



A.I. Doctor

캡스톤 12조

장민혁, 조상연, 황교민, 허진우

국민대학교 소프트웨어융합대학 소프트웨어융합과



Contents

프로젝트 소개

프로젝트 목표

시스템 구성

시연 영상

부록

프로젝트 소개

우울



지속적인 우울한 기분은 곧 우울증을 의미합니다.

2주 이상 지속되는 우울한 기분

갑작스러운 체중 및 식욕의 변화

불면과 과수면

무기력감과 죄책감

집중력의 감소와 결정 곤란

죽음에 대한 반복적인 생각

* DSM-V 진단기준(주요 우울증 삽화의 진단 기준)

프로젝트 목표 ✓



목표

궁극적으로 기존 모니터링 시스템에 기능 모듈로 설치되는 것



Baseline 보다 정확하게
사용자와의 대화에서 우울함을 탐지



2주간의 통계를 분석하고
이를 보호자나 보호 기관에 알리기

시스템 구성



팀원 소개

프로젝트를 함께한 동료들.



장민혁

Backend
IoT Device



조상연

A.I. Model
Backend



허진우

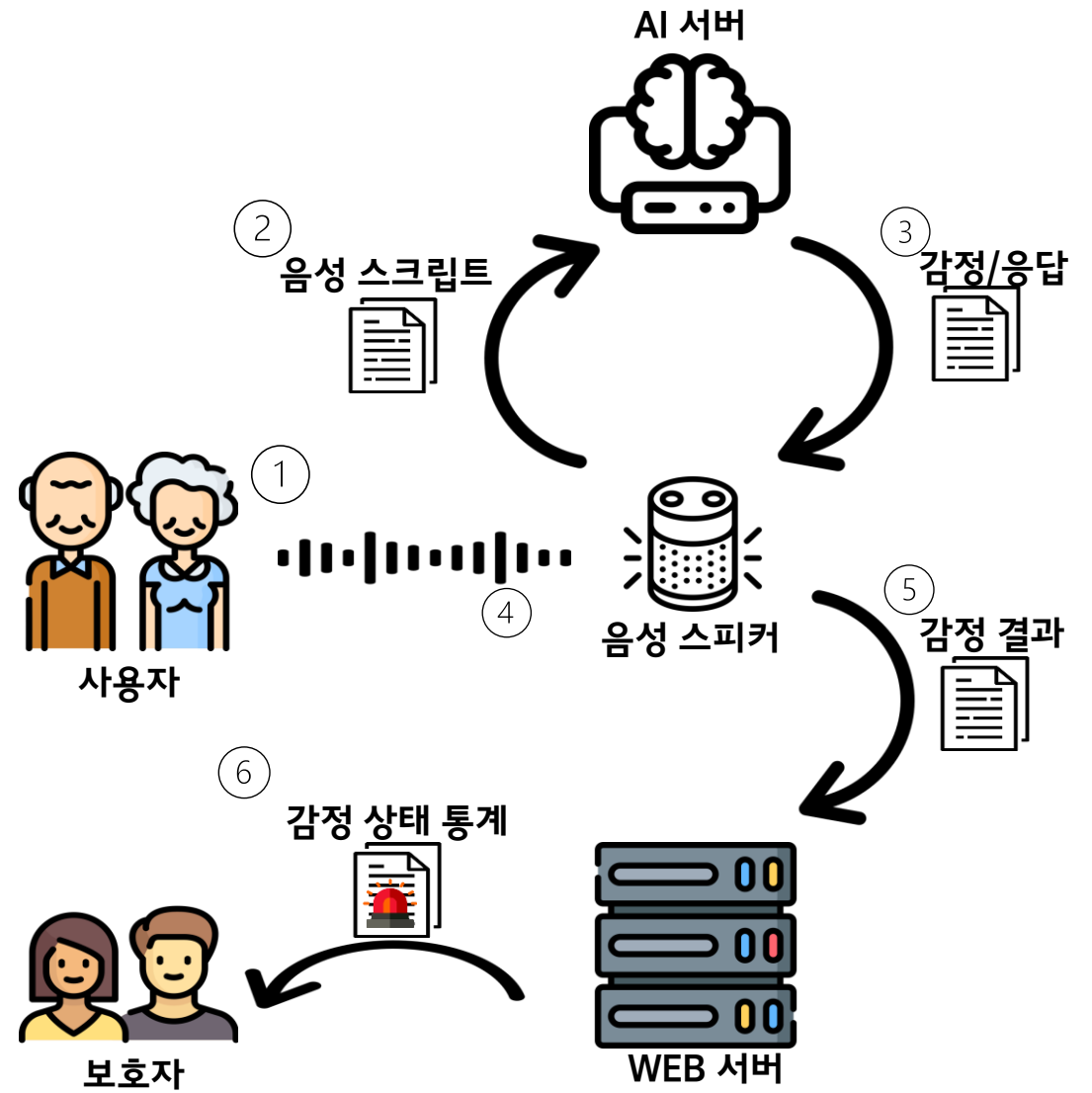
A.I. API
Frontend



황교민

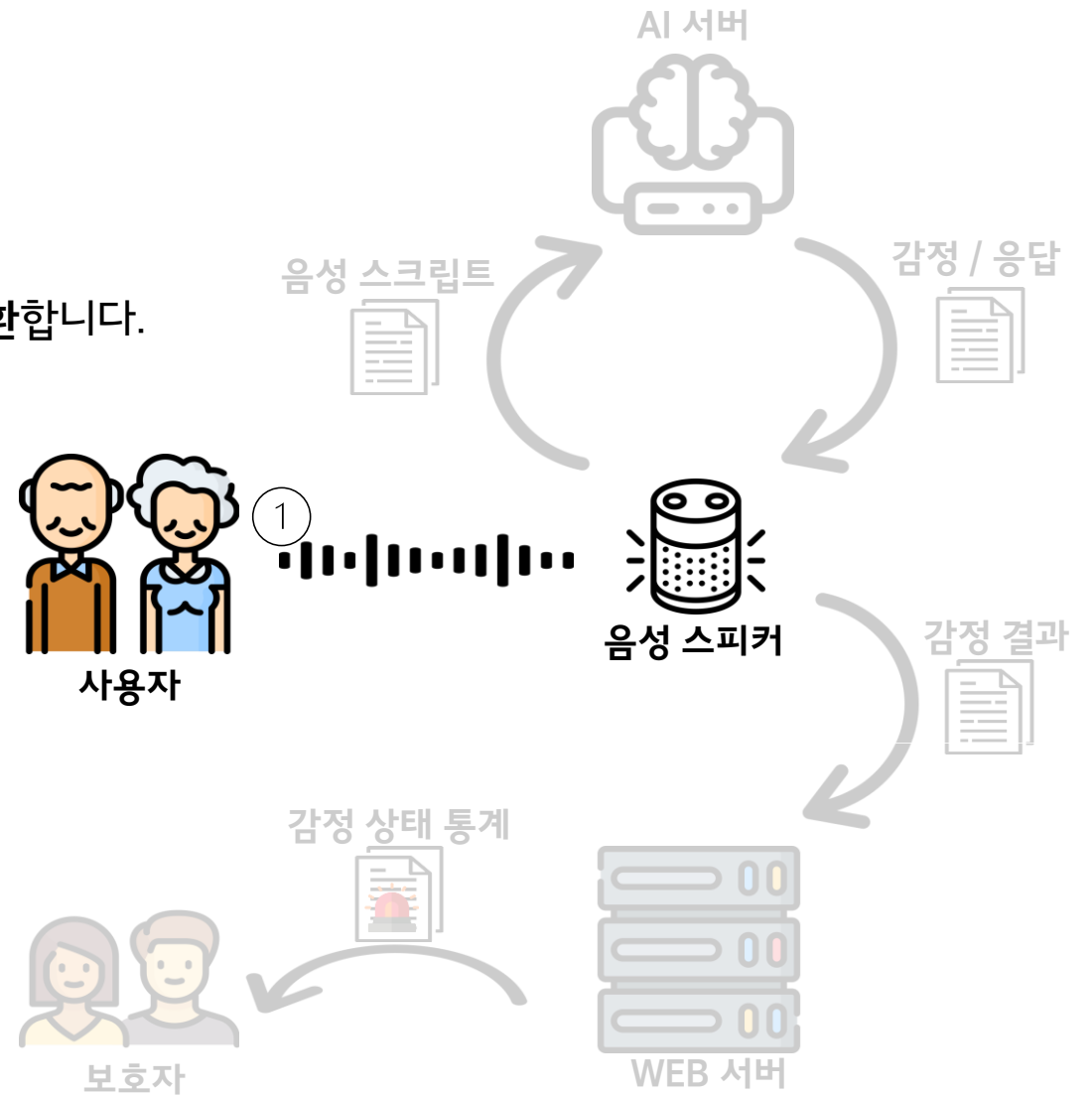
A.I. Model
IoT Device

시스템 흐름



사용자 음성 받기

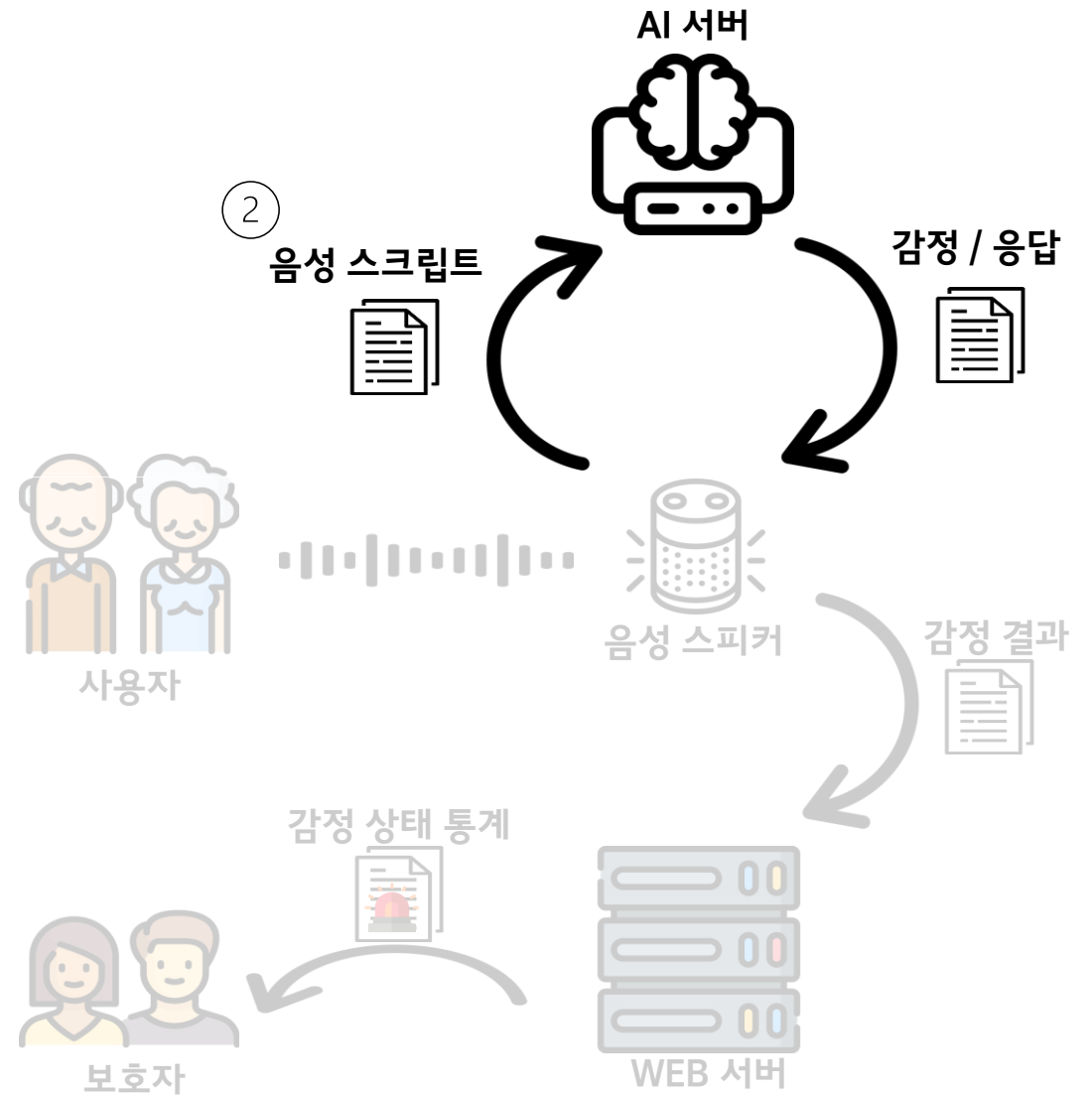
마이크로부터 받은 사용자의 음성을 API를 통해 텍스트로 변환합니다.



AI 서버

AI 서버로 텍스트와 음성 데이터를 전달합니다.

AI 서버에서는 이를 통해 감정 분류와 적절한
응답을 생성합니다.

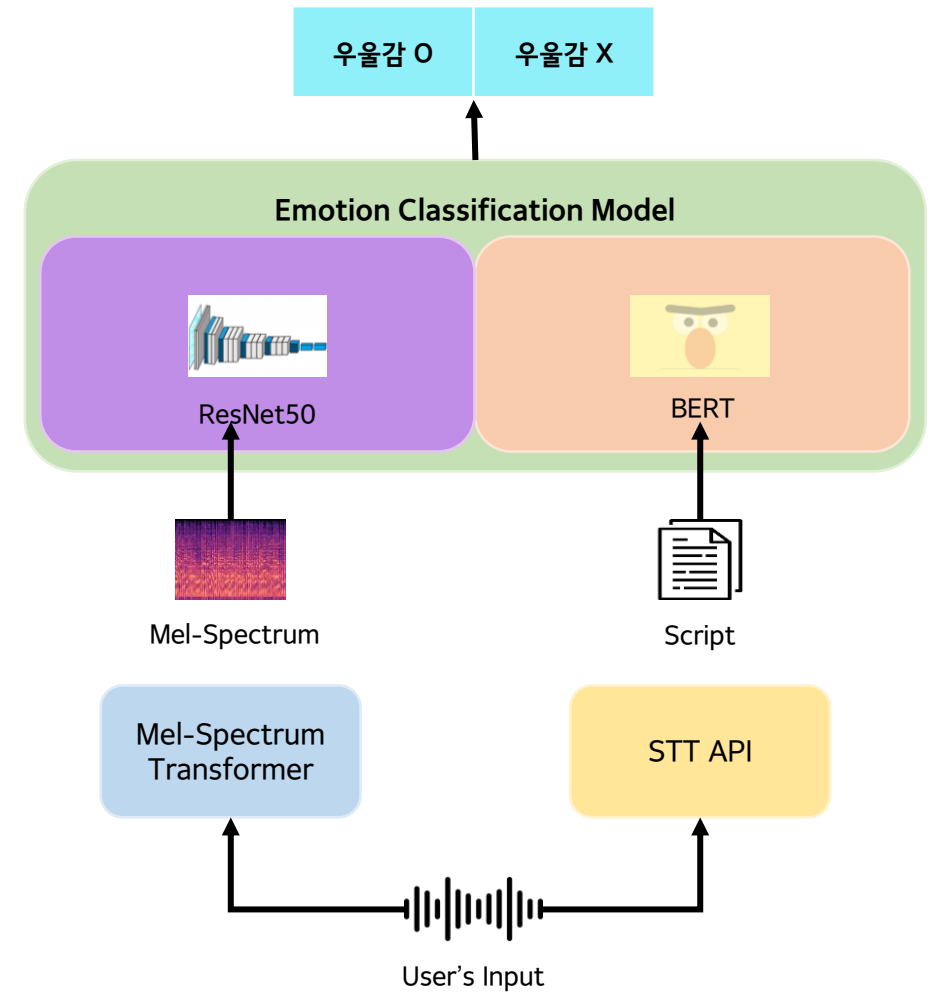


감정 분류 모델

목적에 맞게 우울한 기분 감지에 집중

억양을 반영할 수 있도록 음성 데이터도 함께 사용

AI HUB 감성 대화 말뭉치 데이터 사용

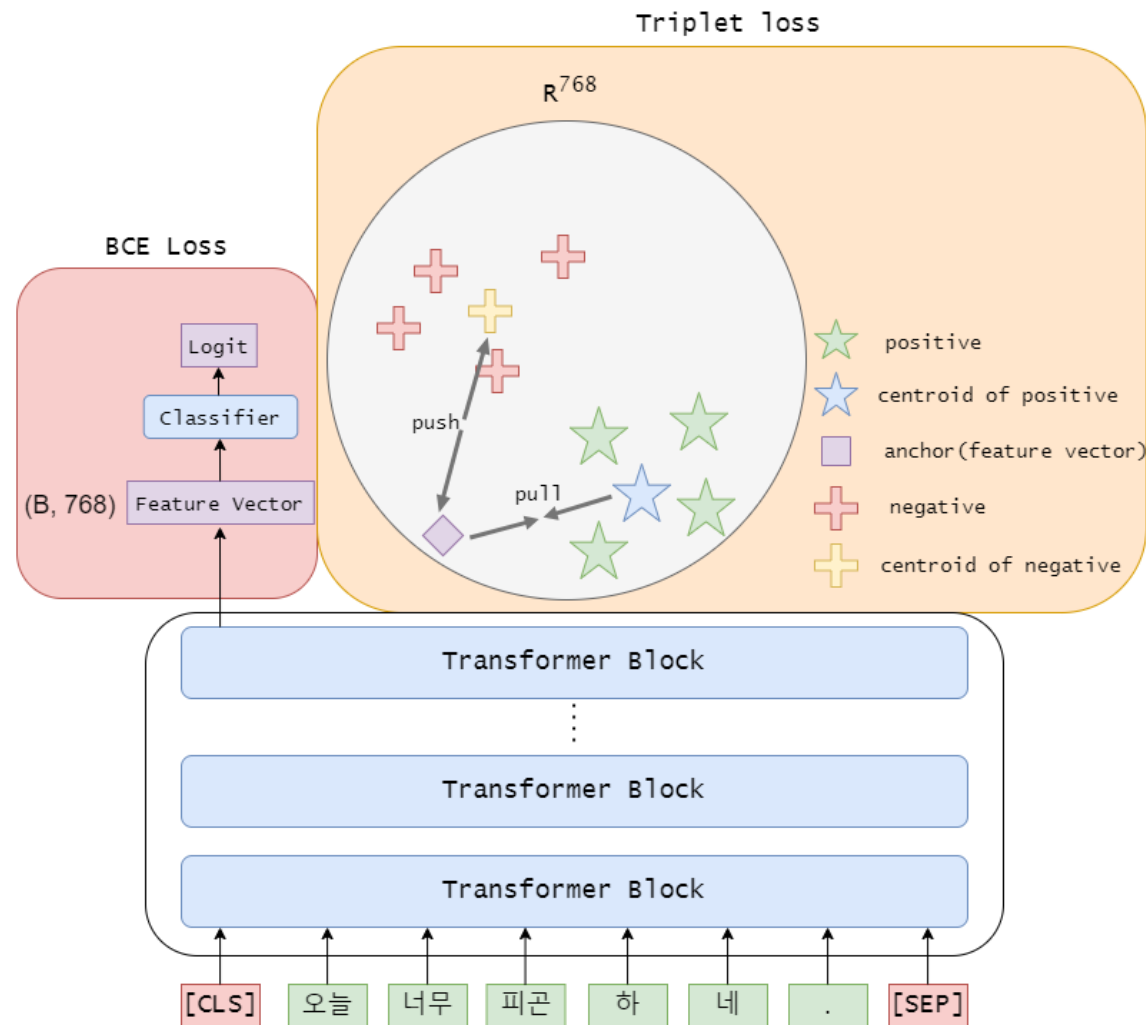


감정 분류 모델

Inference 시간을 늘리지 않으면서 효율적으로 규제를 줄 수 있는 방법으로 triplet loss를 사용합니다.

$Loss_{total}$

$$= Loss_{BCE}(logit, label) \\ + \lambda Loss_{Triplet}(anc, pos, neg, \alpha) \\ (\lambda = 0.05)$$



감정 분류 모델 성능

	Accuracy	Precision	Recall	F1
BERT(Text)	0.86	0.70	0.76	0.73
BERT+Triplet(Text)	0.87	0.75	0.72	0.73
ResNet50(Voice)	0.86	0.72	0.72	0.72
Fusion(Text, Voice)	0.93	0.88	0.84	0.86

감정 분류 예시



기존 생성모델 - 문제점

기존의 챗봇 모델은 KoGpt-2를 통한 생성모델

평이하고 부적절한 응답을 출력하는 문제 발생

해당 문제를 해결하기 위한 방안 모색

GPT-2 챗봇

```
USER >>> 오늘 시험을 망쳤어  
BOT >>> 오늘 시험을 망쳐서 속상하시군요.  
USER >>> 어제 발 다쳤어  
BOT >>> 병원가세요.  
USER >>> 이제 놀이공원 가자!  
BOT >>> 놀이공원 가세요.  
USER >>> exit
```

평이한 응답

```
USER >>> 안녕  
BOT >>> 안녕하세요.  
USER >>> 내일은 야근하는데,,  
BOT >>> 오늘도 야근하세요.  
USER >>> 내일 뭐 먹을까  
BOT >>> 냉장고 파먹기 해보세요.  
USER >>> exit
```

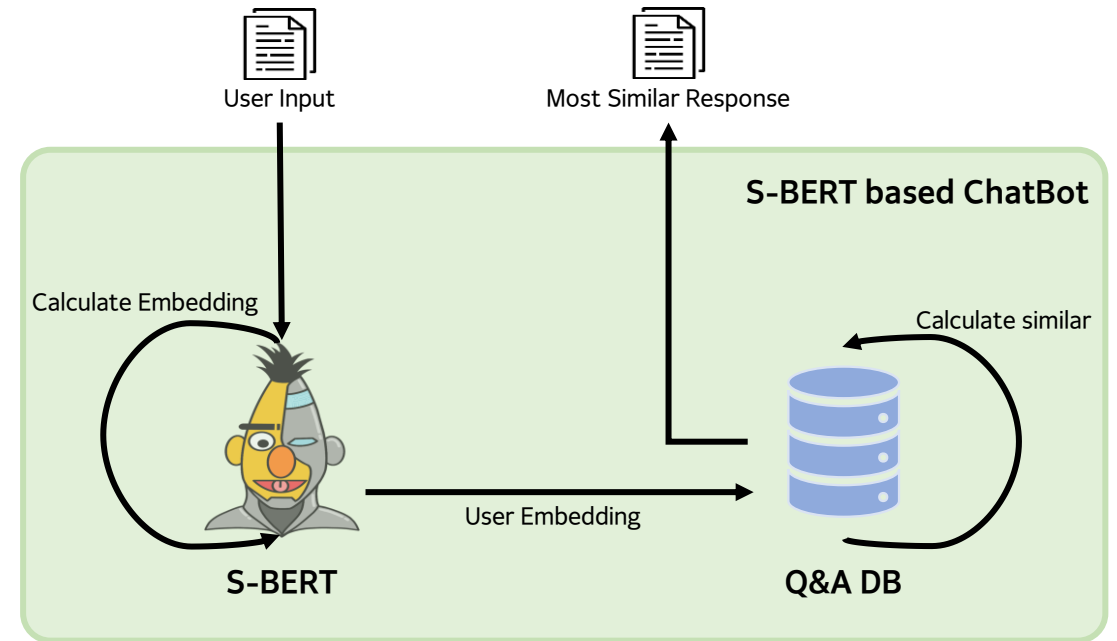
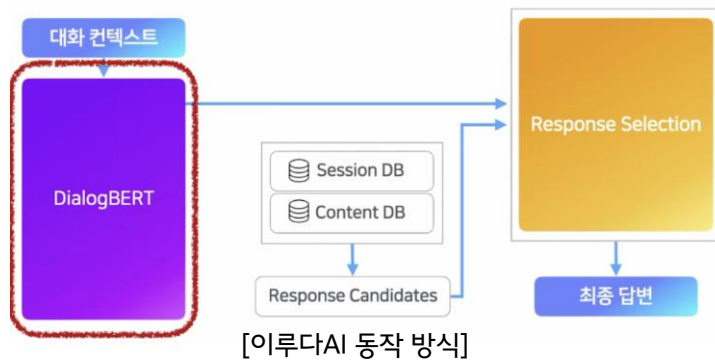
부적절한 응답

문장 출력 모델 - 구조

이루다AI의 동작 방식에서 차용

개선하는 모델의 구조는 정제된 응답 데이터가 담긴 DB에서
적절한 응답을 선택하는 방식

문장의 유사도를 통한 검색기반의 챗봇

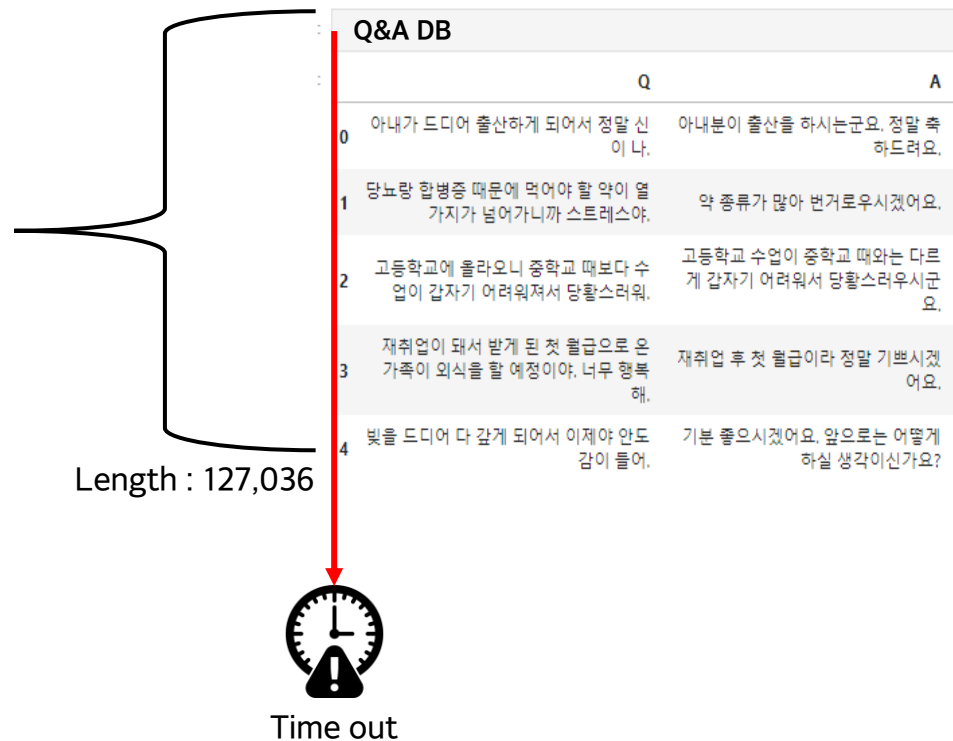


문장 출력 모델 - 문제점

제시하는 모델은 큰 문제점이 존재

DB에 저장된 12만 가량의 데이터를 모두 탐색하여 적절한 문장을 찾는것에 많은 시간이 소요됨

출력시간 단축에 대한 노력

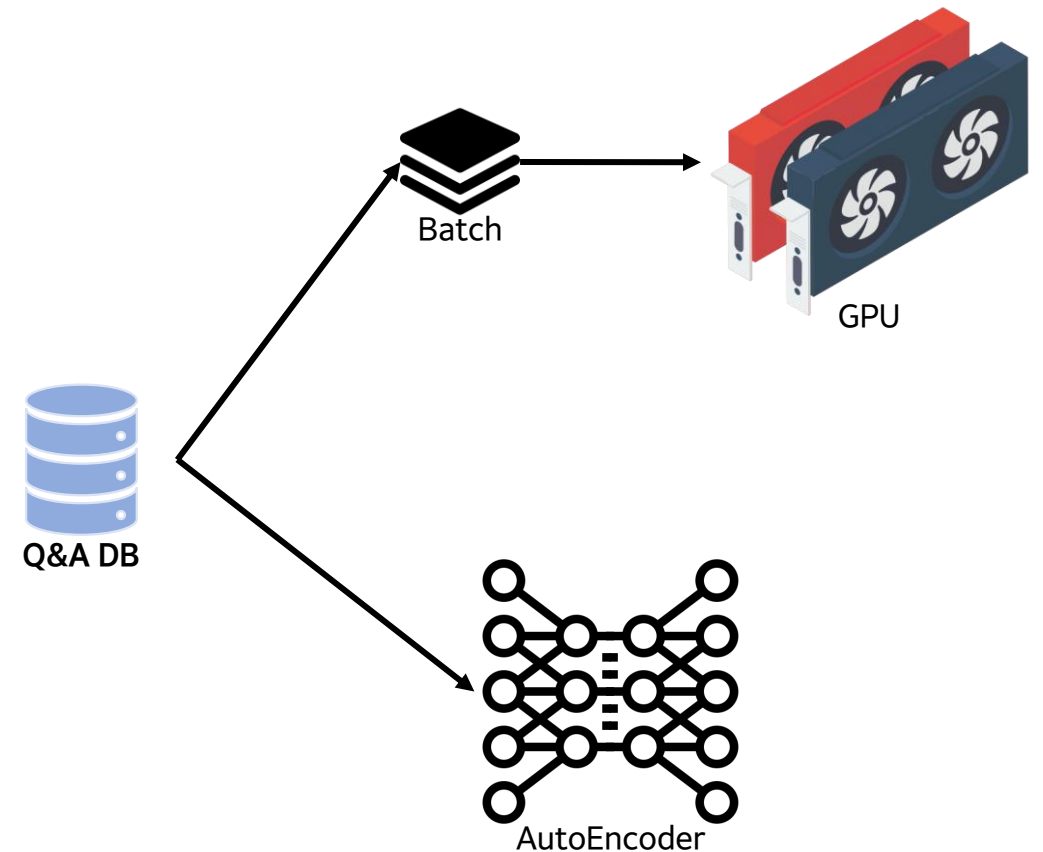


문장 출력 모델 - 개선

2가지 원인을 파악

1. 약 12만개의 많은 데이터
2. 유사도 연산하기에 큰 Embedding 차원

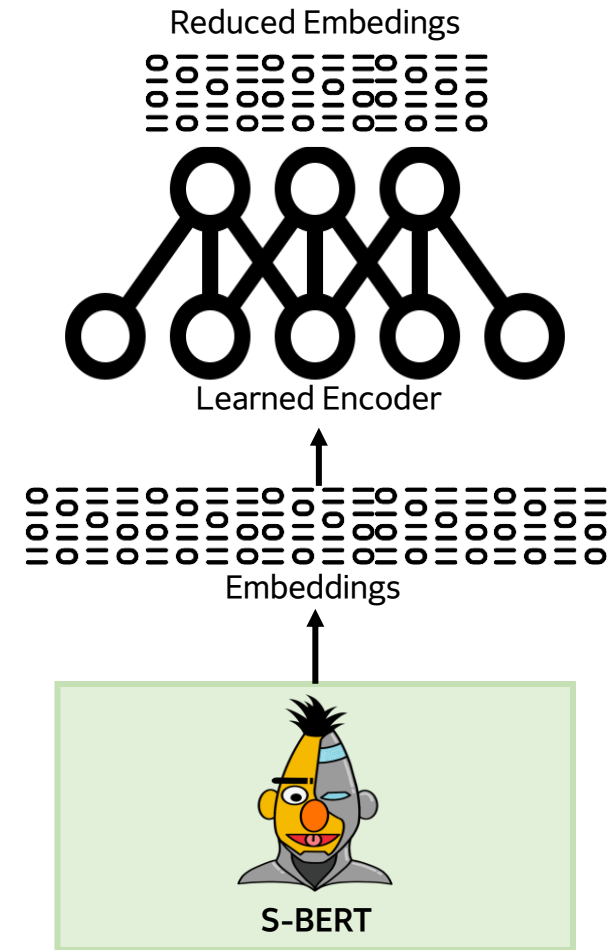
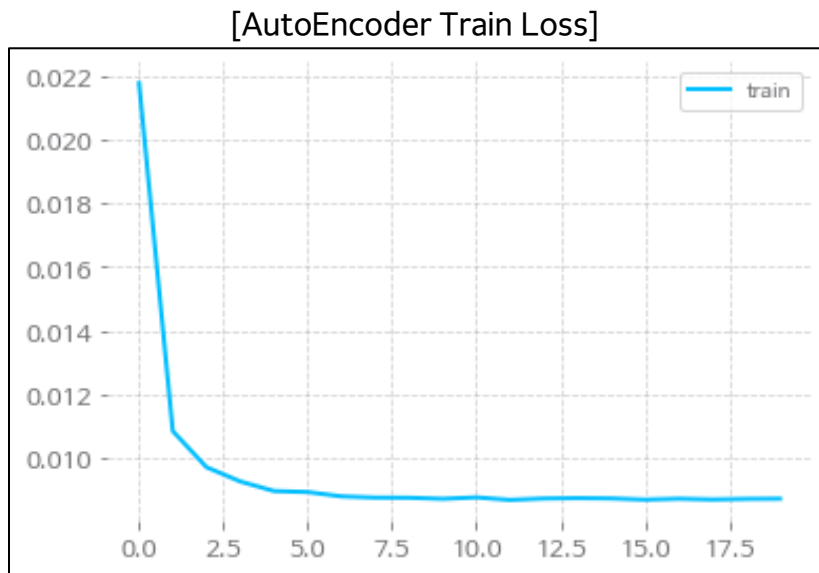
데이터는 Batch로 만들어 GPU에서 더욱 빠르게 연산하도록 설계
768차원의 Embedding은 AutoEncoder로 적절한 차원으로 축소



문장 출력 모델 - 개선

AutoEncoder로 축소한 차원은 각각 256, 128, 64, 32으로

AutoEncoder를 충분히 학습시킨 후 Encoder를 S-BERT 최상위 Layer에 이어 붙여 축소된 Embedding 차원으로 변환



문장 출력 모델 - 성능

	Response Time	Note.
GPT-2	within a second	기존 모델
S-BERT _(CPU) Basis	40 ~ 70 second	PC 성능에 따라 차이폭이 심함
S-BERT _(GPU) Batch_size : 512 Dimension : 768	3 second	GPU를 사용한 가장 기본 Model
S-BERT _(GPU) Batch_size : 512 Dimension : 256	2 second	768 → 256 Dimension
S-BERT _(GPU) Batch_size : 512 Dimension : 128	1.6 second	768 → 128 Dimension
S-BERT _(GPU) Batch_size : 512 Dimension : 64	1.4 second	768 → 64 Dimension 정보 손실과 응답시간의 Trade off를 고려하여 선택
S-BERT _(GPU) Batch_size : 512 Dimension : 32	1.2 second	768 → 32 Dimension 정보 손실의 부담이 크다고 판단

문장 출력 모델 - 예시

문장 출력 모델 예시(상) - 개선 모델

문장 생성 모델 예시(하) - 기존 모델

생성모델에 비해 출력모델의 결과가 더 자연스러운 것을 확인

출력모델의 출력시간문제는 앞선 노력으로 최대한 극복

문장 출력 모델(S-BERT)

```
HONEYCAM.ORG
USER >>> 안녕하세요?
BOT >>> 안녕하세요.
USER >>>
```

```
HONEYCAM.ORG
USER >>> 안녕하세요?
BOT >>> 안녕하세요.
USER >>>
```

```
C:\Anacondas\python.exe C:/Users/JoSangYeon/Pyth
USER >>> 안녕
BOT >>> 안녕하세요.
USER >>> 내일은 야근하는데...
BOT >>> 오늘 밀린 업무와 내일 업무까지 걱정이 되는군요.
USER >>> 뭐 하지 내일?
BOT >>> 맛있는 거 드세요.
USER >>> exit
```

```
C:\Anacondas\python.exe C:/Users/JoSangYeon/Pyth
USER >>> 오늘 시험을 망쳤어
BOT >>> 시험 결과가 좋지 않으셨나 봐요.
USER >>> 어제 발 다쳤어
BOT >>> 맥주병 같은 걸로 살살 문질러 주세요
USER >>> 이제 놀이공원 가자!
BOT >>> 데이트하기 좋은 곳이죠.
USER >>> exit
```

문장 생성 모델(GPT-2)

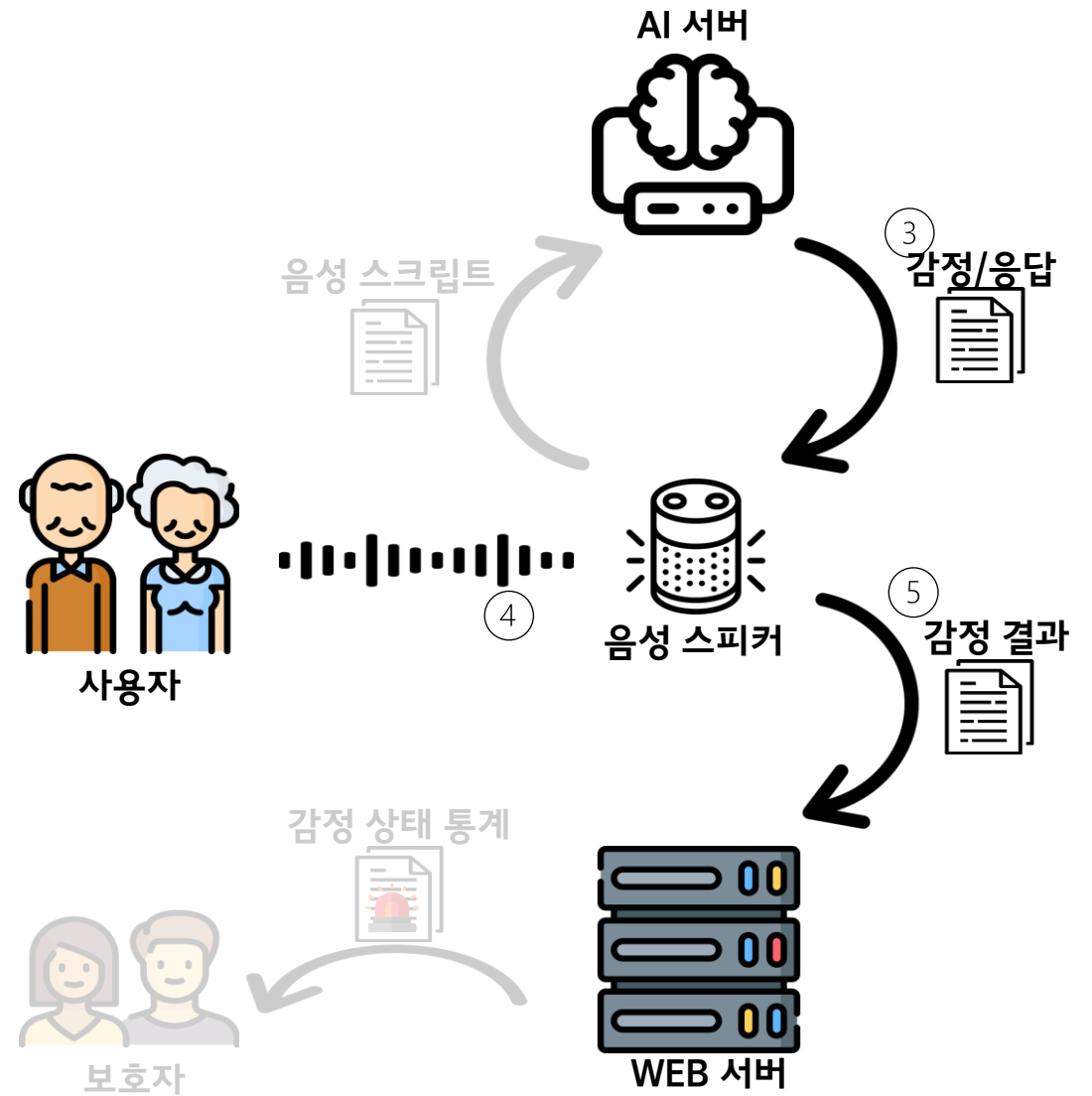
```
USER >>> 안녕
BOT >>> 안녕하세요.
USER >>> 내일은 야근하는데...
BOT >>> 오늘도 야근하세요.
USER >>> 내일 뭐 먹을까
BOT >>> 냉장고 파먹기 해보세요.
USER >>> exit
```

```
USER >>> 오늘 시험을 망쳤어
BOT >>> 오늘 시험을 망쳐서 속상하시군요.
USER >>> 어제 발 다쳤어
BOT >>> 병원 가세요.
USER >>> 이제 놀이공원 가자!
BOT >>> 놀이공원 가세요.
USER >>> exit
```

음성 합성

AI 서버로부터 받은 응답을 gTTS 라이브러리를 통해서 음성합성을 합니다.

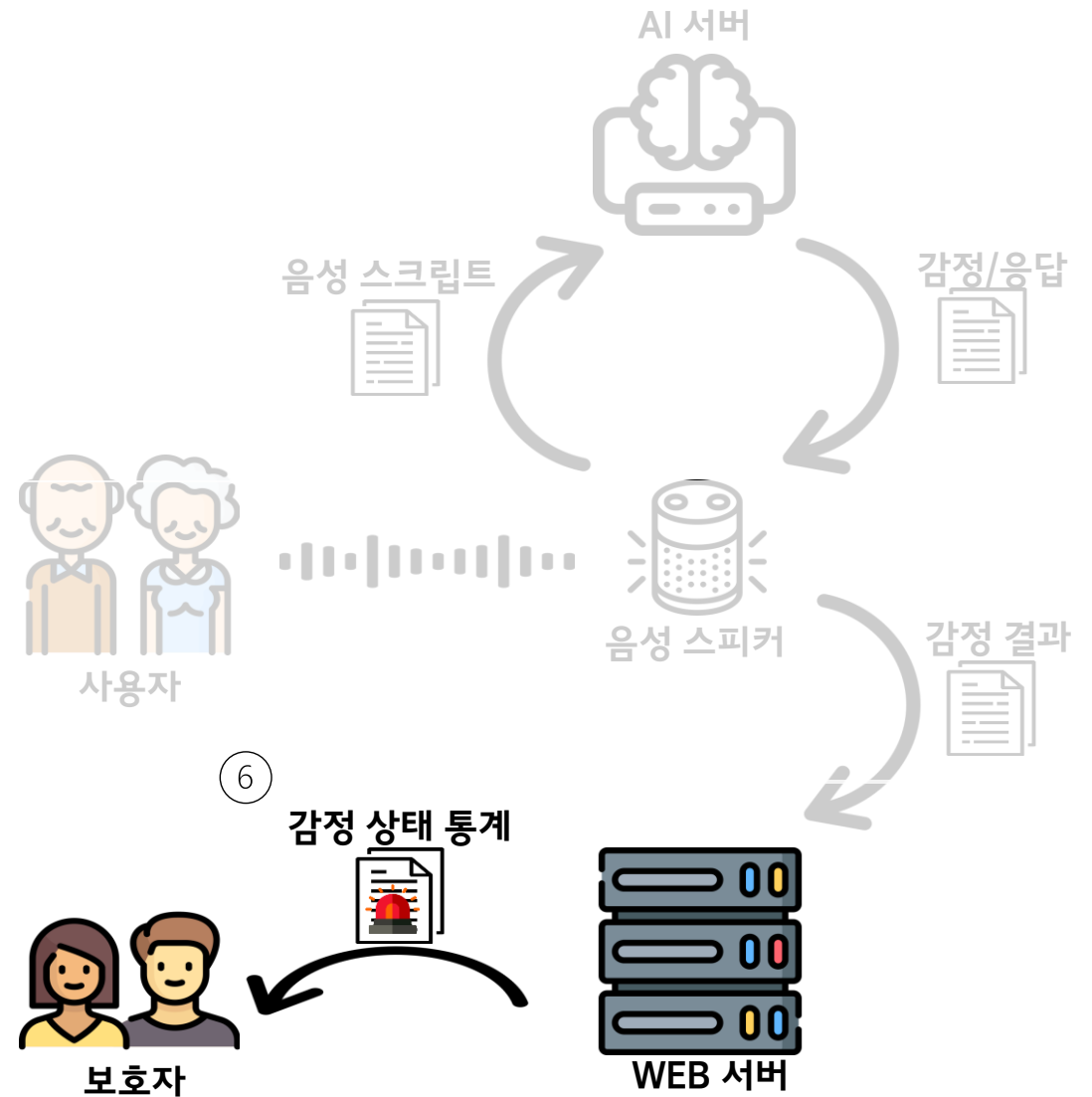
AI 서버로부터 받은 결과를 WEB서버로 전송해 DB에 저장합니다.



모니터링 서비스

2주간의 우울감 비율을 계산한 후 특정 기준이 넘어가면
보호자/기관에게 메일을 통해서 알림을 줍니다.


이를 통해서 보호자와 기관은 사용자에게 적절한 조치를
취할 수 있게 됩니다.



모니터링 서비스 - 예시

로그인 화면

AID에 오신 것을 환영합니다.



로그인

☒ 사용자 ☐ 보호자

아이디 *

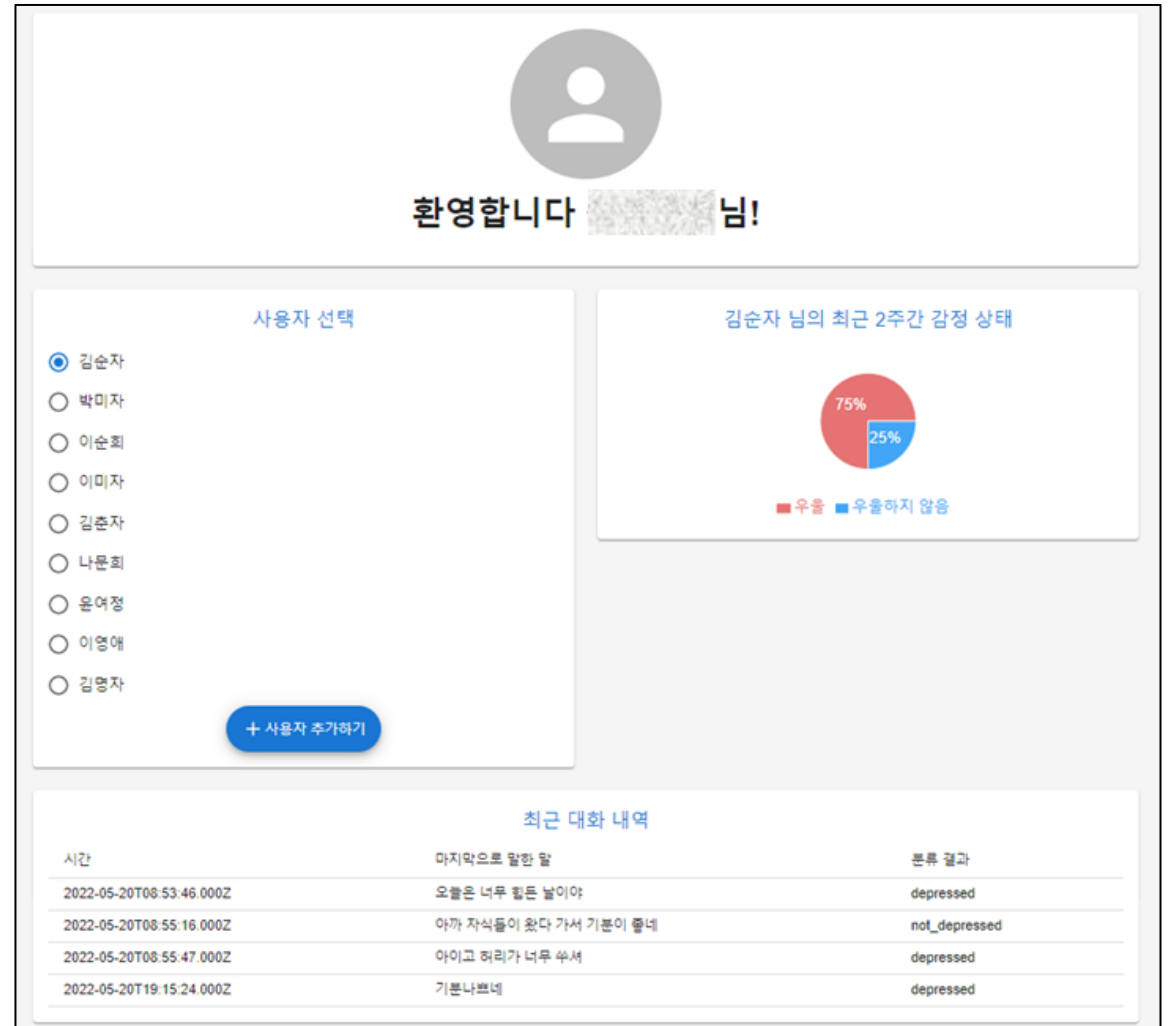
비밀번호 *

로그인하기

[아직 회원이 아니신가요?](#)


모니터링 서비스 - 예시

대화내역 & 감정상태 분석 결과



모니터링 서비스 - 예시

판단 결과 보고서 메일 발송

☆ AID 우울감 판단 결과 

▲ 보낸사람 **VIP** <kookminaid17@gmail.com>

받는사람 <wkdalsgur85@naver.com>

윤여정님의 우울감 위험도는 66%로, 위험합니다!
윤여정님께 방문하시거나 가까운 병원으로 안내해주세요.

지난 2주간의 대화 내역입니다.

- 오늘 기분이 꿀꿀하네
- 요즘 허리가 너무 아프네
- 오늘 우리 영감이 퇴원했어



시연 영상



“ 정신적으로 건강한 사회를 위한 한 걸음...”

감사합니다. 

이상으로 캡스톤 12조의 발표를 마치겠습니다.

부록

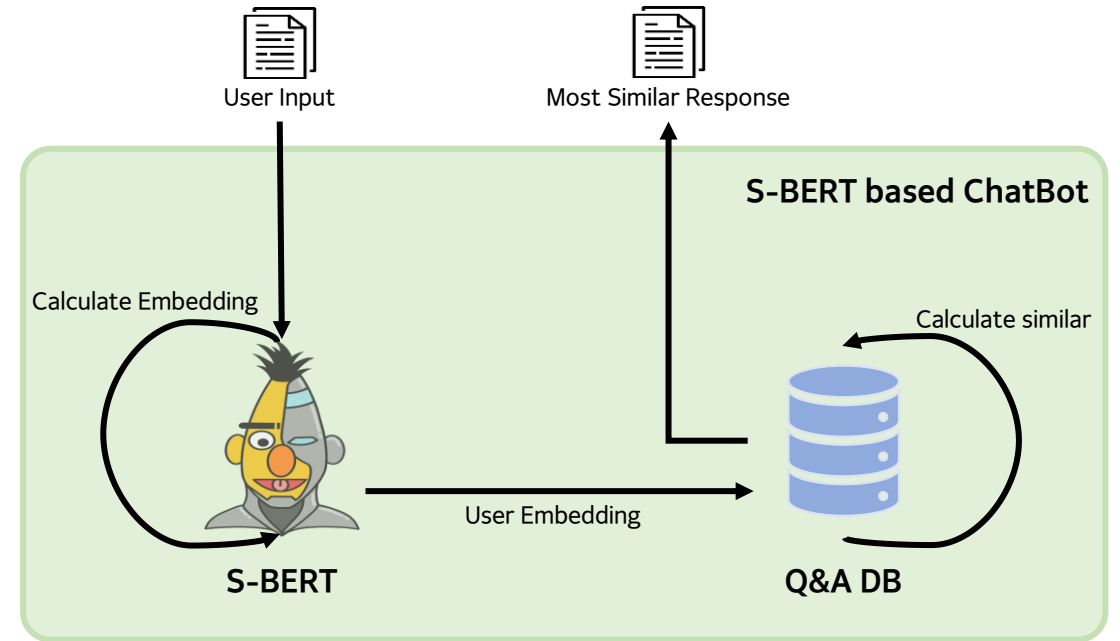




보완할 점

유사한 vector를 검색할 때, $O(n)$ 의 복잡도를 가집니다. 이를 보완하기 위해 Faiss 라이브러리를 활용할 수 있습니다.

Faiss 라이브러리는 정보를 효율적으로 저장하기 위해 quantization을 사용합니다. 그리고 효율적으로 searching을 하기 위해서 모든 vector를 몇 개의 cluster로 만들고, 이 cluster 중에서 몇 개만을 search합니다.

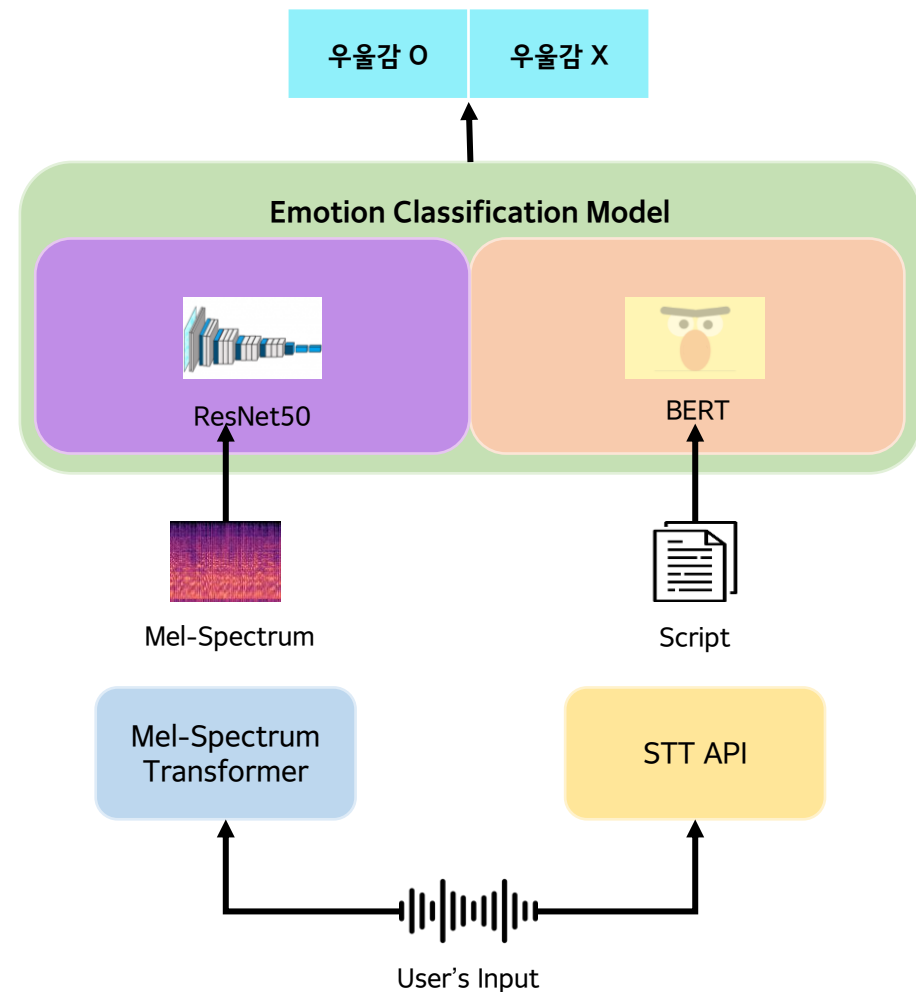




보완할 점

감정 분류를 위해 2가지 모델을 사용하고 있습니다. 이를 Knowledge Distillation과 같은 경량화기법을 적용할 필요가 있을 것입니다.

Knowledge Distillation은 용량이 큰 모델과 용량이 작은 모델 2가지를 준비합니다. 용량이 큰 모델은 미리 pretrain 시키고, 작은 모델에게 용량이 큰 모델의 지식을 전달하는 방식으로 학습을 진행합니다. 최종적으로 용량이 작은 모델이 용량이 큰 모델을 모방하도록 하는 것이 최종 목표입니다.



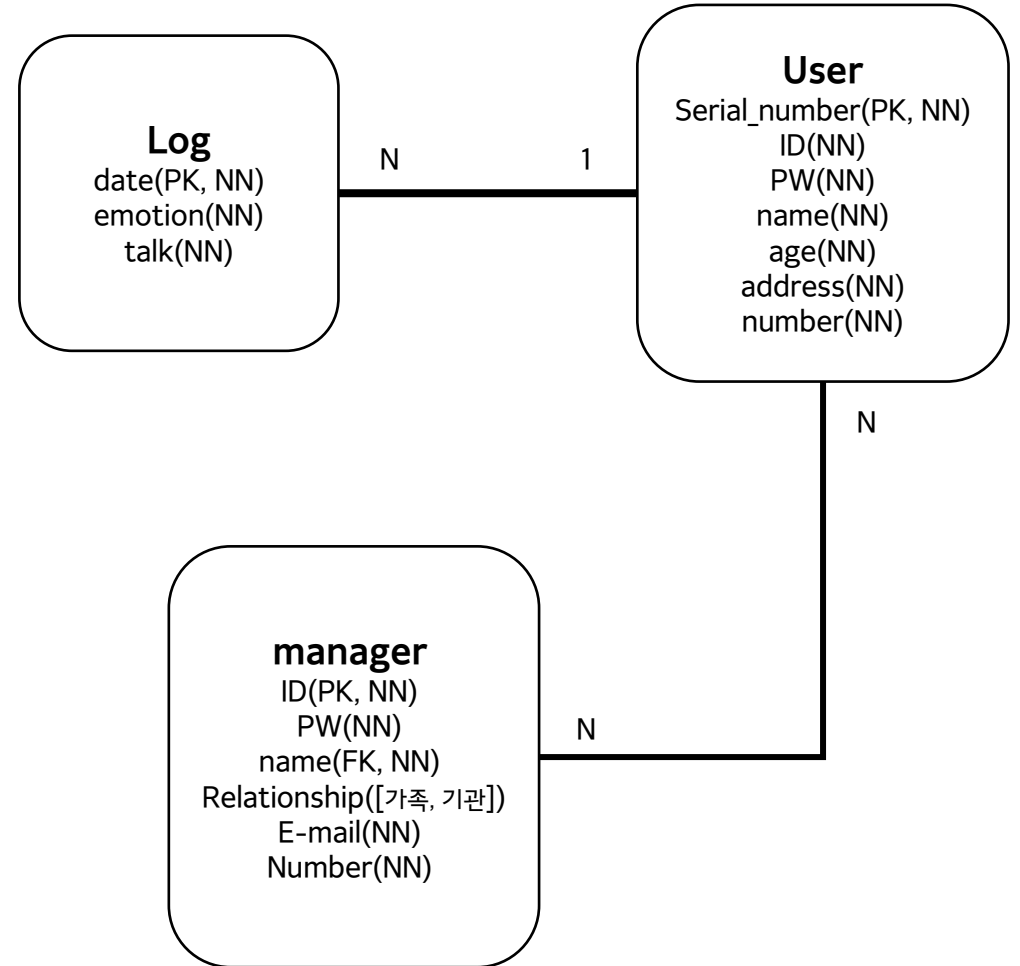


보완할 점

현재는 불필요한 저장소 구매를 하지 않기 위해 대화 내역을 포함한 모든 정보를 DB에 저장하였습니다.

하지만 대화내역은 매번 사용자가 말을 할 때마다 저장해야 하기 때문에 비용이 매우 클 것으로 보입니다.

이에 대한 개선 방법을 말씀드리겠습니다.





보완할 점

실제 서비스를 한다면 저장소를 따로 구매해서 대화내역 및 감정분석 결과는 json파일에

저장하고 DB에는 파일 경로만 저장하려합니다.

이 덕분에 DB에 추가하고 읽어오는 비용과 시간이 줄어들 것으로 보입니다.

./Log/'AAA0000000001.json'

```
{
  "2022-03-24": [[08:17:37, 행복, 안녕하세요~],
                 [11:38:12, 슬픔, 슬프니다...]],
  "2022-03-25": [[09:12:27, 슬픔, 슬프세요],
                 [12:08:37, 행복, 행복합니다...]]
}
```

User

Serial_number(PK, NN)
ID(NN)
PW(NN)
name(NN)
age(NN)
address(NN)
number(NN)
Log_경로(NN)

N

manager

ID(PK, NN)
PW(NN)
name(FK, NN)
Relationship([가족, 기관])
E-mail(NN)
Number(NN)