상담을 위한 사용자의 답변 적극성과 일관성 판단 및 발화 생성 서비스

설계서

(요구기능: 사용자 답변 적극성과 일관성 기반의 발화 생성)

(요구기능 ID: CP_003)

(참조 파일명:

종합설계1_02_두리아_CP002_요구분석정의서.docx)

문서번호 :

Version 0.1

개정 이력

제.개정내역

버전	승인일자	개요	작성자

배포이력

버전	배포일자	배포처

검토이력

버전	검토일자	검토방법	검토자

목 차

1. 요구기능 설명

제공하는 주요 상세 기능

1. 적극성 점수와 일관성에 따른 발화 생성 - 이전 대화를 통해 산출된 적극성 점수와 답변 일관성을 기반으로 적합한 발화를 생성하여 원활한 대화를 하도록 한다.

기능의 동작 원리

1. 상황 및 상태 분석

시스템은 사용자의 현재 상태와 상황을 실시간으로 분석한다. 이 분석은 사용자의 최근 활동, 대화의 맥락, 그리고 기타 관련 데이터를 포함한다.

이러한 데이터는 사용자의 기분, 관심사, 그리고 대화에 대한 의지를 더욱 명확하게 파악할 수 있게 한다.

2. 적합한 발화 생성

사용자 답변 일관성 여부와 적극성 점수를 기반으로 시스템은 적합한 발화를 생성한다. 발화 생성 엔진은 사용자의 적극성 수준에 맞추어 동기를 부여하거나 대화를 유도하는 맞춤형 메시지를 형성한다.

동작 조건

1. 음성 데이터의 정확한 입력과 변환

사용자의 답변은 음성 형식으로 입력되어야 하며, 시스템은 이 음성 데이터를 텍스트로 변환하는 과정에서 높은 정확성을 보장해야 한다. 이는 분석의 기초 데이터로 사용되므로, 오류가 최소화되어야 한다.

2. 실시간 데이터 처리능력**:** 시스템은 사용자의 현재 상태와 상황을 실시간으로 분석할 수 있는 처리 능력을 갖추어야 한다. 이는 사용자의 즉각적인 데이터를 분석하여 동적으로 발화를 생성하는 데 필수적이다.

3. 개인화 데이터의 활용**:** 사용자의 개인화된 데이터, 예를 들어 이전 대화 내용, 사용자의 선호 및 반응 스타일 등을 기반으로 발화를 생성한다. 이를 위해 사용자 프로필과 행동 패턴을 정교하게 분석하고 저장하는 시스템이 필요하다.

주요 처리 대상

1. 사용자 답변: 사용자가 입력한 음성 데이터와 이를 텍스트로 변환한 데이터이다. 이 데이터는 사용자의 적극성을 분석하고 발화 생성의 기반으로 사용된다.

2. 일관성 여부 및 적극성 점수: 사용자의 음성 및 텍스트 데이터 분석을 통해 얻어진 결과와 산출된 적극성 점수이다. 이 점수는 사용자의 참여 수준을 나타내며, 발화 생성 프로세스에서 중요한 참조 지표로 활용된다. 또한 사용자의 답변 일관성 여부를 판단하고 분석하여 대화를 생성한다.

3. 사용자 프로필 및 행동 데이터: 사용자의 선호, 이전 상호작용, 반응 스타일 등을 포함한 개인화 데이터이다. 이 데이터는 발화를 더욱 개인화하고 사용자 맞춤형 대화를 촉진하는 데 사용된다.

처리 결과

1. 발화 생성: 적극성 점수와 사용자의 상태 및 상황에 따라 적합한 발화를 생성한다.

1.1. 주요 상세기능 설명

- ID: FJ_001

- 정의: 주요 상세 기능 #1 은 이전 대화를 통해 산출된 적극성 점수와 사용자의 답변 일관성을 기반으로 적합한 발화를 생성하여 원활한 대화를 하도록 한다.

- 상세 설명: 주요 상세 기능 #1 은 사용자의 답변에 나타나는 열정과 참여도를 분석하는 다양한 요소를 평가한 데이터(답변 음성데이터의 양과 질, 응답 속도, 사용된 어휘와 표현의 다양성 및 전문성을 포함)를 바탕으로 적절한 발화를 생성하여 사용자와의 원활한 대화를 목표로 한다.

1.2. 요구기능 동작 절차

단계 1. CP001- FJ001 을 통해 사용자 발화 수집 및 점수 생성

단계 2. CP002- FJ001 을 통해 일관성 판단

단계 3. 적극성 점수 및 답변 일관성을 기반으로 한 발화 생성

[단계별 상세 내용]

단계 1,2 는 CP001, CP002 문서에서 서술하였기에 3 단계만 서술한다.

단계 3: 단계 1 에서 산출된 점수와 단계 2 에서 답변 일관성 여부를 기반으로 적절한 발화를 생성하여 사용자와 원활한 대화를 진행한다.

1.3. 동작 규칙

동작 규칙 1: 데이터 보호 및 개인 정보 준수

모든 사용자 데이터, 특히 음성 데이터와 개인 식별 정보는 데이터 보호 법규와 개인정보 보호 정책에 따라 처리되어야 한다. 사용자의 데이터는 안전하게 저장되고, 접근은 엄격하게 제한되어야 한다.

동작 규칙 2: 실시간 처리 요구

사용자의 음성 데이터와 상황 분석은 실시간으로 처리되어야 한다. 이는 사용자의 적극성 점수를 즉각적으로 산출하고, 상황에 맞는 발화를 즉시 생성하여 대화의 자연스러움과 효과성을 보장하기 위함이다.

동작 규칙 3: 사용자 맞춤 발화 생성

생성된 발화는 사용자의 개별적인 적극성 점수와 상황에 맞게 맞춤화되어야 한다. 발화는 사용자가 대화에 적극적으로 참여하도록 동기를 부여하고, 관심을 유도하는 내용을 포함해야 한다.

1.4. 가정(Assumptions)

- 상담은 정보보호수집 동의서에 동의를 거쳐야 한다.

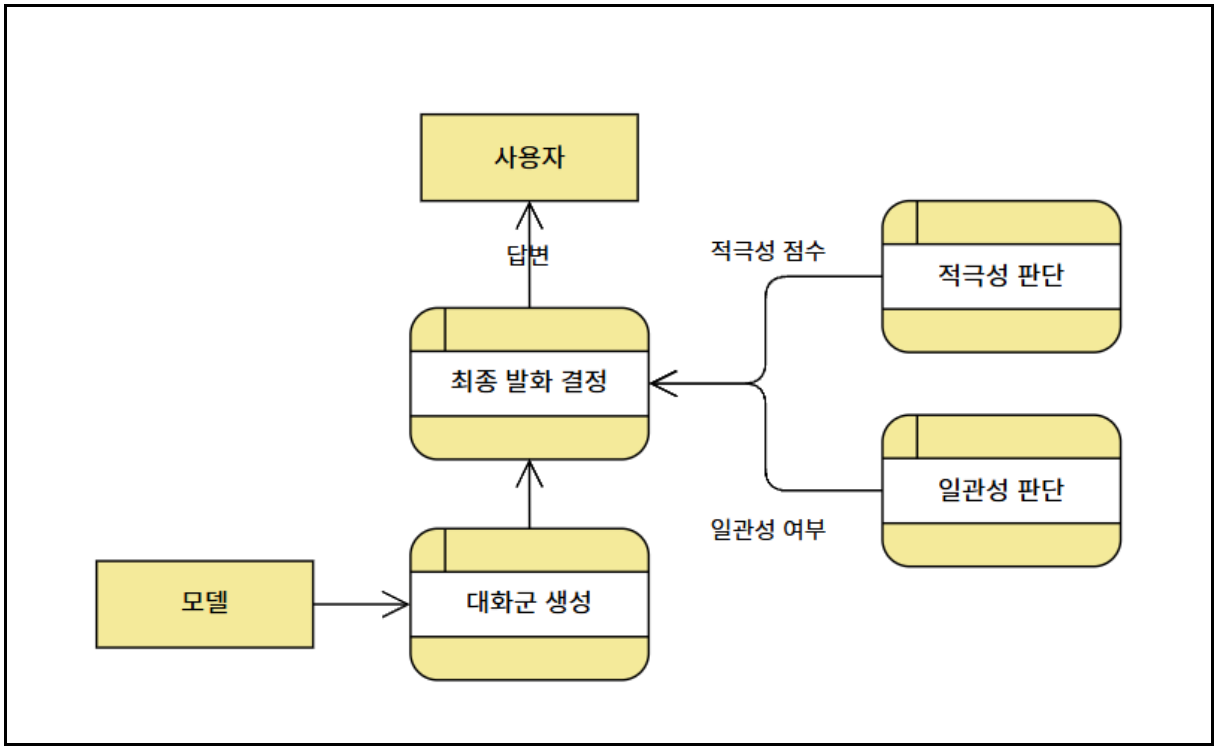
- 사용자가 상담을 거절할 경우 상담을 유도하는 것이 아닌 상담을 종료한다.

- 음성 데이터의 텍스트화가 원할하게 진행되어야 한다.

- 음성 데이터의 개인 정보가 유출되지 않게 잘 보호되어야 한다.

2. 기술설계 #1

2.1. 기능 설계



1. 모델과 토크나이저 로드와 DB 연결

```
model_name = "beomi/Llama-3-Open-Ko-8B"
tokenizer = AutoTokenizer.from_pretrained(model_name)
```

```
model = AutoModelForCausalLM.from_pretrained(model_name)

def connect_db(db_name="user_data.db"):
    conn = sqlite3.connect(db_name)
    return conn
```

- model_name: 사용하려는 모델의 이름 설명
- tokenizer: 입력 텍스트를 모델이 이해할 수 있는 형식으로 변환
- model: 텍스트 생성을 위한 사전 학습된 언어 모델
- db_name: 데이터베이스 파일 이름
- conn: SQLite 데이터베이스 연결 객체

2. DB 에서 사용자 데이터 가져오기

```
def get_user_data(user_id, conn):
    cursor = conn.cursor()
    cursor.execute("SELECT activeness_score, consistency FROM user_scores WHERE user_id=?",
(user_id,))
    row = cursor.fetchone()
    if row:
        activeness_score, consistency = row
        consistency = bool(consistency)
        return activeness_score, consistency
    else:
        raise ValueError(f"No data found for user_id {user_id}")
```

- activeness_score: 사용자의 적극성 점수
- consistency: 사용자의 일관성 여부

3. 발화 생성 및 예시 답변

```
def generate_dynamic_response(activeness_score, consistency):


    prompts = [
        "네, 그 이야기에 대해 더 듣고 싶어요.",
        "더 이야기해 주실 수 있나요?",
        "그것에 관심을 가지게 된 이유가 무엇인가요?",
        "정말 흥미롭네요! 계속 이야기해 주세요.",
        "궁금한데요, 좀 더 설명해 주실 수 있나요?",
        "왜 그것이 흥미로운가요?",
        "좋은 주제네요, 더 이야기해 볼까요?",
        "좀 더 자세히 말해 주시겠어요?",
        "매우 흥미롭네요, 더 이야기해 주세요!",
        "그것에 대해 더 알고 싶어요."
    ]
```

- generate_dynamic_response 함수는 사용자로부터 받은 적극성 점수와 일관성을 기반으로 적절한 응답을 생성하는 역할을 한다.
- prompts 리스트에는 다양한 예시 응답이 포함되어 있음. 이 예시 응답들은 모델이 응답을 생성할 때 기본 프롬프트로 사용됨.

4. 적극성 점수와 일관성을 기반으로 답변 선택

```
activeness_score_index = min(int(activeness_score), 9)
consistency_index = 0 if consistency else 1
prompt_index = (activeness_score_index + consistency_index) % 10
selected_prompt = prompts[prompt_index]
```

- activeness_score_index: 적극성 점수 인덱스
- consistency_index: 일관성 여부에 따라 0 또는 1 로 설정되는 인덱스

- prompt_index: activeness_score_index 와 consistency_index 를 기반으로 계산된 프롬프트 인덱스
- selected_prompt: 선택된 예시 프롬프트

5. 모델을 사용하여 대화 생성

```
inputs = tokenizer(selected_prompt, return_tensors="pt")
outputs = model.generate(inputs.input_ids, max_length=50)
response = tokenizer.decode(outputs[0], skip_special_tokens=True)

return response
```

- selected_prompt: 선택된 예시 프롬프트
- inputs: 토큰화된 입력 텍스트
- outputs: 모델이 생성한 출력 텍스트
- response**:** 최종적으로 생성된 응답 텍스트

2.2. 요소 설계

2.2.1. Process

프로세스 1: 모델과 토크나이저 로드와 DB 연결

- model_name 변수는 사용하려는 모델의 이름을 지정. 여기서는 "beomi/Llama-3-Open-Ko-8B" 모델을 사용한다.

- AutoTokenizer.from_pretrained(model_name): 해당 모델의 토크나이저를 로드. 입력 텍스트를 모델이 이해할 수 있는 형식으로 변환하는 역할

- AutoModelForCausalLM.from_pretrained(model_name): 텍스트 생성에 사용되는 사전 학습된 언어 모델

- def connect_db(db_name="user_data.db"): SQLite 데이터베이스에 연결

- sqlite3.connect(db_name): 지정된 데이터베이스 파일에 연결.

프로세스 2: DB 에서 사용자 데이터 가져오기

- def get_user_data(user_id, conn): 특정 사용자 ID 를 기반으로 데이터베이스에서 사용자 데이터를 가져옴

- conn.cursor(): 데이터베이스 커서를 생성.

- cursor.execute("SELECT activeness_score, consistency FROM user_scores WHERE user_id=?", (user_id,)): 지정된 사용자 ID 에 대한 적극성 점수와 일관성을 가져오는 SQL 쿼리를 실행

- cursor.fetchone(): 쿼리 결과에서 한 행을 가져 옴. 결과가 있으면 activeness_score 와 consistency 값을 추출.

프로세스 3: 발화 생성 및 예시 답변

- def generate_dynamic_response(activeness_score, consistency): 사용자로부터 받은 적극성 점수와 일관성을 기반으로 적절한 응답을 생성

- prompts 리스트에는 다양한 예시 응답이 포함되어 있음. 이 예시 응답들은 모델이 응답을 생성할 때 기본 프롬프트로 사용

프로세스 4: 적극성 점수와 일관성을 기반으로 답변 선택

- activeness_score_index = min(int(activeness_score), 9): 적극성 점수를 0 에서 9 사이의 값으로 변환.

- consistency_index = 0 if consistency else 1: 일관성이 있으면 0, 없으면 1 로 설정

- prompt_index = (activeness_score_index + consistency_index) % 10: 적극성 점수 인덱스와 일관성 인덱스를 더한 후 10 으로 나눈 나머지 값으로 계산.

- selected_prompt = prompts[prompt_index]: 계산된 인덱스를 사용하여 prompts 리스트에서 선택된 프롬프트를 가져옴.

프로세스 5: 모델을 사용하여 대화 생성

- tokenizer(selected_prompt, return_tensors="pt"): 선택된 프롬프트를 토크나이저를 통해 토큰화 및 텐서 형식 변환

- model.generate(inputs.input_ids, max_length=50): 모델이 입력된 프롬프트를 기반으로 응답을 생성

- tokenizer.decode(outputs[0], skip_special_tokens=True): 생성된 토큰을 텍스트 형식으로 디코딩

2.2.2. Data

이름	타입	범위	출력 프로세스	입력 프로세스	비고
user_id	문자열	사용자의 고유 ID	대화 데이터 저장, 대화 데이터 조회, 일관성 분석 결과 반환	대화 데이터 저장, 과거 대화 데이터 요청, 현재 대화 데이터 요청, 일관성 분석	사용자를 식별하는 고유 ID

				요청	
activeness_score	실수	0.0 ~ 1.0	분석 요청 응답 반환	음성 데이터 분석	NLP 분석 결과
consistency	불리언	True, False	일관성 분석 결과 반환	일관성 분석 요청	사용자의 대화 일관성 여부
selected_prompt	문자열	대화텍스트 데이터	모델을 통해 다수의 대화 데이터 추출	-	모델 입력 프롬프트
response	문자열	대화텍스트 데이터	다수의 대화 데이터 중 문자열 추출	-	모델 출력 응답 텍스트

2.2.3. 저장소

저장소 이름	목적	사용 프로세스	구성 요소
사용자 정보 데이터베이스 (UserDB)	사용자 개인 정보 및 인증 데이터 저장	음성 데이터 업로드, 분석 요청, 결과 저장	Users 테이블
점수 데이터베이스 (scoreDB)	일관성 여부와 적극성 점수 데이터 저장	점수 수집, 점수 기반 분석	Score 테이블
대화 데이터베이스 (responseDB)	모델을 통해 생성된 대화 데이터 저장	대화 추출	Response 테이블

1. 사용자 정보 데이터베이스 (UserDB)

목적: 사용자 개인 정보 및 인증 데이터 저장

구성 요소: Users 테이블:

- user_id (문자열, PRIMARY KEY): 사용자 고유 ID

- username (문자열): 사용자 이름

- password (문자열): 사용자 암호

- email (문자열): 사용자 이메일

2. 점수 데이터베이스 (scoreDB)

목적: 일관성 여부와 적극성 점수 데이터 저장

구성 요소: Score 테이블:

- user_id (문자열, FOREIGN KEY REFERENCES UserDB.Users(user_id)): 사용자 고유 ID

- consistency (불리언): 대화 일관성 여부

- activeness_score (실수): 대화 적극성 점수

3. 대화 데이터베이스 (responseDB)

목적: 모델을 통해 생성된 대화 데이터 저장

구성 요소: Response 테이블:

- user_id (문자열, FOREIGN KEY REFERENCES UserDB.Users(user_id)): 사용자 고유 ID

- selected_prompt (문자열): 대화 중에서 고르기 위한 다수의 문자열

- response (문자열): selected_prompt 중에서 골라진 최적의 문자열

3. 기타사항

CP 001을 통해 사용자의 답변을 바탕으로 적극성 점수를 수집하고 CP002을 통해 사용자 답변의 일관성을 분석한 후 이를 바탕으로 위의 발화 생성 알고리즘을 사용하여 사용자와 대화를 지속한다.