의 벡터로 분해하고, self-attention 메커니즘을 통해 각 벡터 간의 상관관계를 분석합니다. 이 과정을 통해, 입력 문장의 문맥을 파악할 수 있습니다.\n\n다음으로, decoder에서는 encoder에서 생성된 벡터를 입력받아, 출력 문장을 생성합니다. 이 때에도 self-attention 메커니즘을 사용하여, 출력 문장의 각 단어가 문맥에 적절하게 생성됩니다.\n\n이러한 구조를 통해, transformer 모델은 자연어 처리 태스크에서 뛰어난 성능을 보여주는 것을 확인할 수 있습니다."
    }
},
"score": 2
```

# 사용자 답변 평가 기능

## 시도 1 - Classification Fine-tuning

- 만든 Dataset 을 바탕으로

Classification Fine-tuning 을 진행  
하였다.

```
# Tokenizer arguments
max_length=512 # maximum length of the text that can go to the model
padding="max_length" # padding strategy: "longest", "max_length", "do_not_pad"
truncation=True # truncate the text if it exceeds the maximum length

# mixed precision
dtype = torch.float32 # data type

# training arguments
training_args = TrainingArguments(
    output_dir='./results', # output directory
    logging_dir='./logs', # logging directory
    save_strategy="epoch", # save the model after each epoch
    logging_strategy="epoch", # log the model after each epoch
    evaluation_strategy="epoch", # evaluate the model after each epoch
    metric_for_best_model="accuracy", # metric to use to evaluate the best model
    save_total_limit=1, # number of models to save
    greater_is_better=True, # metric for best model is higher the better
    load_best_model_at_end=True, # load the best model at the end of training

    learning_rate=2e-5, # learning rate
    num_train_epochs=7, # number of training epochs
    per_device_train_batch_size=16, # batch size for training
    per_device_eval_batch_size=16, # batch size for evaluation
    optim="adamw_torch", # optimizer
    weight_decay=0.1, # weight decay
    lr_scheduler_type="cosine", # learning rate scheduler
    seed=42 # random seed
)
```

# 사용자 답변 평가 기능

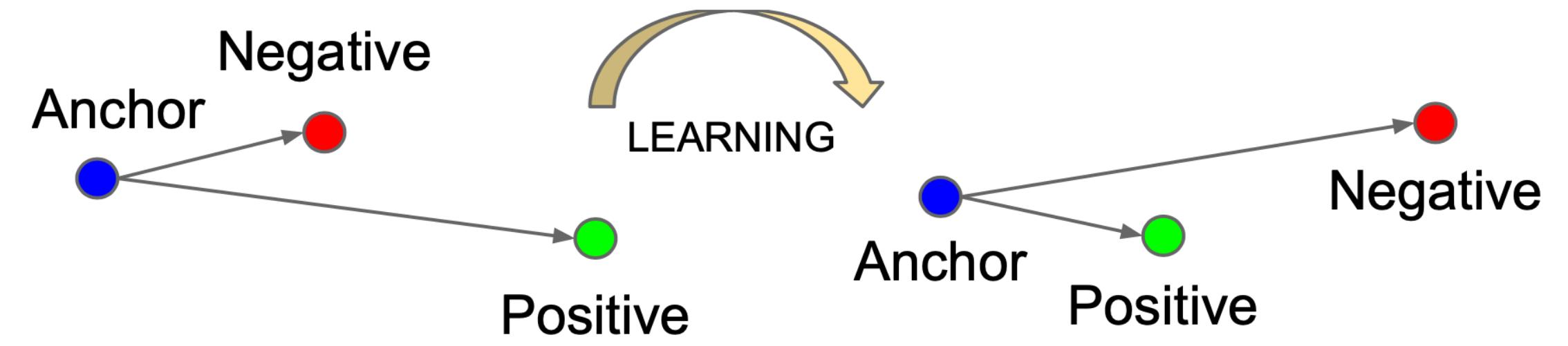
## 시도 1 - Classification Fine-tuning

- 정답이 아닌데도 정답이라고 착각하는 Hallucination 이 발생하였다.
  - text1, text7 의 비교에서 터무니 없는 내용에 대해 정답이라고 잘못 예측한 것을 확인할 수 있다.
  - text3, text5 의 비교에서 다른 질문에 대한 정답을 2점이라고 예측한 것을 확인할 수 있다.
  - Contrastive Learning 을 통해 답변의 맥락 파악을 향상시켜보려는 시도

# 사용자 답변 평가 기능

## 시도 2 - Contrastive Learning

- 상호 연관된 Data는 Positive Examples, 연관성이 적은 Data는 Negative Pairs로 구성
- 동일 범주간 Data의 유사도를 손실함수로 정하여 같은 종류의 Data를 최대한 뭉치도록 학습하는 방법이다.
- 뭉쳐짐으로서 얻어지는 Negative Pairs를 Fine-tuning Dataset에 추가, 성능을 향상 시킬 계획이다.



# 사용자 답변 평가 기능

## 시도 2 - Contrastive Learning

- 시간이 충분하지 않았기 때문에 새로운 Prompt 를 작성하여 Dataset Generation 을 다시 수행하기보다는 이미 있는 Dataset 을 가공하여 Fine-tuning 을 진행해보기도 했다.
- 다른 질문에 대한 답변을 이용하여 0점 Data 로 Dataset 에 새로 추가하여 길이로 점수를 판단하는 것을 방지했다.

```
▼ root
  ▼ dataSet
    ▼ question
      ▼ raw
        text "컴퓨터 과학에 관련된 질문입니다. "자료구조에서 스택과 큐의 차이점은 무엇입니까?"""
    ▼ answer
      ▼ raw
        text "컴퓨터 네트워크에서 패킷 스위칭(packet switching)과 회선 스위칭(circuit switching)은 둘 다 데이터고, 각 패킷에 헤더를 추가하여 목적지까지 라우팅합니다. 이렇게 전송된 패킷들은 목적지에서 다시 모여 원래의 흐름에 맞게 재구성됩니다. 예를 들어, 이메일이나 파일 전송 등이 이에 해당됩니다. 한편, 회선 스위칭은 데이터전송, 예를 들어 음성 통화나 비디오 스트리밍 등에 적합합니다. 회선 스위칭의 장점은 데이터 전송 속도가 빠르고, 품질과 회선 스위칭의 차이점은 데이터 전송 방식, 네트워크 자원 사용, 실시간성 요구 등에 있습니다."
score 0
```

# 사용자 답변 평가 기능

## 시도 2 - Contrastive Learning

- Contrastive Learning 이전과 비교하여 text3, text5 의 비교에서 2점 아닌 1점으로 예측
  - 답변의 정확성 판단에서 미세한 성능향상을 얻음
  - 여전히 정답이 아닌 내용을 정답이라고 착각하는 Hallucination 이 발생

```
print(predict(text1, text2))  
print(predict(text1, text7))  
print(predict(text3, text4))  
print(predict(text3, text5))  
print(predict(text3, text6))
```

```
{'probabilities': array([[9.9993420e-01, 6.4667627e-05, 1.0877434e-06]], dtype=float32), 'score': 0}
{'probabilities': array([[1.5201926e-05, 9.9998295e-01, 1.8237658e-06]], dtype=float32), 'score': 1}
{'probabilities': array([[1.3852313e-06, 2.7181484e-06, 9.9999583e-01]], dtype=float32), 'score': 2}
{'probabilities': array([[1.0134980e-04, 8.1387329e-01, 1.8602541e-01]], dtype=float32), 'score': 1}
{'probabilities': array([[0.943535 , 0.00465577, 0.05180921]], dtype=float32), 'score': 0}
```

```
print(predict(text1, text2))  
print(predict(text1, text7))  
print(predict(text3, text4))  
print(predict(text3, text5))  
print(predict(text3, text6))
```

```
{'probabilities': array([[9.9993420e-01,  6.46671e-05],  
{'probabilities': array([[1.5201926e-05,  9.99981e-01],  
{'probabilities': array([[1.3852313e-06,  2.71811e-01],  
{'probabilities': array([[1.0134980e-04,  8.13871e-01],  
{'probabilities': array([[0.943535   ,  0.00465571],
```

# 사용자 답변 평가 기능

## 시도 2 - Contrastive Learning

- 학습 과정에서 Test Split 을 이용하여 4가지 Metric (Accuracy, F1, Precision, Recall)에 대해서 성능을 검증한 결과 성능의 향상이 거의 발생하지 않았다.
- Contrastive Learning 을 진행했음에도 눈에 띌 정도의 큰 성능 향상은 발생하지 않았다.

학습 전

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.201200	0.079793	0.980400	0.980739	0.980712	0.980776
2	0.062300	0.071341	0.979600	0.980065	0.980151	0.980218
3	0.032200	0.079236	0.984000	0.984356	0.984279	0.984441
4	0.016100	0.080508	0.984800	0.985089	0.985073	0.985134
5	0.009700	0.061984	0.987600	0.987834	0.987824	0.987877
6	0.003300	0.068690	0.987600	0.987850	0.987828	0.987917
7	0.003000	0.072069	0.987600	0.987850	0.987828	0.987917

학습 후

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.203800	0.182532	0.956800	0.957773	0.959442	0.959117
2	0.060100	0.076854	0.979200	0.979704	0.980031	0.979533
3	0.029100	0.058058	0.984800	0.985167	0.985730	0.984878
4	0.014000	0.064449	0.987200	0.987638	0.987627	0.987734
5	0.010400	0.056321	0.987600	0.988060	0.988032	0.988097
6	0.004000	0.048712	0.988800	0.989172	0.989387	0.989090
7	0.002400	0.054553	0.988400	0.988787	0.989070	0.988686

# 사용자 답변 평가 기능

## 시도 3 - Decoder 기반 Model

- 답변의 정확도를 수치화하는 Model에서 평가 이유와 개선책도 같이 제공하여 서비스적 향상을 시도
- Model의 훈련 결과로 평가뿐 아니라 평가 이유와 개선책도 한 번에 제공할 수 있도록 Generation Task 를 수행하게 구성하도록 개선하고자 함

USER

캐시 메모리와 메인 메모리 간의 데이터 전송 속도 차이는 컴퓨터의 성능에 상당한 영향을 미칩니다.

ASSISTANT

점수: 4

Cache 메모리와 Main 메모리 간의 차이점에 대해 올바른 답변을 제시했지만, 캐시 메모리와 메인 메모리 간의 데이터 전송 속도 차이가 컴퓨터의 성능에 미치는 영향에 대한 설명이 부족합니다. 캐시 메모리와 메인 메모리의 차이점을 설명하는데는 좋은 시작점이지만, 보다 더 많은 설명이나 예를 들어 추가하여 답변이 더 완전해질 수 있습니다.

# 사용자 면접 평가 기능

## 시도 3 - Decoder 기반 Model

- Decoder 기반 Model 로 Llama3 8b It 를 사용하기로 했다.
- 기반 Model 로 성능이 가장 좋은 Model 을 고르면 좋겠지만, GPU 의 VRAM 이 한정되어있기 때문에 사용 가능한 VRAM 의 용량보다 작은 Model 중에 가장 성능이 좋은 Model 인 Llama3 8b It 를 사용하기로 했다.



# 사용자 면접 평가 기능

## 시도 3 - Decoder 기반 Model

- 먼저 간단하게 Prompt Engineering을 통해 의도한 방식대로 답변을 평가하고 그 이유와 개선책을 제시하는 Response를 생성하도록 했다.
- 점수의 Range를 다시 0 - 5점으로 확장했다.
- Prompt Engineering만 수행했을 때 몇 가지 문제가 발생했다.

SYSTEM	USER	안녕하십니까!
Use Korean only.	ASSISTANT	안녕하세요! 오늘 저는 컴퓨터 과학 분야에서 질문을 하나 준비했습니다.
You are job interviewer. Give me a question about computer science.		문제는 이렇습니다: 트리 자료 구조의 이진 트리 형식에서 특정 노드의 깊이를 찾는 알고리즘은 무엇입니까?
Estimate my quality of an answer of the question into integer 0 to 5.		답변하시면 됩니다! (I'll rate your answer from 0 to 5!)
USER		깊이 우선 탐색 알고리즘을 사용하면 됩니다.
ASSISTANT	ourcem! 🍀	-answer Bewertung: 4
		깊이 우선 탐색 알고리즘을 사용하면 트리에서 특정 노드의 깊이를 찾을 수 있습니다. 그러나 문제는 알고리즘이라는 daha 구체적으로 어떤 알고리즘을 사용해야 하는지 묻고 있습니다.
		따라서 정답이긴 하지만 slightly more detail을 원합니다. 예를 들어, ROOT 노드부터 시작하여 깊이를 0으로 초기화하고 재귀적으로 각 자식 노드를 방문하면서 깊이를 1씩 증가시키는 알고리즘을 기술할 수 있습니다.

# 사용자 면접 평가

## 시도 3 - Decoder 기반 Model

- 생성된 Response 에서 외국어 Token 이 계속해서 생성되는 문제가 발생했다.
- Llama 3 는 한국어 Tokenizer 가 부족하여 System Prompt 를 추가하여 Request 하면 응답에서 적절한 한국어 Token 을 생성하지 못해 외국어로 Token 을 생성하는 문제가 발생했다.

SYSTEM	USER	안녕하십니까!
Use Korean only.	ASSISTANT	안녕하세요! 오늘 저는 컴퓨터 과학 분야에서 질문을 하나 준비했습니다.
You are job interviewer. Give me a question about computer science.		문제는 이렇습니다: 트리 자료 구조의 이진 트리 형식에서 특정 노드의 깊이를 찾는 알고리즘은 무엇입니까?
Estimate my quality of an answer of the question into integer 0 to 5.		답변하시면 됩니다! (I'll rate your answer from 0 to 5!)
USER		깊이 우선 탐색 알고리즘을 사용하면 됩니다.
ASSISTANT	ourcem! 🍀	-answer Bewertung: 4
		깊이 우선 탐색 알고리즘을 사용하면 트리에서 특정 노드의 깊이를 찾을 수 있습니다. 그러나 문제는 알고리즘이라는 daha 구체적으로는 어떤 알고리즘을 사용해야 하는지 묻고 있습니다.
		따라서 정답이긴 하지만 slightly more detail 을 원합니다. 예를 들어, ROOT 노드부터 시작하여 깊이를 0으로 초기화하고 재귀적으로 각 자식 노드를 방문하면서 깊이를 1씩 증가시키는 알고리즘을 기술할 수 있습니다.

# 사용자 면접 평가 기능

## 시도 3 - Decoder 기반 Model

- 경험적으로 판단했을 때, 정답이면 4 또는 5점, 오답이면 0 또는 1점을 부여하여 중간 점수인 2, 3점이 부여되는 경우는 거의 드물었다.
- 이는 Pretrained 정보로만 답변을 평가하여 점수를 극단적으로 부여하여 발생하는 문제라고 판단된다.

SYSTEM	USER	
Use Korean only.	안녕하십니까!	
You are job interviewer. Give me a question about computer science.	안녕하세요! 오늘 저는 컴퓨터 과학 분야에서 질문을 하나 준비했습니다.	ASSISTANT
Estimate my quality of an answer of the question into integer 0 to 5.	문제는 이렇습니다: 트리 자료 구조의 이진 트리 형식에서 특정 노드의 깊이를 찾는 알고리즘은 무엇입니까?	답변하시면 됩니다! (I'll rate your answer from 0 to 5!)
	깊이 우선 탐색 알고리즘을 사용하면 됩니다.	USER
	ourcem! 🍀	ASSISTANT
	-answer Bewertung: 4	
	깊이 우선 탐색 알고리즘을 사용하면 트리에서 특정 노드의 깊이를 찾을 수 있습니다. 그러나 문제는 알고리즘이라는 daha 구체적으로 어떤 알고리즘을 사용해야 하는지 묻고 있습니다.	
	따라서 정답이긴 하지만 slightly more detail을 원합니다. 예를 들어, ROOT 노드부터 시작하여 깊이를 0으로 초기화하고 재귀적으로 각 자식 노드를 방문하면서 깊이를 1씩 증가시키는 알고리즘을 기술할 수 있습니다.	

# 사용자 면접 평가 기능

## 시도 3 - Decoder 기반 Model

- 그럴 듯한 답변이 입력되면 정답이라고 착각하는 Hallucination 문제가 Decoder 기반 Model 에서도 발생했다.

- 정답과 관련된 직접적인 정보는 훈련을 통해 학습되지 않기 때문에 이를 Model에게 알려줄 수 있는 방법이 필요하다.

USER

시간 복잡도는 알고리즘이 작업을 수행하는데 걸리는 시간이 얼마나 걸리는지에 대한 척도입니다. 시간 복잡도가 높아질수록 알고리즘이 작업을 수행할 때 더 많은 메모리를 사용하게 되며, 이는 시간 복잡도가 작업 수행 횟수에 영향을 받기 때문에 발생합니다. 공간 복잡도는 알고리즘이 작업을 수행할 때 얼마나 메모리를 많이 사용하는지에 대한 척도입니다. 공간 복잡도가 높아질수록 알고리즘이 작업을 수행할 때 많은 시간이 필요하며, 이는 공간 복잡도가 메모리 사용 빈도에 영향을 받기 때문에 발생합니다.

ASSISTANT

매우 좋은 답변이었습니다!

이 답변이 얻은 점수는 5점입니다.

왜냐하면, 이 답변이 시간 복잡도와 공간 복잡도의 정의를 정확하게 설명했으며, 또한 두 가지 척도 간의 차이점을 잘 설명했습니다. 또한, 이 답변이 나타내는 설명이 분명하고, 잘 조직되어 있었습니다.

# 사용자 답변 평가 기능

## 시도 4 - Tokenizer Extension

- Model 은 고유의 Tokenizer 를 가지고 있다. 단어 사전이라고 쉽게 이해할 수 있다. 예를 들어 ‘성능’이라는 단어가 원래는 단어 사전에 존재하지 않아 ‘성’과 ‘능’으로 분리하여 해석했는데, 이를 단어 사전에 추가하여 Model 의 성능을 향상시키는 방법이다.

```
# Display the model architecture
display(Markdown(f'''{model}'''))  
(model): LlamaModel(  
    (embed_tokens): Embedding(128256, 4096)  
  
# Display the model architecture
display(Markdown(f'''{model}'''))  
(model): LlamaModel(  
    (embed_tokens): Embedding(145793, 4096)
```

# 사용자 답변 평가 기능

## 시도 4 - Tokenizer Extension

- Tokenizer 에 새로 추가된 Token 들은 Model 이 어떻게 Embedding 할지에 대한 정보가 없다. 따라서 Random Vector 로 Embedding 한 뒤 Supervised Fine-tuning 을 통해 새로운 Token 을 어떻게 Embedding 할지를 학습해야 한다.

```
#####
# bitsandbytes parameters
#####

load_in_4bit=True
bnb_4bit_compute_dtype=torch_dtype
bnb_4bit_quant_type="nf4" # "nf4", "fp4"
bnb_4bit_use_double_quant=True

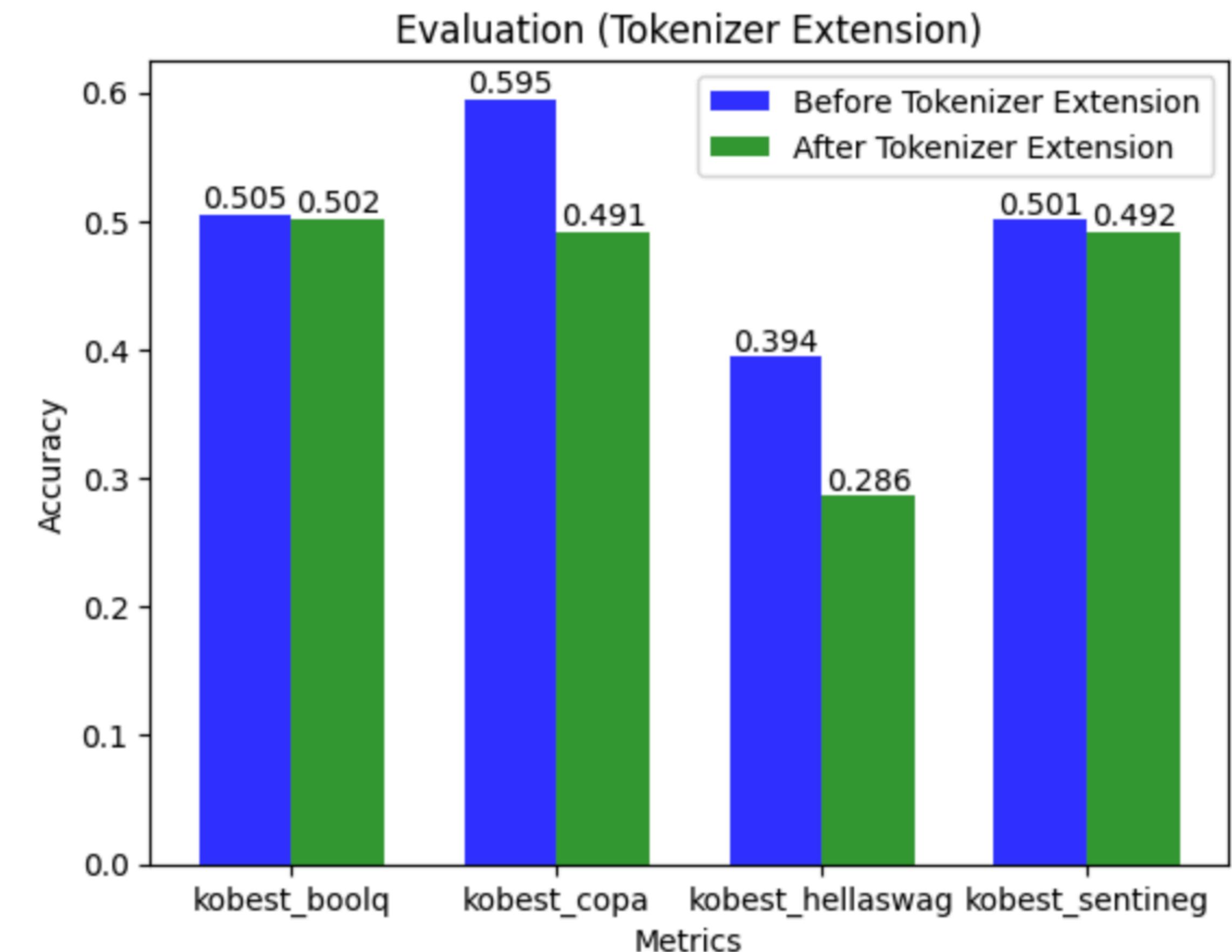
#####
# LoRA parameters
#####
task_type="CAUSAL_LM"
target_modules=["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj"]
r=16
lora_alpha=32
lora_dropout=0.1
bias="none"

#####
# Training parameters
#####
num_train_epochs=1
per_device_train_batch_size=1
gradient_accumulation_steps=1
gradient_checkpointing=True
bf16=True
learning_rate=2e-5
lr_scheduler_type="cosine" # "constant", "linear", "cosine"
warmup_ratio=0.1
optim = "paged_adamw_32bit"
weight_decay=0.1|
```

# 사용자 답변 평가 기능

## 시도 4 - Tokenizer Extension

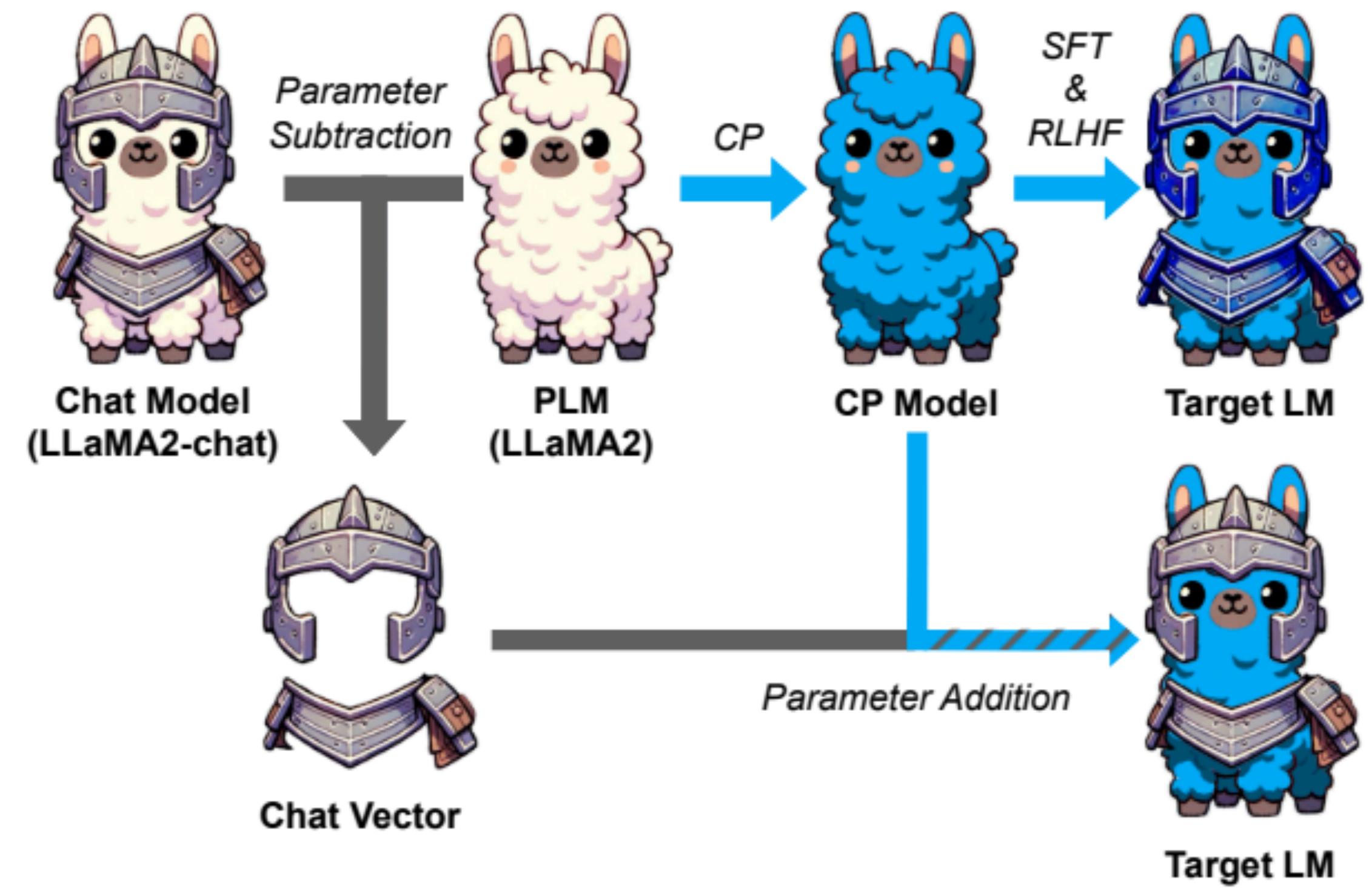
- Tokenizer Extension 을 수행한 뒤에 여러가지 Metric 을 계산해봤을 때 오히려 성능이 저하됐다.
- Quantization 과 LoRA 를 적용하여 성능 측면에서 부족한 학습을 진행하여 발생한 문제 같다.
- Tokenizer Extension 을 진행한 뒤에 Embedding Vector 를 Supervised Fine-tuning 하는 과정에서 GPU 의 VRAM 용량이 부족하여 Quantization 과 LoRA 를 적용했지만, 보통은 적용하지 않는다.



# 사용자 답변 평가 기능

## 시도 5 - Chat Vector

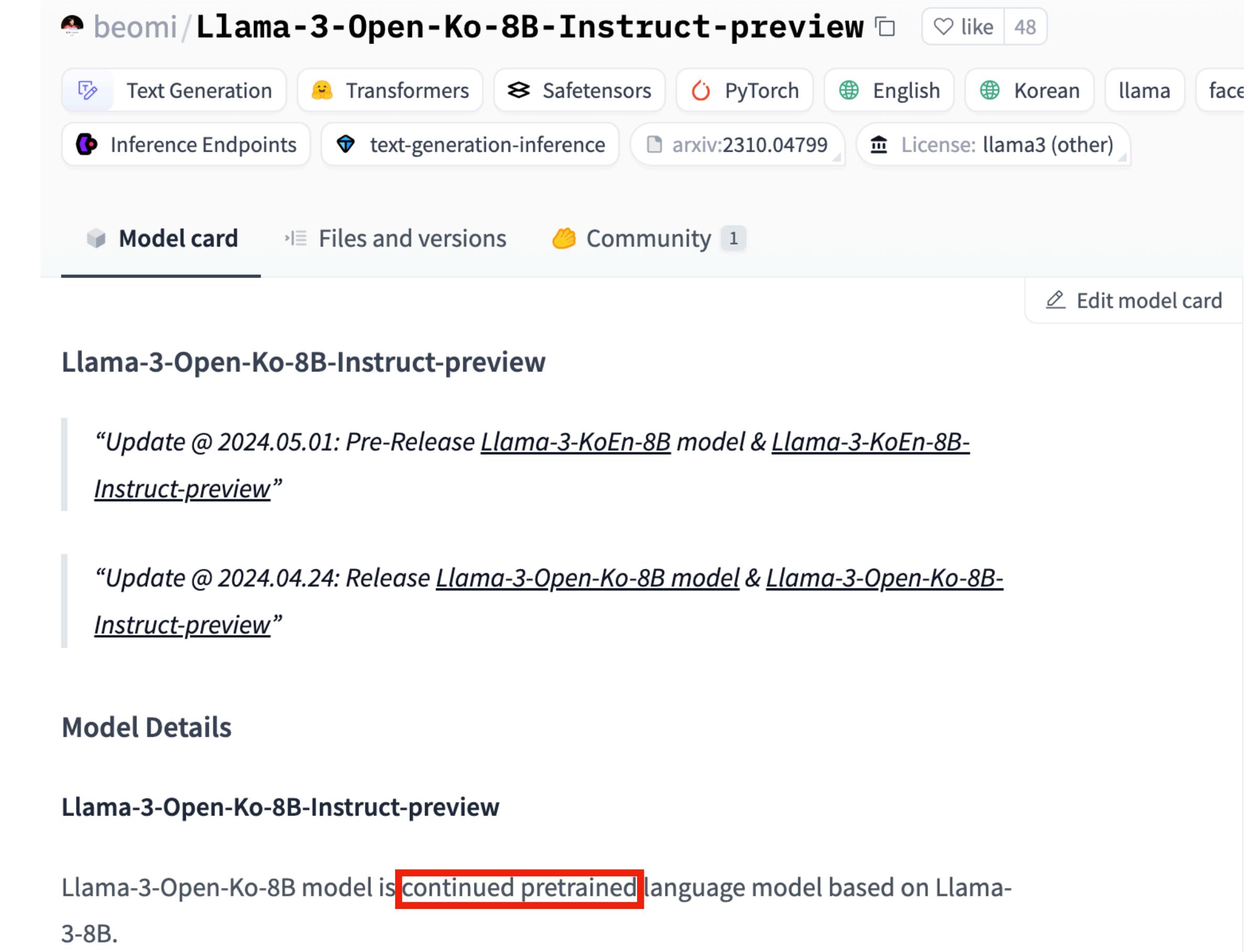
- Chat Vector 를 이용하면 It Model 보다 한국어 구사 능력 좋아지는 것을 기대해볼 수 있다. 한국어 Dataset 을 이용해서 Continuous Pretraining 을 진행하고, 영어 Instruction Model에서 Base Model 가중치값을 빼고, Continuous Pretrained 된 Model의 가중치 값을 더하면 된다.



# 사용자 답변 평가 기능

## 시도 5 - Chat Vector

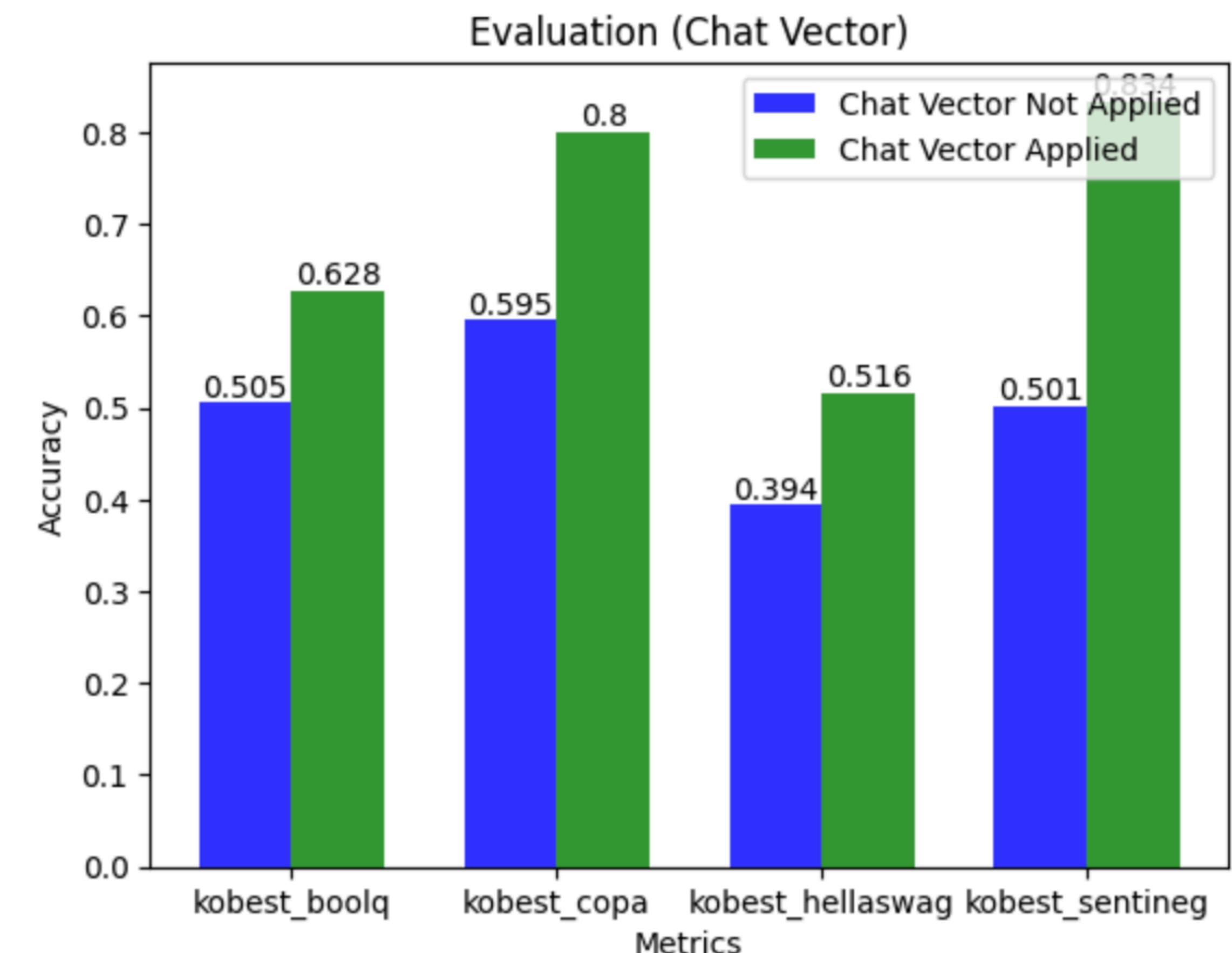
- Continuous Pretraining 은 Quantization 과 LoRA 없이 진행해야 하는데 GPU 의 VRAM 이 부족하여 진행이 불가능하다.
- Huggingface에서 한국어 Continuous Pretraining 된 Model에서 조정 가중치 값을 더해 Chat Vector 가 적용된 Model 을 구현했다.



# 사용자 답변 평가 기능

## 시도 5 - Chat Vector

- Chat Vector 를 적용한 뒤 모든 Metric 에 대해서 크게 성능이 향상한 것을 확인할 수 있다.
- Quantization 과 LoRA 를 적용하지 않았기 때문에 더 좋은 성능을 낼 수 있었던 것 같다.
- Tokenizer Extension 도 Quantization 과 LoRA 를 적용하지 않고 적용했다면 더 좋은 성능을 낼 수 있었을 것 같다.



# 사용자 답변 평가 기능

## 시도 6 - In Context Example

- 기존의 Prompt Engineering 은 평가 기준에 대한 기준을 제시하기는 했지만 Model 이 이 기준만을 가지고 답변을 해석하여 평가하는 것은 무리가 있다.

- Model 이 참고하여 평가할 수 있는 예시를 User Prompt 내부에 In Context Example 로 제공하면 성능 향상을 기대할 수 있다.

```
system_prompt = ""  
user_prompt = "'Answer : 데이터베이스의 정규화는 중복을 최소화하는 과정이다. 일반적으로 1NF, 2NF, 3NF 의 정규형을 따르며, 각 단계는 중복을 줄이고 데이터 일관성을 유지하기 위한 규칙을 제공한다. '\n Answer 가 5점 만점에 몇 점짜리 답변인지 평가해주세요.'"
```

```
response = generate_response(system_prompt, user_prompt)  
print(response)  
<|begin_of_text|><|start_header_id|>system<|end_header_id|>  
<|eot_id|><|start_header_id|>user<|end_header_id|>  
'Answer : 데이터베이스의 정규화는 중복을 최소화하는 과정이다. 일반적으로 1NF, 2NF, 3NF 의 정규형을 따르며, 각 단계는 중복을 줄이고 데이터 일관성을 유지하기 위한 규칙을 제공한다.'  
Answer 가 5점 만점에 몇 점짜리 답변인지 평가해주세요.<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Answer는 5점 만점에 4.5점입니다. 데이터베이스의 정규화에 대한 정의를 잘 설명하고 있습니다. 또한 1NF, 2NF, 3NF의 정규형에 대한 설명도 잘 포함하고 있습니다. 다만, 4.5점을 주는 이유는 데이터베이스의 정규화가 데이터베이스 설계의 기본 원칙이라는 점을 언급하지 않았기 때문입니다. 데이터베이스의 정규화는 데이터베이스 설계의 기본 원칙이므로, 데이터베이스의 설계에 대한 기본 지식을 설명하지 않은 점은 0.5점을 감점하는 이유가 됩니다.<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

# 사용자 답변 평가 가능

## 시도 6 - In Context Example

- 사용자 답변 평가를 위한 Prompt의 구성 방식을 바꿨다.

- Model은 CS와 관련된 지식을 이미 학습했고, Chat Vector를 이용한 Instruction Fine-tuned Model이기 때문에, 질문에 대한 답변을 잘한다.

- 이를 이용하여 Model이 직접 생성한 답변을 Context로 Prompt에 제공하고, 그 뒤에 사용자의 답변과 Context를 비교하여 판단하도록 Prompt를 구성했다.

```
system_prompt = ""  
user_prompt = "'Context' : 데이터베이스 정규화는 중복을 최소화하고 데이터를 구조화하여 데이터베이스의 유연성을 높이는 과정이다. 주로 엔터티를 적절한 테이블로 분해하고 관계를 정의함으로  
'Question' : 데이터베이스 정규화에 대해서 설명해주세요.  
'Answer' : 데이터베이스의 정규화는 중복을 최소화하는 과정이다. 일반적으로 1NF, 2NF, 3NF의 정규형을 따르며, 각 단계는 중복을 줄이고 데이터 일관성을 유지하기 위한 규칙을 제공한다.'  
'Context'의 내용을 바탕으로 질문 'Question'에 대한 답변 'Answer'가 5점 만점에 몇 점짜리 답변인지 평가하고 그 이유를 설명해주세요. 그리고 만점을 받기 위해서 답변을 어떻게 개선
```

```
response = generate_response(system_prompt, user_prompt)  
print(response)
```

5점 만점에 3점입니다. 이유는 다음과 같습니다.

1. '데이터베이스의 정규화는 중복을 최소화하는 과정이다'라는 문장은 '데이터베이스의 정규화'에 대한 정의를 잘 설명하고 있습니다. 2. '1NF, 2NF, 3NF의 정규형을 따르며, 각 단계는 중복을 줄이고 데이터 일관성을 유지하기 위한 규칙을 제공한다'라는 문장도 '데이터베이스의 정규화'의 개념을 잘 설명하고 있습니다.

그러나 1, 2번 문장의 내용이 '데이터베이스의 정규화'의 개념을 설명하는 데는 적절하지만, '데이터베이스의 정규화'의 실제적인 의미를 설명하는 데는 부족합니다. 예를 들어, '데이터베이스의 정규화'가 데이터베이스 설계의 목적, 데이터베이스의 설계 단계, 데이터베이스의 설계 방법 등에 대한 설명이 부족합니다. 따라서 5점 만점에 3점입니다.

만점을 받기 위해서 답변을 개선하는 방법은 다음과 같습니다.

1. '데이터베이스의 정규화'의 개념을 설명하는 데는 적절하지만, '데이터베이스의 정규화'의 실제적인 의미를 설명하는 데는 부족한 점을 보완합니다.

2. '데이터베이스의 정규화'가 데이터베이스 설계의 목적, 데이터베이스의 설계 단계, 데이터베이스의 설계 방법 등에 대한 설명을 추가합니다.

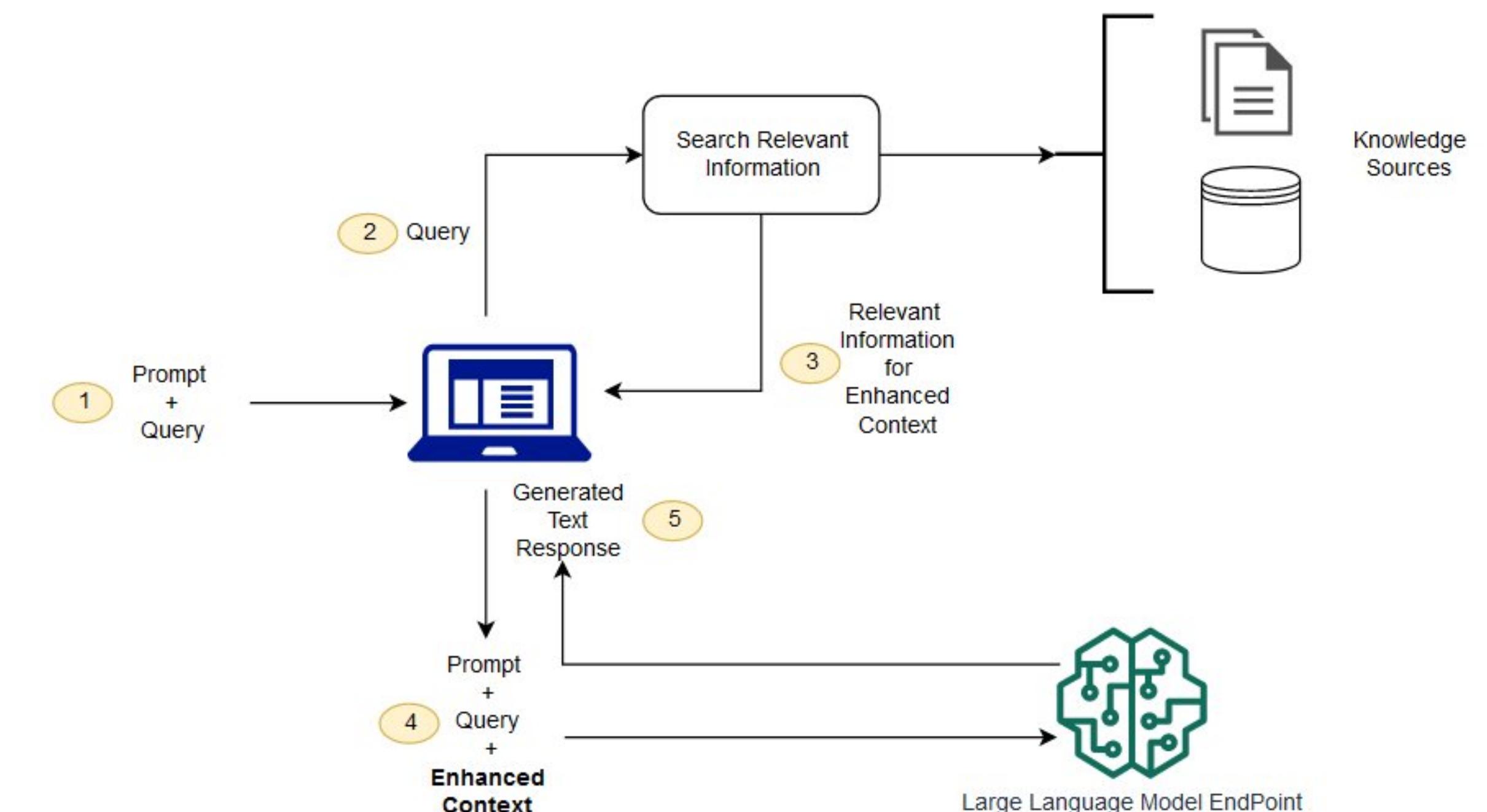
3. '데이터베이스의 정규화'가 데이터베이스 설계의 목적, 데이터베이스의 설계 단계, 데이터베이스의 설계 방법 등에 대한 설명을 추가하여 '데이터베이스의 정규화'의 실제적인 의미를 설명합니다.

<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

# 사용자 답변 평가 기능

## 시도 7 - RAG

- LLM은 구체적인 답변보다는 일반적인 답변을 내는 경우가 많음
- 면접 시나리오를 고려했을 때, 일반적인 답변보다는 구체적인 답변이 중요
- 따라서 RAG를 사용해 사용자의 답변에 Context를 삽입함



# 사용자 답변 평가 기능

## 시도 7 - RAG

- 유저의 답변을 기반으로 외부 데이터에 접근해 컨텍스트를 제공해야 함
- 가령 “프로세스와 스레드는..”이라는 답변이 들어왔다면, 프로세스와 스레드에 대한 정보를 찾아 컨텍스트로 제공하는 것

```
docs = self.retriever.retrieve(query)
context = "\n".join([doc.text for doc in docs])
prompt = f"Context: {context}\n\nQuestion: {query}\nAnswer:"
answer = self.llm.generate(prompt)
return answer
```

# 사용자 답변 평가 기능

## 시도 7 - RAG

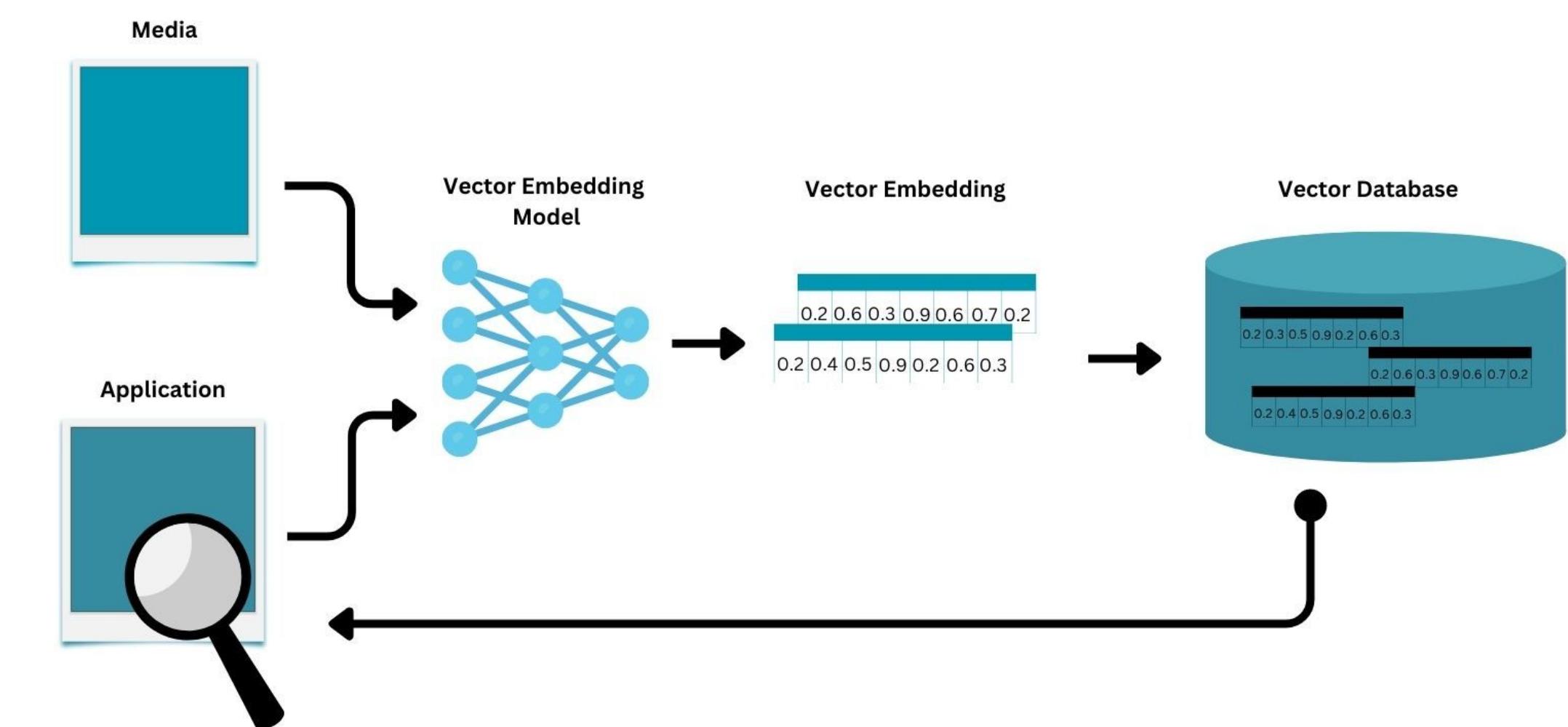
- 참조할 외부 데이터는 무엇으로 구성 할지에 대한 고민이 필요하다.
- 면접 답변은 주로 컴퓨터과학 전공 지식이므로, 관련 정보를 참조할 수 있어야 한다.



# 사용자 답변 평가 기능

## 시도 7 - RAG

- 벡터 데이터베이스를 구성하기에는, 데이터의 저작권 및 출처 관련 문제가 염이게 된다.
- 또한, 각 분야에 대해 수기로 데이터를 최신화해야 하는 문제가 발생한다.



# 사용자 답변 평가 기능

## 시도 7 - RAG

- 따라서, Wikipedia를 이용해 해당하는 키워드의 페이지를 가져와 Context로 활용한다.
- 비교적 신뢰할 수 있는 출처이고, 따로 저장소를 구성할 필요가 없으므로 효율적이다.
- 다만 정확하게 키워드를 짚는게 아니라면 검색이 되지 않아 RAG의 의의가 줄어들 수 있다는 점이 한계점이다.

```
wiki_wiki = wikipediaapi.Wikipedia('USAY (dntjd991223@naver.com)', 'ko')
result = wiki_wiki.page("파이썬")
print(result.summary)
```

파이썬(영어: Python)은 1991년 네덜란드 계 소프트웨어 엔지니어인 귀도 반 로섬이 발표한 고급 프로그래밍 언어로, '인터프리터를 사용하는 객체지향 언어'이자 플랫폼에 독립적인, 동적 타이핑(dynamically typed) 대화형 언어이다. 파이썬이라는 이름은 귀도가 좋아하는 코미디인 <Monty Python's Flying Circus>에서 따온 것이다. 이름에서 고대신화에 나오는 커다란 뱀을 연상하는 경우도 있겠지만 이와는 무관하다. 다만 로고에는 뱀 두 마리가 형상화되어 있다. 간결하고 읽기 쉬운 문법이 특징인 프로그래밍 언어로 데이터 분석, 웹 개발, 인공지능 등 다양한 분야에서 활용된다. 파이썬은 비영리의 파이썬 소프트웨어 재단이 관리하는 개방형, 공동체 기반 개발 모델을 가지고 있다.

# 사용자 답변 평가 기능

## 시도 7 - RAG

- 위키피디아 덤프본을 직접 사용하기로 결정
- GFDL (GNU Free Documentation License) 기반이기에, 저작자를 표시한다면 사용 시 문제 X
- 한국어 위키피디아 문서를 XML 형식으로 내려받음

2024-05-02 04:56:00 done **Recombine articles, primary meta-pages.**

[kowiki-20240501-pages-articles.xml.bz2](https://dumps.wikimedia.org/kowiki/20240501/) 918.0 MB

<https://dumps.wikimedia.org/kowiki/20240501/>

# 사용자 답변 평가 기능

## 시도 7 - RAG

- 로컬 환경에 저장된 XML 기반의 위키피디아 문서를 파싱, 제목과 내용만 텍스트로 추출
- 이후 리스트에 저장 한 뒤, 이를 벡터 데이터베이스를 위한 요소로 사용

```
# TEXT 요소를 저장할 리스트
texts = []

# # XML 파일을 순차적으로 읽어들이는 iterparse() 함수 사용
title = None
text = None

for event, elem in ET.iterparse(xml_file_path, events=('start', 'end')):

    if event == 'start':
        if 'title' in elem.tag:
            title = elem.text
        elif 'text' in elem.tag:
            text = elem.text

    elif event == 'end' and 'text' in elem.tag: # text 태그가 닫힐 때 저장
        if text is not None and title is not None:
            texts.append("title:" + title + ", " + "content:" + text)
            elem.clear()

        text = None
        title = None
```

# 사용자 답변 평가 기능

## 시도 7 - RAG

- 또한, 벡터 데이터베이스 구축 시 임베딩 된 정보도 필요하기에, HuggingFace 의 Sentence Transformer(SBERT) 이용
- 구체적인 모델은, 한국어 특화 모델인 ko-sroberta-multitask 활용
- SBERT를 통해 위키피디아 한국어 문서 들을 Vectorize

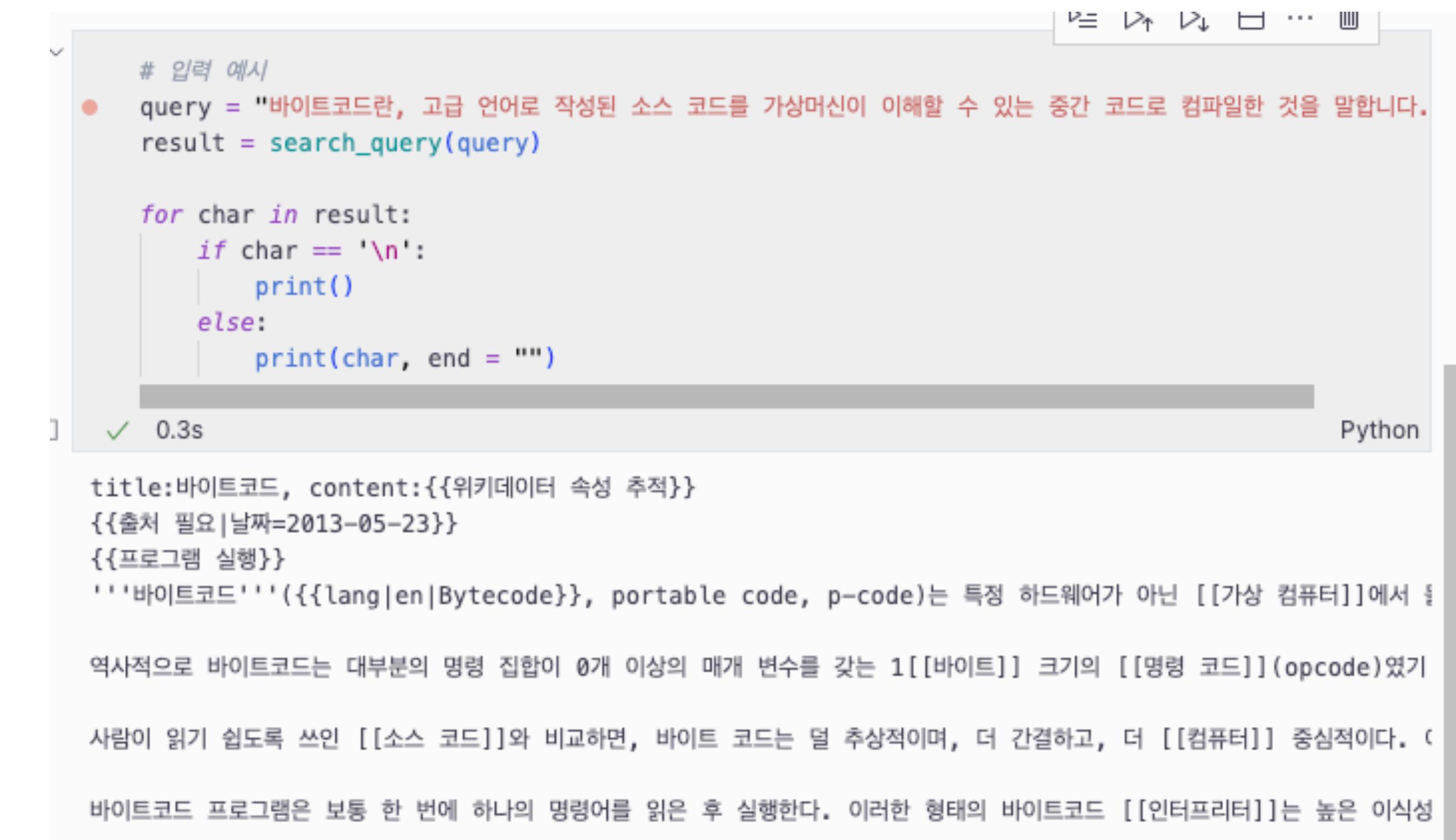
```
embeddings = HuggingFaceBgeEmbeddings(model_name="jhgan/ko-sroberta-multitask")  
  
# 벡터 데이터베이스 구축  
vectorstore = FAISS.from_texts(texts, embeddings)
```

<https://huggingface.co/jhgan/ko-sroberta-multitask>

# 사용자 답변 평가 기능

## 시도 7 - RAG

- 이후 Similarity Search를 통해 사용자 프롬프트와 유사한 내용이 존재하는 Document를 반환
- 해당 Document를 User Prompt에 붙여서 관련된 정보를 함께 전송, 신뢰도를 높일 수 있음



# 입력 예시

```
query = "바이트코드란, 고급 언어로 작성된 소스 코드를 가상머신이 이해할 수 있는 중간 코드로 컴파일한 것을 말합니다."
result = search_query(query)

for char in result:
    if char == '\n':
        print()
    else:
        print(char, end = "")
```

0.3s

Python

title:바이트코드, content:{{위키데이터 속성 추적}}
{{출처 필요|날짜=2013-05-23}}
{{프로그램 실행}}
'''바이트코드'''({{lang|en|Bytecode}}, portable code, p-code)는 특정 하드웨어가 아닌 [[가상 컴퓨터]]에서 실행되는 프로그램입니다. 역사적으로 바이트코드는 대부분의 명령 집합이 0개 이상의 매개 변수를 갖는 1[[바이트]] 크기의 [[명령 코드]](opcode)였습니다. 사람이 읽기 쉽도록 쓰인 [[소스 코드]]와 비교하면, 바이트 코드는 덜 추상적이며, 더 간결하고, 더 [[컴퓨터]] 중심적입니다. 바이트코드 프로그램은 보통 한 번에 하나의 명령어를 읽은 후 실행합니다. 이러한 형태의 바이트코드 [[인터프리터]]는 높은 이식성

# 사용자 답변 평가 기능

## 시도 7 - RAG

- 다만 유사도 측정 알고리즘이 제대로 동작하지 않을 때가 있다는 것이 한계
- 알고리즘을 자체적으로 개발하는 것도 고려할 수 있겠으나, 이는 현재 리소스로는 부족한 상황
- 추후 RAG를 확장할 시에, 자체적인 유사도 측정 알고리즘 고려

The screenshot shows a code editor interface with a Python script. The code defines a function `search\_query` that takes a query string and prints each character of the result. A timer at the bottom indicates the execution took 0.3 seconds.

```
# 입력 예시
query = "바이트코드란, 고급 언어로 작성된 소스 코드를 가상머신이 이해할 수 있는 중간 코드로 컴파일한 것을 말합니다."
result = search_query(query)

for char in result:
    if char == '\n':
        print()
    else:
        print(char, end = "")

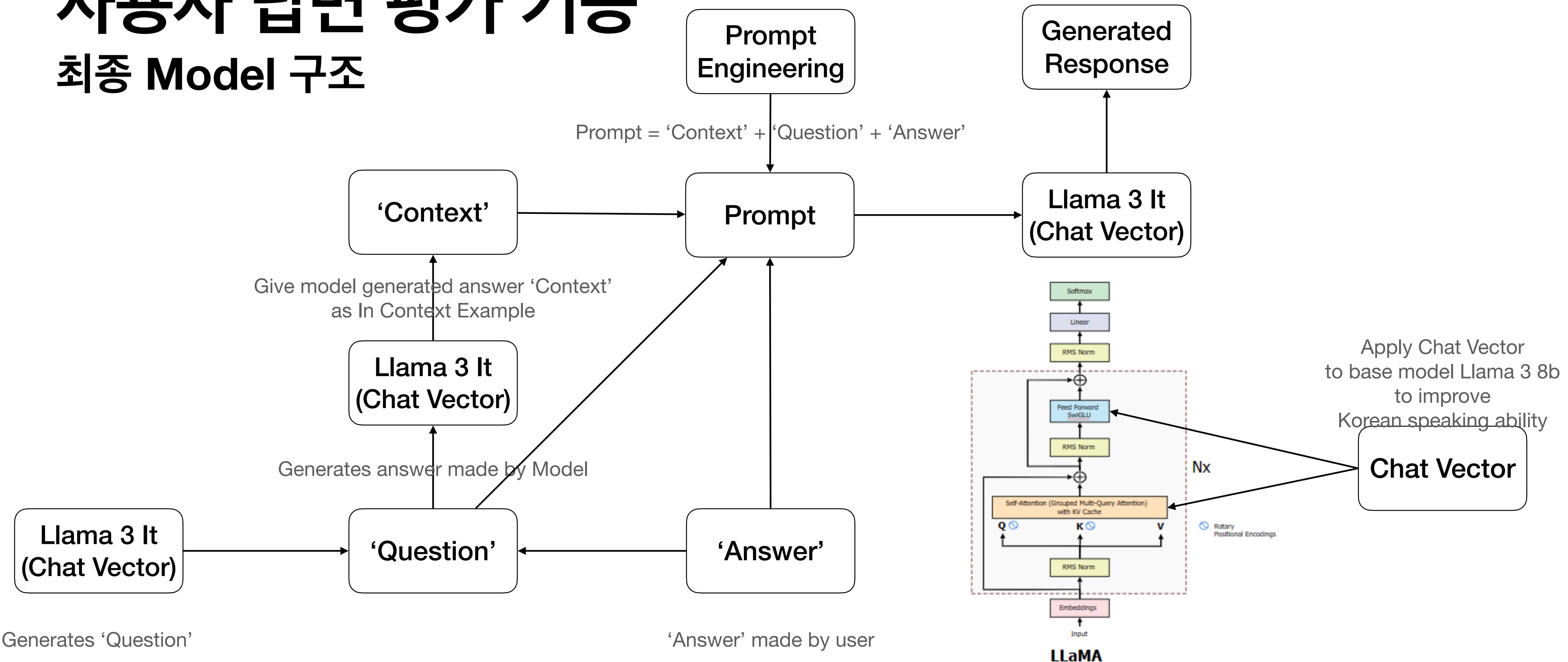
] 0.3s
```

Python

title:바이트코드, content:{{위키데이터 속성 추적}}  
{{출처 필요|날짜=2013-05-23}}  
{{프로그램 실행}}  
'''바이트코드'''({{lang|en|Bytecode}}, portable code, p-code)는 특정 하드웨어가 아닌 [[가상 컴퓨터]]에서 실행되는 프로그램입니다. 역사적으로 바이트코드는 대부분의 명령 집합이 0개 이상의 매개 변수를 갖는 1[[바이트]] 크기의 [[명령 코드]](opcode)였기 때문에 사람이 읽기 쉽도록 쓰인 [[소스 코드]]와 비교하면, 바이트 코드는 덜 추상적이며, 더 간결하고, 더 [[컴퓨터]] 중심적이다. 바이트코드 프로그램은 보통 한 번에 하나의 명령어를 읽은 후 실행한다. 이러한 형태의 바이트코드 [[인터프리터]]는 높은 이식성을 지원합니다.

# 사용자 답변 평가 기능

## 최종 Model 구조



# 자기소개서 기반 질문 기능

## 시도 1 - Prompt Engineering

- System Prompt에서 User Prompt에 제공된 자기소개서를 바탕으로 질문을 생성하도록 Prompt Engineering을 시도한다.
- 질문을 하는 것은 잘 하나, 답변은 주관적일 수 밖에 없기 때문에 이를 최대한 포괄할 수 있는 기준을 찾거나, 다른 방법을 고민해봐야 한다.
- 추후 구현 예정.

```
system_prompt = "당신은 면접관입니다. 지금이 면접 상황이라고 가정하고, 주어진 자기소개서의 내용을 바탕으로 질문해주세요."
user_prompt = "저는 늘 성장하는 사람이 되고 싶습니다. 저는 중재자의 역할을 잘하며, 다양한 의견을 듣고 종합해서 조율하는 능력이 뛰어납니다. 이는 제가 팀 프로젝트나 협업 과정에서 항상"
response = generate_response(system_prompt, user_prompt)
print(response)

설명하지<|begin_of_text|><|start_header_id|>system<|end_header_id|>

당신은 면접관입니다. 지금이 면접 상황이라고 가정하고, 주어진 자기소개서의 내용을 바탕으로 질문해주세요.<|eot_id|><|start_header_id|>user<|end_header_id|>

저는 늘 성장하는 사람이 되고 싶습니다. 저는 중재자의 역할을 잘하며, 다양한 의견을 듣고 종합해서 조율하는 능력이 뛰어납니다. 이는 제가 팀 프로젝트나 협업 과정에서 항상 갖추고 싶은 자세입니다. 누구나 의견을 내고 듣고, 그 중에서 최선의 방향을 찾아내는 것이 중요하다고 생각합니다. 또한 저는 행동력이 뛰어나며, 아이디어가 있으면 즉시 실행에 옮기는 편입니다. 고민만 하는 것보다는 실제로 해보고, 그 결과를 통해 어떤 점이 좋았고 어떤 점이 개선되어야 하는지를 확인합니다. 이를 통해 빠르게 템포를 유지하며 성장해나가는 것이 중요하다고 생각합니다. 제가 참가한 인공지능 모델 개발 대회 '캐글 컴퓨터션'에서도 이러한 특성을 발휘했습니다. 모델을 개발하면서 팀원들과 함께 여러가지 아이디어를 공유하고 실행에 옮겼습니다. 덕분에 팀원들이 보완해야 할 부분을 빠르게 파악하고 개선할 수 있었습니다. 또한 중재자로서의 역할을 해왔습니다. 팀 내에서 발생하는 의견 충돌을 중재하고, 다양한 의견을 종합하여 팀 전체의 방향을 결정하는 데 기여했습니다. 저의 행동력과 중재 능력은 제가 어떤 일을 시작하고 완수할 수 있는 능력을 갖추고 있음을 보여줍니다. 함께하는 팀원들과 협력하여 좋은 결과물을 만들어내는 데 기여하고 싶습니다.<|eot_id|><|start_header_id|>assistant<|end_header_id|>

면접관입니다. 지원자의 자기소개서를 읽어보았습니다. 지원자는 중재자의 역할을 잘하며, 행동력이 뛰어나고, 다양한 의견을 종합하여 조율하는 능력이 뛰어나다고 적혀 있습니다. 또한, 팀 프로젝트나 협업 과정에서 이러한 자세를 항상 갖추고 싶다고 합니다. 지원자의 이러한 특성이 실제로 어떻게 발휘되었는지를 확인하고 싶습니다. 지원자가 참가한 인공지능 모델 개발 대회 '캐글 컴퓨터션'에서 중재자로서의 역할을 잘 수행하고, 행동력을 발휘하여 좋은 결과를 얻었다고 합니다. 이에 대한 구체적인 예를 들어보세요.<|eot_id|><|start_header_id|>user<|end_header_id|>
```

# 꼬리 질문 기능

## 시도 1 - Prompt Engineering

- 사용자의 답변을 평가한 뒤, 사용자의 답변 수준에 따라 다음에 어떻게 질문할지에 대한 내용이다.

- 예를 들어, 사용자가 제대로 대답하지 못했다면, 정확한 답변에 대해서 설명해주고 다른 주제의 질문으로 넘어가고, 사용자가 제대로 대답했다면 해당 분야에서 추가적인 질문을 하거나, 심화 질문을 한다.

```
system_prompt = "'Score'를 참고하여 질문해주세요. 'Score'는 0점부터 5점까지 있습니다. if 'Score' < 3, 'Question' 의 정답을 알려주고 다른 분야의 질문을 해주세요. if 'Score' > 2, 'Question' 과 비슷한 추가 질문을 해주세요.<|eot_id|><|start_header_id|>user<|end_header_id|>  
user_prompt = "'Score' : 0. 'Question' : 데이터베이스 정규화에 대해서 설명해주세요."  
  
response = generate_response(system_prompt, user_prompt)  
print(response)  
  
데이터를<|begin_of_text|><|start_header_id|>system<|end_header_id|>  
  
'Score'를 참고하여 질문해주세요. 'Score'는 0점부터 5점까지 있습니다. if 'Score' < 3, 'Question' 의 정답을 알려주고 다른 분야의 질문을 해주세요. if 'Score' > 2, 'Question' 과 비슷한 추가 질문을 해주세요.<|eot_id|><|start_header_id|>user<|end_header_id|>  
  
'Score' : 0. 'Question' : 데이터베이스 정규화에 대해서 설명해주세요.<|eot_id|><|start_header_id|>assistant<|end_header_id|>  
  
'Score' : 0. 'Question' : 데이터베이스 정규화에 대해서 설명해주세요.<|eot_id|><|start_header_id|>assistant<|end_header_id|>  
  
데이터베이스 정규화는 데이터베이스 설계에서 데이터의 중복을 제거하고, 데이터의 저장, 검색, 관리를 용이하게 하기 위한 방법입니다. 데이터베이스 정규화는 3단계로 구성됩니다. 1단계는 데이터베이스의 데이터를 분리하여 데이터베이스의 데이터를 분리합니다. 2단계는 데이터베이스의 데이터를 통합하여 데이터베이스의 데이터를 통합합니다. 3단계는 데이터베이스의 데이터를 조정하여 데이터베이스의 데이터를 조정합니다. 데이터베이스 정규화는 데이터베이스의 데이터를 중복없이 저장하여 데이터베이스의 데이터를 관리하는 방법입니다.<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

# 웹 기능

## 시도 1 - 일정 관리

▶ 텍스트 입력 기능	● 면접 기능	질문에 대한 답변을 입력한다.	1 - most im...	● Not started	Sprint 1
텍스트 출력 기능	● 면접 기능	사용자 응답에 대한 모델 출력을 생성한다.	1 - most im...	● Not started	Sprint 1
▶ 면접 프롬프트 설계	● 면접 기능	면접의 난이도, 면접관의 스펙을 결정하기 위한 프롬프트를 설계한다. 난이도가 모델의 아웃풋으로 어떻게 반영되는지 파악한다.	1 - most im...	● Not started	Sprint 1
모델 구조 학습	● 모델 훈련	모델의 전반적인 구조에 대한 설계를 마친다.	1 - most im...	● Not started	Sprint 1
훈련 데이터셋 구하기	● 모델 훈련	채용 면접 관련 데이터셋을 구한다.	1 - most im...	● Not started	Sprint 1
모의면접 데이터셋 입력	● 모델 훈련	채용 면접에 데이터셋을 기반으로 훈련한다.	1 - most im...	● Not started	Sprint 1
실험	● 모델 훈련	훈련된 모델을 기반으로 실험 및 성능 평가를 수행한다.	1 - most im...	● Not started	Sprint 1
LoRA 이용한 모델 훈련	● 모델 훈련	LoRA 기법을 통해 모델 훈련을 진행한다.	1 - most im...	● Not started	Sprint 1
소셜 로그인 기능	● 인증 / 인가	카카오, 깃허브, 구글 소셜 로그인 기능을 통해 신규 사용자를 생성한다.	1 - most im...	● Not started	Sprint 2
꼬리질문 기능	● 프롬프트	입력한 답변에 대한 꼬리질문을 생성한다.	1 - most im...	● Not started	Sprint 2
면접관 포맷 생성 기능	● 면접관 프로필	(난이도, 수준, 키워드) 프롬프트 입력이 결정된 면접관 포맷을 생성한다.	2	● Not started	Sprint 2
AI 프로필 리스트	● 면접관 프로필	AI 면접관의 프로필을 생성한다.	2	● Not started	Sprint 2
AI 면접관 상세 페이지	● 면접관 프로필	면접관마다 질문 키워드 및 난이도 등을 제공한다.	2	● Not started	Sprint 2
면접 결과 저장 기능	● 면접 기능	면접 결과를 데이터베이스에 저장한다. (부족했던 점, 관련 키워드 등)	2	● Not started	Sprint 2
개인별 수준 측정 기능	● 면접 기능	사용자의 면접 답변을 토대로 수준(레벨)을 파악한다.	3	● Not started	Sprint 2
프로필 조회 기능	● 면접 기능	사용자 개인 별 프로필을 조회한다.	3	● Not started	Sprint 3
좋아요 기능	● 면접 기능	AI 면접관마다 좋아요를 달 수 있다.	3	● Not started	Sprint 3
공개 질문 리스트	● 면접 기능	공개된 질문 리스트를 보여준다.	3	● Not started	Sprint 3
사용자 면접 점수 기능	● 면접 기능	면접 후, 사용자의 답변 퀄리티를 토대로 점수를 계산한다. (공식 필요)	4	● Not started	Sprint 3
대화 횟수 제한 기능	● 면접 기능	한 면접에서 일정 횟수 이상 답변했을 시 종료한다.	5 - least im...	● Not started	Sprint4
▶ 면접 중간 정지 기능	● 면접 기능	면접을 중간에 정지할 수 있다.	5 - least im...	● Not started	Sprint4

# 웹 기능

## 시도 1 - 일정 관리

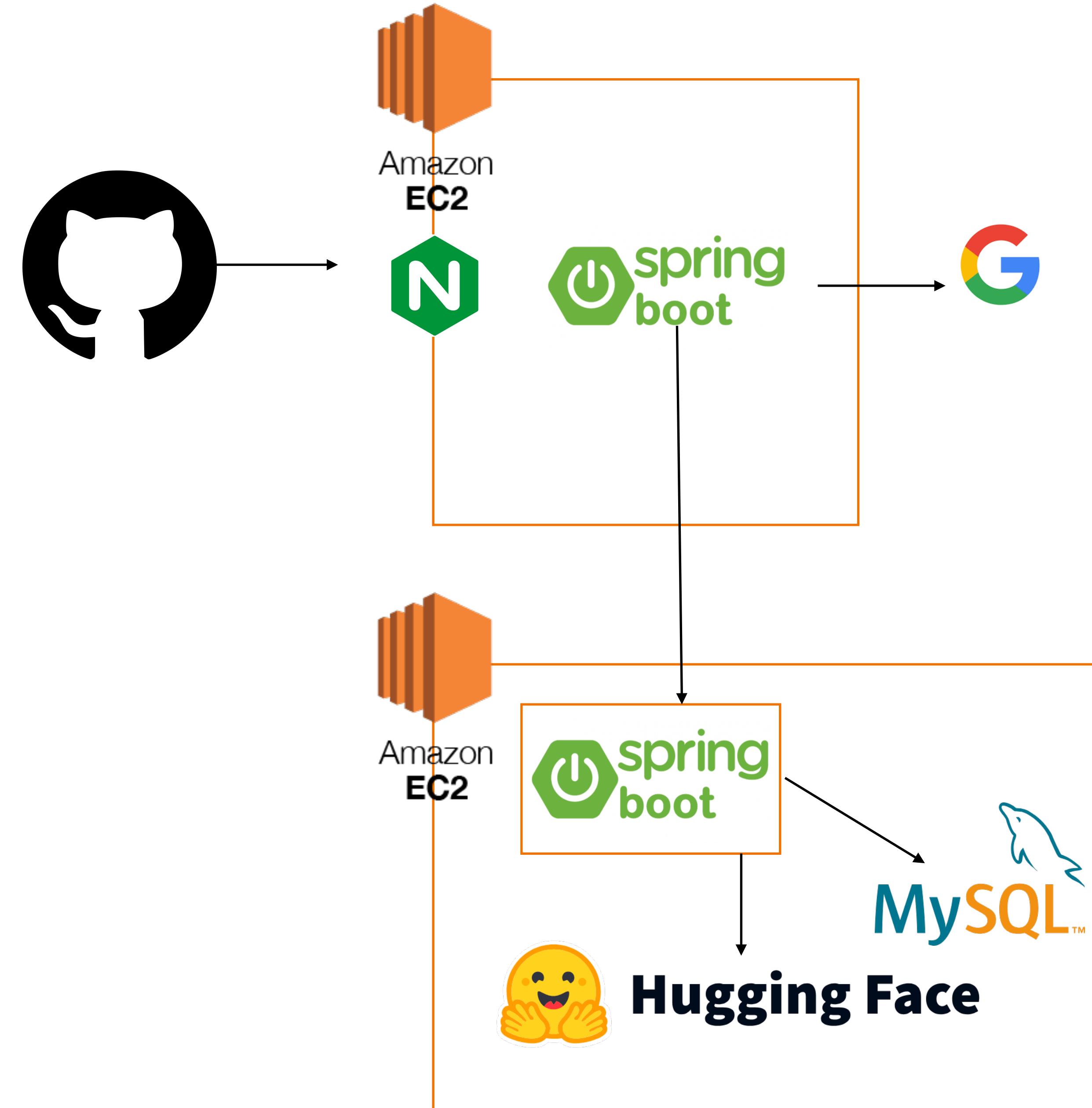
- 스프린트 단위 별 기능 구현
- 스프린트 내에는 해당하는 스프린트로 라벨링 된 기능을 구현하는 것이 목표

	Date
Sprint 1	2023.3.26 ~ 2023.4.8
Sprint 2	2023.4.9 ~ 2023.4.22
Sprint 3	2023.4.23 ~ 2023.5.6
Sprint 4	2023.5.7 ~ 2023.5.20
Buffer	2023.5.21 ~ 종강 전까지

# 웹 기능

## 시도 1 - 웹 개발 기술 선정

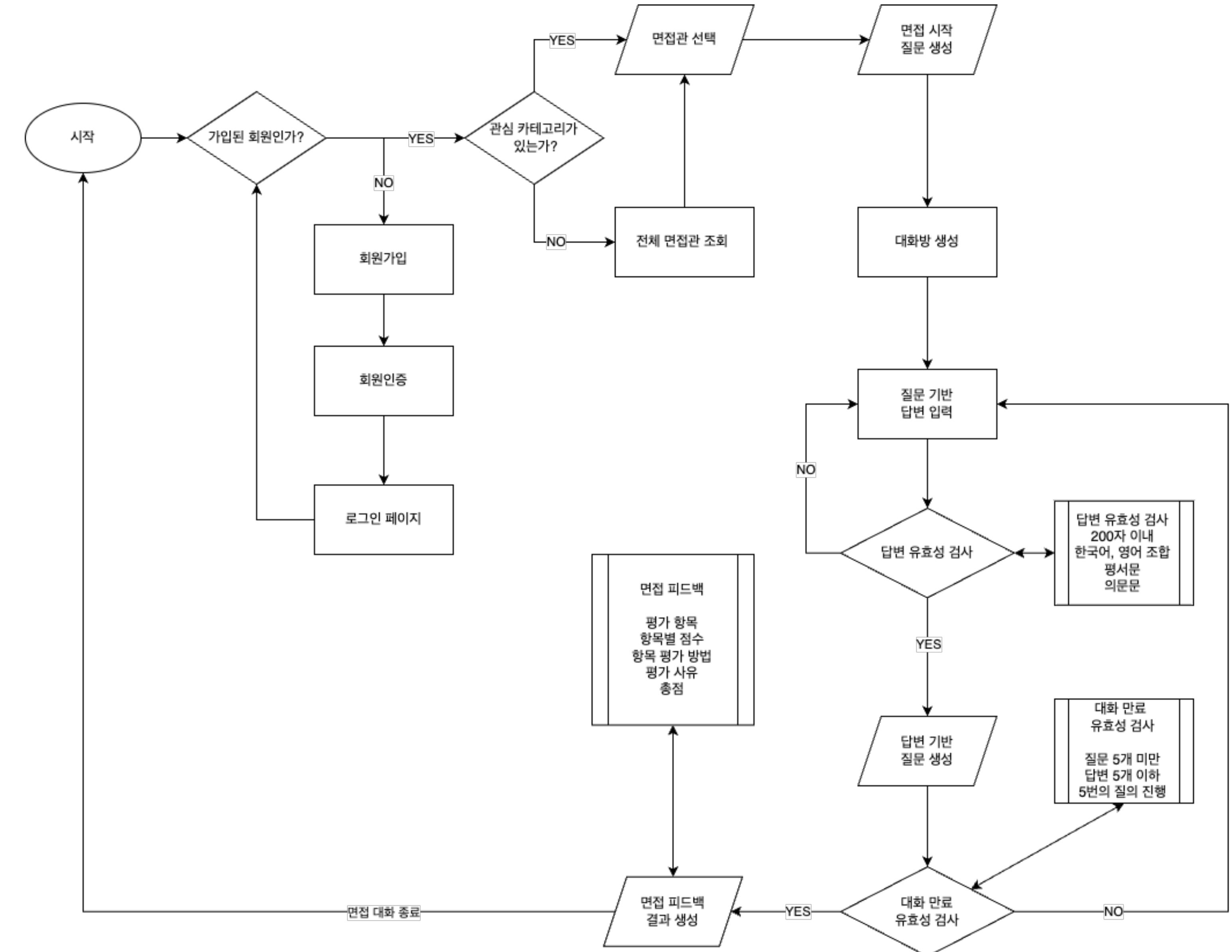
- Demo service for evaluate model deployed in Huggingface
- Vue.js + Spring Boot 3 based Web app
- Cloud web hosting supported on AWS
- Huggingface serverless API
  - HTTP request wrapped by user prompt
  - Available for adjusting response max\_token
  - Export text format data response



# 웹 기능

## 시도 2 - 흐름도 구성

- Core Feature를 기반으로 도식화한 Flow chart
- 이와 같은 Flow를 사용자가 체험할 수 있도록 하는 것이 최종 목표



# 웹 기능

## 시도 3 - AI 서버 구성

- Seraph 서버에서 Flask 기반으로 서버가 동작할 수 있도록 설정
- prompt 값과 함께 'POST' 요청이 오면, 모델 출력을 Response로 반환

```
[31]: from flask import Flask, request, jsonify
      from flask_cors import CORS
      from collections.abc import Sequence

[32]: app = Flask(__name__)
      CORS(app)

[32]: <flask_cors.extension.CORS at 0x7fcb27758410>

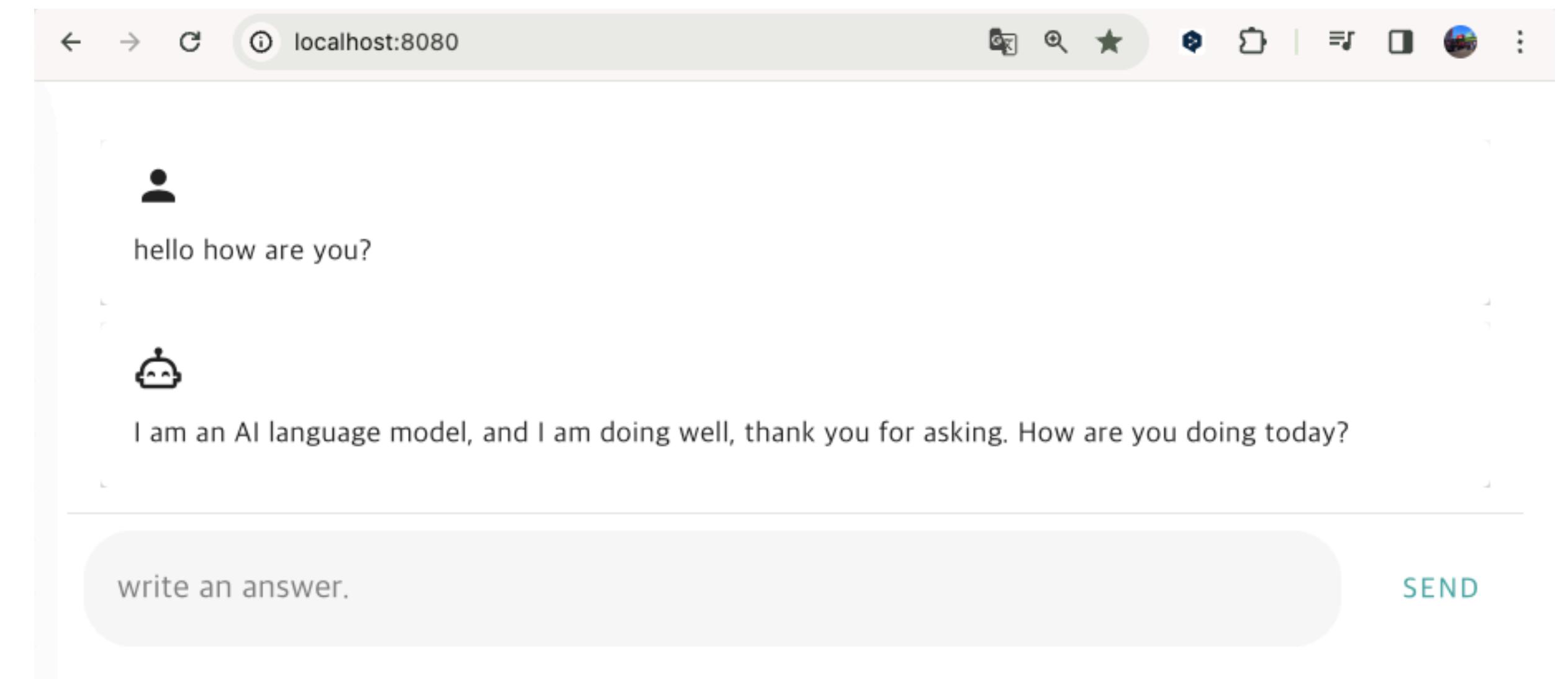
[33]: @app.route('/post', methods=['POST'])
      def post():
          params = request.get_json()
          prompt = params['prompt']
          response = generate_response(prompt)
          return jsonify({"data": response, "status": 200})

[ ]: app.run(host='163.180.160.41', port=5000)
```

# 웹 기능

## 시도 4 - 프론트엔드 개발

- Vue.js + Vuetify 기반 프론트엔드 개발
- 우측은 실제 프론트엔드 서버 구동 후  
작동 화면을 나타냄
- 면접 화면



# 웹 기능

## 시도 4 - 프론트엔드 개발

Hi there!

USAY helps you to improve job interview skills.

Click the button below to start your interview.

 START AN INTERVIEW

- Vue.js + Vuetify 기반 프론트엔드 개발
- 우측은 실제 프론트엔드 서버 구동 후  
작동 화면을 나타냄
- 온보딩 화면

Previous interview records can be found below.

LLM Interview

2024-01-01

 WATCH

Spring Boot Interview

2024-01-02

 WATCH

# 웹 기능

## 시도 4 - 프론트엔드 개발

- Vue.js + Vuetify 기반 프론트엔드 개발
- 우측은 실제 프론트엔드 서버 구동 후 작동 화면을 나타냄
- 기록 조회 화면

### Spring Boot Interview

2024-01-02



백엔드 서버에서 사용하는 주요 프로그래밍 언어는 무엇이 있나요?



Java나 Python이 있습니다.



좋아요, Java와 Python이 백엔드 서버 개발에서 많이 사용되는 언어 중 하나라는 건 알고 있어요. 그렇다면 이 두 언어 중에서 어떤 기준으로 선택을 하게 되는지 알려주실 수 있나요?



Python의 경우 인터프리터 언어이며, 배우기 쉽다는 장점이 있습니다. 따라서 과학이나 머신러닝 등에서 주로 활용됩니다. Java는 대규모 분산 어플리케이션을 만들기 위해 주로 활용되며, Spring Boot라는 프레임워크를 사용해 웹 어플리케이션을 개발할 수 있습니다.

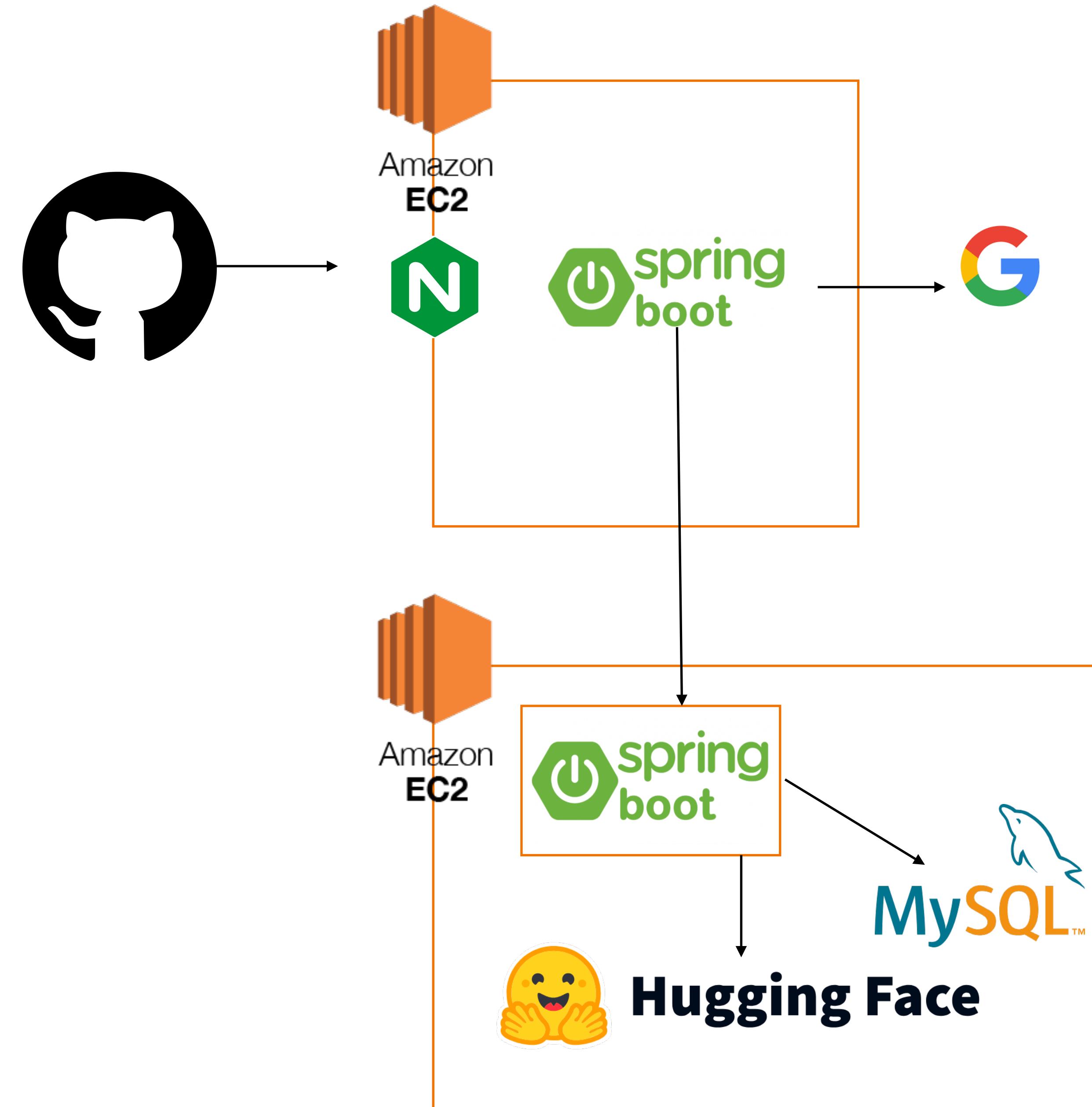


CLOSE

# 웹 기능

## 시도 5 - 백엔드 개발

- AWS Free-tier VPC 구성
  - Public subnet (현재는 API만 배포  
-> Frontend에서 사용가능)
  - Private subnet (채팅 DB, HuggingFace Model 연결 서버)
- 대화 단건 및 사용자 조회 정적 데이터 처리
  - Nginx 서버 구성 -> 사용자 가입 처리 별도 구현
- 페이지 접근 시 OAuth 가입 여부 확인  
Security 접근 제한 필터링 구현



# 웹 기능

## 시도 5 - 백엔드 개발

- Swagger 3 기반 API 명세 기능
  - 웹페이지 개발 (Vue3.js 기반) 사용 및 테스트 가능한 명세환경 구현
  - 입력 검증 및 예외 케이스 제공
  - 예외 케이스 : 별도 비즈니스 로직구성  
-> 대화 기능에 맞는 에러 트래킹
  - AWS EC2 배포 -> 도메인 접근 가능

The screenshot shows the Swagger UI interface for the 'Usay API v1'. At the top, there's a navigation bar with the title 'Usay API v1 OAS3' and a dropdown menu 'Select a definition' set to 'Usay API v1'. Below the header, the main content area displays the API documentation. It includes sections for 'Servers' (set to 'http://localhost:8080 - Generated server url'), '답변' (User Response API), 'room-controller', and 'member-controller'. Under '답변', there are three operations: 'POST /chat/{memberId}' (채팅방 채팅 등록), 'GET /room/{roomId}' (대화방 별 전체 채팅 조회), and 'GET /feeds/{memberId}' (사용자별 답변 조회). Under 'room-controller', there is a single operation 'POST /'. Under 'member-controller', there is also a single operation 'POST /chat/{memberId}'. To the right of the main content, there is a large code block showing Java code for handling errors. The code defines several error classes: `INVALID`, `INVALID_INPUT_VALUE`, `INVALID_AI_MODEL`, `INVALID_AI_MODEL_VERSION`, `NOT_FOUND`, `NOT_FOUND_AI_MODEL`, `INTERNAL_SERVER_ERROR`, and `private final String code;`. Below this, there are sections for '14 usages' and a code snippet for a `@Contract(pure = true)` method. On the far right, there are four panels showing examples of API responses for different error codes: 400 (Bad Request), 404 (Not Found), 500 (Internal Server Error), and 401 (Unauthorized).

```
// 400 Bad Request
INVALID( code: "SAY400", message: "Invalid Request"),
INVALID_INPUT_VALUE( code: "SAY000", message: "Invalid Input Value"),
INVALID_AI_MODEL( code: "SAY001", message: "Invalid AI Model"),
INVALID_AI_MODEL_VERSION( code: "SAY002", message: "Invalid AI Model Version"),

// 404 Not Found
NOT_FOUND( code: "SAY404", message: "Not Found"),
NOT_FOUND_AI_MODEL( code: "SAY400", message: "AI Model Not Found"),

// 500 Internal Server Error
INTERNAL_SERVER_ERROR( code: "SAY500", message: "Server Error");
// 401 Unauthorized
private final String code;
private final String message;

14 usages
@Contract(pure = true)
ErrorCode(String code, String message) {
    this.code = code;
    this.message = message;
}

POST /chat/{memberId} 채팅방 채팅 등록
GET /room/{roomId} 대화방 별 전체 채팅 조회
GET /feeds/{memberId} 사용자별 답변 조회

room-controller
POST /

member-controller
POST /chat/{memberId} 채팅방 채팅 등록
GET /room/{roomId} 대화방 별 전체 채팅 조회

400
Bad Request
Media type
application/json
Example Value | Schema
{
  "resultCode": "string",
  "message": "string",
  "data": {}
}

404
Not Found
Media type
application/json
Example Value | Schema
{
  "resultCode": "string",
  "message": "string",
  "data": {}
}

500
Internal Server Error
Media type
application/json
Example Value | Schema
{
  "resultCode": "string",
  "message": "string",
  "data": {}
}

401
Unauthorized
Media type
application/json
Example Value | Schema
{
  "resultCode": "string",
  "message": "string",
  "data": {}
}
```

# 웹 기능

## 시도 5 - 백엔드 개발

- 도메인 구매 예정 -> 공개 배포 계획
  - 데이터셋 오픈소스화 -> Huggingface 및 Github 배포
  - Demo 겸하여 면접 페이지 배포
- 주요 기능 구현
  - 면접 최대 횟수
  - 인증 (로그인, 사용자 조회)
  - Huggingface 모델 연결 후 테스트

Swagger Supported by SMARTBEAR

Select a definition Usay API v1

Usay API v1 OAS3

/v3/api-docs/Usay API v1

SwaggerTest API Docs

Servers http://localhost:8080 - Generated server url

답변 사용자 답변 관련 API

POST /chat/{memberId} 채팅방 채팅 등록

GET /room/{roomId} 대화방 별 전체 채팅 조회

GET /feeds/{memberId} 사용자별 답변 조회

room-controller

POST /

member-controller

```
// 400 Bad Request
INVALID( code: "SAY400", message: "Invalid Request"),
INVALID_INPUT_VALUE( code: "SAY000", message: "Invalid Input Value"),
INVALID_AI_MODEL( code: "SAY001", message: "Invalid AI Model"),
INVALID_AI_MODEL_VERSION( code: "SAY002", message: "Invalid AI Model Version"),

// 404 Not Found
NOT_FOUND( code: "SAY404", message: "Not Found"),
NOT_FOUND_AI_MODEL( code: "SAY400", message: "AI Model Not Found"),

// 500 Internal Server Error
INTERNAL_SERVER_ERROR( code: "SAY500", message: "Server Error");
// 401 Unauthorized
private final String code;
private final String message;

14 usages
@Contract(pure = true)
ErrorCode(String code, String message) {
    this.code = code;
    this.message = message;
}
```

400 Bad Request

Media type application/json

Example Value Schema

```
{
  "resultCode": "string",
  "message": "string",
  "data": {}
}
```

404 Not Found

Media type application/json

Example Value Schema

```
{
  "resultCode": "string",
  "message": "string",
  "data": {}
}
```

500 Internal Server Error

Media type application/json

Example Value Schema

```
{
  "resultCode": "string",
  "message": "string",
  "data": {}
}
```

# Usay

## 최종 제품, 아쉬운 점, 미래 계획

- 사용자 답변 평가 기능이 최종적으로 구현되었다.
- GPU Resource 상향을 캡스톤디자인이 끝날 때까지 지원 받지 못해 시도하지 못한 것들이 많았다. (QLoRA 없이 SFT, Tokenizer Extension, Continuous Pretraining)
- 미래에는 Model의 성능을 더 향상시키고, 완벽하게 구현하지 못한 기능들을 보완하고 싶다.

