

AI 기반 추억 앨범 및 치매 예방 서비스 중간 보고서



부산대학교 정보컴퓨터공학부

지도교수 : 조준수
팀 명 : 치매노노
팀 원 : 202155514 김나림
202155540 김채현

목차

1. 요구조건 및 제약 사항 분석에 대한 수정 사항
 1. 음성 데이터
 2. 표정 데이터
 3. YOLO 모델
2. 설계 상세화 및 변경 내역
3. 갱신된 과제 추진 계획
4. 구성원별 진척도
5. 보고 시점까지의 과제 수행 내용 및 중간 결과
 1. 소셜 로그인 기반 사용자 인증 및 인가 기능 구현 (카카오, 네이버)
 2. AI 대화 세션 종료 후 대화 내용 요약 기능 구현
 3. 감정 일기 생성 기능 구현

1. 요구조건 및 제약 사항 분석에 대한 수정 사항

1-1. 음성 데이터

1. 제약 사항

사용자의 감정 분석 시 사용하는 데이터 중 하나인 음성 데이터 수집 중, 50대 이상의 한국인 음성 데이터를 수집하기 어려운 점을 확인하였다.

2. 수정 사항

음성 데이터 대신, 50대 이상 고령층의 텍스트 데이터를 활용하기로 결정하고, 텍스트 데이터를 확보하였다.

		기쁨	당황	분노	불안	상처	슬픔
Training	라벨링데이터	2882	4508	4760	4722	4748	4596
	원천데이터	2882	4508	4760	4722	4748	4596
Validation	라벨링데이터	696	527	663	682	556	507
	원천데이터	696	527	663	682	556	507

그림 1. 음성 데이터 전처리 결과 데이터 수

1-2. 표정 데이터

1. 제약 사항

표정 데이터의 데이터 전처리 중, 50대 이상의 데이터를 분류하는 과정에서 표정 데이터의 수가 현저히 적었다.

2. 수정 사항

연령의 기준점을 하향하여 표정 데이터에 한하여 40대 이상의 데이터까지 포함하도록 수정하였다.

		기쁨	당황	분노	불안	상처	슬픔
Training	라벨링데이터	1569	1604	1588	1574	1597	1612
	원천데이터	1569	1604	1588	1574	1597	1612
Validation	라벨링데이터	214	183	203	192	193	195
	원천데이터	214	183	203	192	193	195

그림 2. 표정 데이터 분류 결과 데이터 수 (40대 미포함)

		기쁨	당황	분노	불안	상처	슬픔
Training	<u>라벨링데이터</u>	8670	8729	8695	8624	8665	8686
	<u>원천데이터</u>	8670	8729	8695	8624	8665	8686
Validation	<u>라벨링데이터</u>	1136	1046	1063	1056	1056	1110
	<u>원천데이터</u>	1136	1046	1063	1056	1056	1110

그림 3. 표정 데이터 분류결과 데이터 수(40대 포함)

1-3. YOLO 모델

1. 제약 사항

표정 데이터에 대해 감정 라벨(기쁨, 당황, 분노, 불안, 상처, 슬픔)을 부여하고 YOLO 학습용 라벨(txt 파일)을 생성하는 과정에서, json 파일에서 bounding box 좌표와 감정 라벨을 추출하고 변환하는 작업에 많은 시간이 소요되었다. 특히 한 감정 카테고리 기준으로 약 10시간 이상이 걸리는 경우도 발생하여, 전체 개발 일정에 지연이 발생하였다.

2. 수정 사항

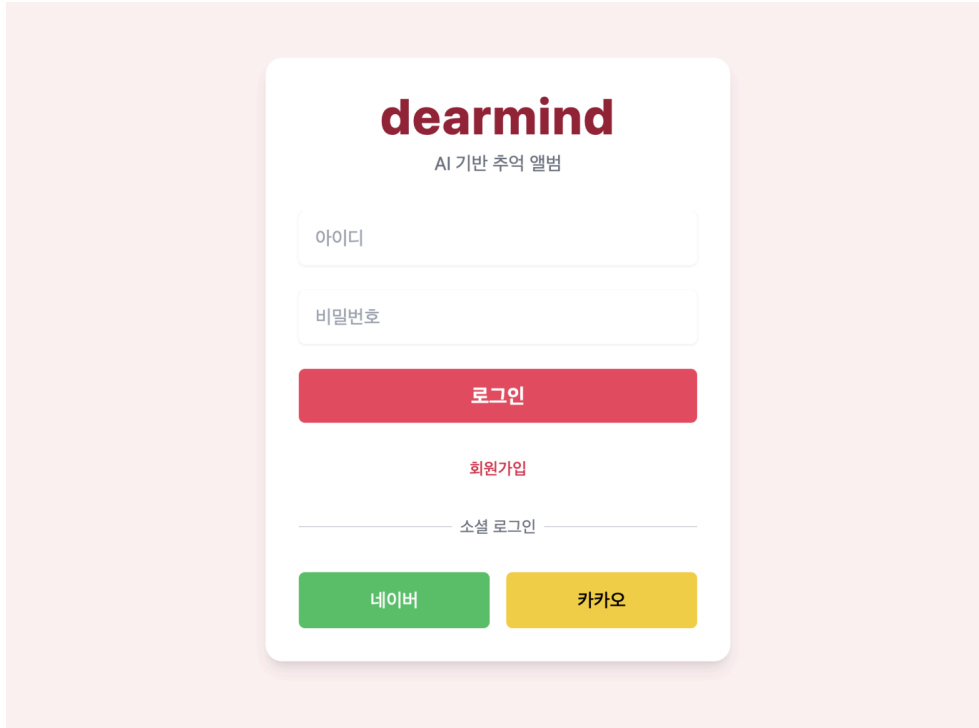
기존에는 AI 모델 구현(데이터 라벨링 포함)을 완료한 후에 웹 개발을 시작하는 순차적인 방식이었으나, 위에서 언급된 라벨링 지연 문제에 유연하게 대응하고 전반적인 프로젝트 기간을 최적화하기 위해, 라벨링 작업과 웹 개발을 동시에 진행하는 병행 방식으로 일정을 전면 수정하였다.

2. 설계 상세화 및 변경 내역

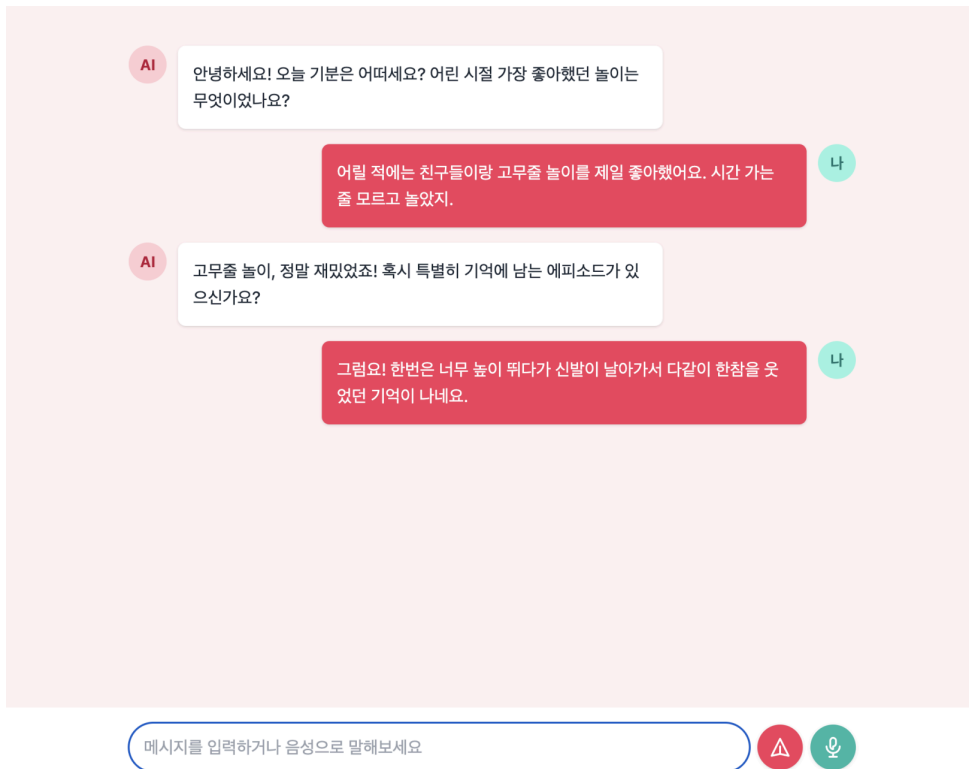
- a. 사용자 접속 및 로그인
 - i. 사용자는 **AWS EC2**에 배포된 웹 서비스에 접속한다.
 - ii. 본인 계정으로 로그인하거나, 가족 계정으로 로그인한다.
- b. 대화 세션 시작
 - i. 사용자는 '대화 시작' 버튼을 눌러 **AI**와의 회상 대화를 시작한다.
 - ii. 초기 추천 질문 리스트 중 하나를 선택한다.
- c. 음성 및 표정 데이터 수집
 - i. 사용자의 웹캠 및 마이크가 활성화된다.
 - ii. **Whisper API**를 통해 사용자의 발화 내용을 실시간 텍스트로 변환한다.
 - iii. **YOLO** 모델을 활용하여 실시간으로 사용자의 얼굴 표정을 분석하고, 감정 상태를 추출한다.
- d. AI 응답 생성 및 출력
 - i. **KoBERT** 모델을 통해 텍스트 감정 분석을 수행한다.
 - ii. 표정 감정과 텍스트 감정을 종합하여 현재 감정 상태를 판단한다.
 - iii. 표정 감정과 텍스트 감정을 종합하여 현재 감정 상태를 판단한다.
 - iv. **Naver CLOVA Voice API**를 이용해 **AI**의 텍스트 응답을 음성으로 변환하여 사용자에게 전달한다.
 - v. 사용자에게는 텍스트와 음성 두 가지 형태로 동시에 출력된다.
- e. 대화 기록 저장
 - i. 사용자와 **AI** 간의 모든 질문, 응답, 감정 분석 결과를 서버에 저장한다.
 - ii. 저장된 데이터는 개인별 일기 생성에 활용된다.
- f. 대화 종료 및 감성 일기 자동 생성
 - i. 사용자가 대화 종료를 선택하여 시스템이 종료됨을 감지한다.
 - ii. 종료 후 **ChatGPT API**를 통해 전체 대화 내용을 1인칭 시점의 감성 일기로 자동 생성한다.
 - iii. 전체 감정 분위기에 맞는 **YouTube** 음악 및 표지 디자인을 추천하고 일기에 함께 저장한다.
- g. 일기 및 앨범 저장 및 열람
 - i. 생성된 일기, 음악, 표지 데이터를 서버에 저장하고, 사용자 앨범에 등록한다.
 - ii. 사용자는 일기의 공개 범위(비공개, 가족 공개)를 설정할 수 있다.
 - iii. 가족 사용자가 접근 가능한 경우, 일기에 사진 첨부 기능이 활성화된다.
- h. 데이터 보관 및 관리
 - i. 모든 일기 및 대화 기록은 **MySQL** 기반 데이터베이스에 암호화 저장된다.
 - ii. 사용자 요청 시 데이터 삭제 또는 공개 범위 변경이 가능하다.
- i. 서비스 종료
 - i. 사용자가 로그아웃하거나 브라우저를 종료하면 서비스 세션이 종료된다.

5. 보고 시점까지의 과제 수행 내용 및 중간 결과

5-1. 소셜 로그인 기반 사용자 인증 및 인가 화면, 기능 구현 (카카오, 네이버)



5-2. 채팅 화면, 기능 구현



5-3. 감정 일기 생성 화면, 기능 구현

1974년 7월 17일

고무줄 놀이 하던 즐거운 날

공개

😊

행복



오늘 AI와 대화를 나누다 어린 시절 가장 좋아했던 놀이가 무엇이였는지에 대한 질문을 받았다. 잠시 생각에 잠기자, 뽀얀 흙먼지가 날리던 초등학교 운동장과 친구들의 웃음소리가 아련하게 떠올랐다. 그 시절 우리는 해가 지는 줄도 모르고 고무줄 놀이에 푹 빠져 살았다. '높고 낮은 새'를 외치며 발을 엇갈리고, 고무줄을 넘을 때마다 심장이 두근거렸다. 특히 기억에 남는 날이 있다. 유난히 맑은 가을날, 동네 친구들과 다 같이 모여 마지막 단계인 '할머니' 단계에 도전하고 있었다. 내가 경총 뛰어오르는 순간, 아끼던 내 하얀 운동화 한 짝이 벗겨져 포물선을 그리며 멀리 날아가 버렸다. 그 우스꽝스러운 모습에 우리는 모두 배를 잡고 한참을 웃었다. 신발을 주우러 달려가면서도 웃음이 멈추지 않았던 그 순간이 눈앞에 생생하다. 까맣게 잊고 지냈던 소중한 기억이었다. AI 덕분에 낡은 앨범을 넘기듯 행복한 추억 한 조각을 다시 꺼내볼 수 있었다. 덕분에 오늘 하루, 마음이 참 따뜻하고 풍요로워졌다.

이문세 - 소녀
응답하라 1988 OST

5-4. KOBERT 모델

```
# 클래스 가중치 계산
train_df = pd.read_csv('/content/drive/MyDrive/data/텍스트데이터/Training/원천데이터/감성대화말뭉치(최종데이터)_Training_언어추출_csv변환_50%샘플링.csv')
y_train = train_df['감정_대분류'].map(label_map).values
class_weights = compute_class_weight(class_weight='balanced', classes=np.unique(y_train), y=y_train)
weights_tensor = torch.tensor(class_weights, dtype=torch.float)

# 모델
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = BertForSequenceClassification.from_pretrained('monologg/kobert', num_labels=6)
model.to(device)

# Loss + Optimizer
loss_fn = torch.nn.CrossEntropyLoss(weight=weights_tensor.to(device))
optimizer = AdamW(model.parameters(), lr=2e-5)

# 학습 루프
epochs = 3
for epoch in range(epochs):
    model.train()
    total_loss = 0
    for batch in tqdm(train_loader):
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        labels = batch['labels'].to(device)

        outputs = model(input_ids=input_ids, attention_mask=attention_mask, labels=labels)
        loss = outputs.loss # 내부에서 CrossEntropy + 가중치 계산됨

    loss.backward()
    optimizer.step()
    optimizer.zero_grad()
```

- KoBERT 기반 감정 분류 모델 학습을 위한 1차적인 코드 구현을 완료하였으며, 현재는 기본적인 학습 루프와 클래스 가중치 적용을 포함한 형태로 구성되어 있다. 향후에는 학습 효율성과 모델 성능 향상을 위해 하이퍼파라미터 튜닝, 검증 루프 추가, Early Stopping 및 Learning Rate Scheduler 적용 등 최적화 작업을 진행할 예정이다.