

기술 명세서

SUMARRY

챗봇은 명확하고 문화적으로 관련된 지침을 제공함으로써 경계선 지적 기능을 가진 개인이 일상적인 문제에 답을 주는데 도움을 주기 위해 특별히 설계했습니다. 네 가지 주요 기능에 목표를 두었습니다.

의류 추천 시스템: TensorFlow 기반CNN 모델을 활용하여 한국의 장례식장과결혼식장에 적합한 복장을 추천해 주며, 한국 전통의 맥락 안에서 올바른 행동, 조의금, 축의금 등 에티켓에 대한 필수지침을 제공합니다. 또한 우리의 권장 사항은 노출, 색상 및 기타 문화적 차이와 같은 요소를 신중하게 고려하여 정중하고 적절한 외모를 보장합니다.

일상 대화: 표준 챗봇과 대화에 참여할 수 있지만, 널리 알려진ChatGPT와 같은 많은 챗봇은 주로 글로벌데이터세트에 대해 교육을 받았기 때문에 특히 한국의 맥락과 문화에 초점을 맞추는 것이 어려운 경우가 많습니다. 이 문제를 해결하기 위해 우리는한국을 강조하는 프롬프트를 미세 조정하여 문화적으로 더 관련성이 있고 한국 사용자의 요구에 맞게 맞춤화되도록 접근 방식을 개선했습니다.

OCR 기반 매뉴얼 단순화: 사용자가 이해하기 어려운 지침이나 매뉴얼을 접하면간단히 챗봇에 사진을 올려 입력하면 됩니다. 챗봇은OCR 기술을 활용하여 텍스트를 추출하고, 경계선 지적 기능을 가진 개인의 요구 사항을인식하여 초등학생도 쉽게 이해할 수 있도록 설명을 전달합니다. 또한 이러한 설명은 한국의 문화와 관습을 반영하도록 신중하게 맞춤화되어 명확성과문화적 관련성을 보장했습니다.

문장 단순화 및 요약: 이해하기 어려운 텍스트가 나오면 사용자는 이를챗봇에 입력할 수 있습니다. 그런 다음 챗봇은 문화적 맥락과 관련성을 유지하면서 복잡한 문장을 명확하고 이해하기 쉬운 언어로 단순화합니다.

챗봇의 강점은 문화적 적응, 간단한 언어 사용, 복잡한 정보에 접근 가능하게 만드는능력에 있으며 경계선에 있는 지능적인 개인이 일상 생활에서 사용할 수 있는 맞춤형 도구가 되고자 합니다.

Github 코드<https://github.com/Batwan01/BIF-chatbot>

```
project_root/
|
|—— image/                # 이미지 파일을 저장하는 디렉토리
|
|—— models/               # 모델 파일을 저장하는 디렉토리
|   |—— funeral_model.keras  # 장례식장 착장 모델
|   |—— wedding_model.keras  # 결혼식장 착장 모델
|
|—— static/              # 정적 파일(자바스크립트, CSS) 저장
|   |—— app.js             # 애플리케이션 자바스크립트 파일
|   |—— style.css          # 스타일시트 파일
|
|—— templates/           # HTML 템플릿 파일 저장
|   |—— index.html         # 메인 페이지 템플릿
|
|—— utils/               # 유틸리티 파일 저장
|   |—— chatbot_response.py  # 챗봇 응답 관련 유틸리티
|   |—— image_processing.py  # 이미지 처리 관련 유틸리티
|   |—— ocr_response.py     # OCR 응답 관련 유틸리티
|
|—— requirements.txt      # 프로젝트에서 사용하는 패키지와 해당 버전
|
|—— app.py               # 메인 애플리케이션 파일
```

app.py

```
|—— models/                                # 모델 파일을 저장하는 디렉토리
|   |—— funeral_model.keras                # 장례식장 착장 모델
|   |—— wedding_model.keras              # 결혼식장 착장 모델
|
|—— utils/                                # 유틸리티 파일 저장
|   |—— chatbot_response.py              # 챗봇 응답 관련 유틸리티
|   |—— image_processing.py              # 이미지 처리 관련 유틸리티
|   |—— ocr_response.py                  # OCR 응답 관련 유틸리티
```

기능	설명
일상 대화기능	<div>utils/chatbot_response.py</div> <div>create_prompt() AI 모델에 전달할 수 있는 특정 형식의 프롬프트로 변환하는 함수 AI 모델과의 상호작용에서 사용자에게 적절한 답변을 제공하기 위해 사용</div> <div>generate_response() api를 이용하여 사용자로부터 입력받은 prompt에 대한 응답을 생성함수 prompt이용하여 이용자에 맞춤 대답</div> <div>summarize_text_with_gpt() 주어진 텍스트를 요약하는 함수 긴 문서나 내용을 짧게 요약할 때 유용하게 사용할 수 있다</div>
착장에 대한 판별	<div>utils/image_processing.py models\funeral_model.keras models\funeral_model.keras</div> <div>classify_image() 주어진 이미지를 특정 행사(장례식장, 결혼식장)에 적합한지 여부를 분류하 는 역할 사전 훈련된 모델을 사용하여 이미지를 평가 평가 결과는 해당 이벤트에 적합한지 여부를 판단한 후, 결과를 반환</div>

기능	설명
설명기능	<p>utils/ocr_response.py</p> <p>extract_text_using_google_vision() 이미지를 분석하고, 해당 이미지에서 텍스트를 추출하는 역할 주어진 이미지를 바이너리 데이터로 변환하고, API에 전달하여 이미지 속의 텍스트를 인식한 후, 추출된 텍스트를 반환</p> <p>simplify_text_with_gpt() 쉽게 이해할 수 있도록 다시 작성하는 역할 초등학생도 이해할 수 있는 언어를 사용하여 텍스트를 단순화하며, 공손하고 정중한 어조로 답변을 생성</p> <p>save_text_to_jpeg() 주어진 텍스트를 이미지에 그려서 JPEG 파일로 저장하는 기능을 수행 텍스트를 주어진 너비에 맞게 줄바꿈 처리하고, 필요한 경우 이미지의 높이를 자동으로 조정하여 텍스트가 잘리지 않도록 함</p>

기능	설명
요약기능	<p>utils/chatbot_response.py</p> <p>summarize_text_with_gpt() 긴 텍스트를 간결하게 요약하여 사용자가 핵심 내용을 빠르게 파악할 수 있도록 도움</p>

기능	설명
이미지 인코딩	<p>encode_image() 이미지 데이터를 JSON으로 전송할 때, 또는 데이터 URI 스킴을 통해 이미지를 HTML에 직접 포함할 때 사용 JPEG 형식으로 통일 메모리 내에서 모든 작업을 수행하므로, 디스크 I/O를 최소화하고, 성능을 향상</p>

Model

funeral_model.keras
wedding_model.keras

기능	설명
라벨 설정	<pre># 라벨을 적합/부적합으로 수정 funeral_label_map = { 'Funeral_Appropriate': 1, # 1은 장례식에 적합함을 나타냅니다. 'Funeral_Inappropriate': 0, # 0은 장례식에 부적합함을 나타냅니다. 'Wedding_Appropriate': -1, # -1은 결혼식과 관련된 폴더로, 장례식과는 관련이 없음을 나타냅니다. 'Wedding_Inappropriate': -1, # -1은 결혼식과 관련된 폴더로, 장례식과는 관련이 없음을 나타냅니다. } wedding_label_map = { 'Funeral_Appropriate': -1, # -1은 장례식과 관련된 폴더로, 결혼식과는 관련이 없음을 나타냅니다. 'Funeral_Inappropriate': -1, # -1은 장례식과 관련된 폴더로, 결혼식과는 관련이 없음을 나타냅니다. 'Wedding_Appropriate': 1, # 1은 결혼식에 적합함을 나타냅니다. 'Wedding_Inappropriate': 0, # 0은 결혼식에 부적합함을 나타냅니다. }</pre>
기능	설명
데이터 분할	<pre># 장례식 데이터셋을 학습 세트와 검증 세트로 분리 # 데이터의 20%를 검증 세트로 사용하고, 나머지 80%를 학습 세트로 사용 # funeral_labels에 따라 데이터가 균형 있게 분포되도록 계층적 분할(stratified split)을 수행 # random_state=42를 설정하여 결과를 재현 가능하도록 무작위성을 제어 funeral_train_df, funeral_val_df = train_test_split(funeral_data, test_size=0.2, stratify=funeral_data['funeral_labels'], random_state=42) # 결혼식 데이터셋을 학습 세트와 검증 세트로 분리 # 데이터의 20%를 검증 세트로 사용하고, 나머지 80%를 학습 세트로 사용 # wedding_labels에 따라 데이터가 균형 있게 분포되도록 계층적 분할(stratified split)을 수행 # random_state=42를 설정하여 결과를 재현 가능하도록 무작위성을 제어 wedding_train_df, wedding_val_df = train_test_split(wedding_data, test_size=0.2, stratify=wedding_data['wedding_labels'], random_state=42)</pre>

기능	설명
전처리	<pre> # 이미지 데이터 제너레이터 설정 (이미지 픽셀 값을 0-1 사이로 정규화) train_datagen = ImageDataGenerator(rescale=1./255) val_datagen = ImageDataGenerator(rescale=1./255) # 장례식 학습 데이터 제너레이터 생성 # 학습 데이터프레임에서 이미지 파일 경로와 라벨을 불러와 배치 단위로 학습 데이터 생성 funeral_train_generator = train_datagen.flow_from_dataframe(funeral_train_df, # 장례식 학습 데이터프레임 x_col='filepaths', # 이미지 파일 경로가 저장된 열 이름 y_col='funeral_labels', # 장례식 라벨이 저장된 열 이름 target_size=(150, 150), # 모든 이미지를 150x150 픽셀로 크기 조정 batch_size=32, # 배치 크기를 32로 설정 class_mode='binary' # 이진 분류를 위해 라벨을 0 또는 1로 처리) # 장례식 검증 데이터 제너레이터 생성 # 검증 데이터프레임에서 이미지 파일 경로와 라벨을 불러와 배치 단위로 검증 데이터 생성 funeral_val_generator = val_datagen.flow_from_dataframe(funeral_val_df, # 장례식 검증 데이터프레임 x_col='filepaths', # 이미지 파일 경로가 저장된 열 이름 y_col='funeral_labels', # 장례식 라벨이 저장된 열 이름 target_size=(150, 150), # 모든 이미지를 150x150 픽셀로 크기 조정 batch_size=32, # 배치 크기를 32로 설정 class_mode='binary' # 이진 분류를 위해 라벨을 0 또는 1로 처리) # 결혼식 학습 데이터 제너레이터 생성 # 학습 데이터프레임에서 이미지 파일 경로와 라벨을 불러와 배치 단위로 학습 데이터 생성 wedding_train_generator = train_datagen.flow_from_dataframe(wedding_train_df, # 결혼식 학습 데이터프레임 x_col='filepaths', # 이미지 파일 경로가 저장된 열 이름 y_col='wedding_labels', # 결혼식 라벨이 저장된 열 이름 target_size=(150, 150), # 모든 이미지를 150x150 픽셀로 크기 조정 batch_size=32, # 배치 크기를 32로 설정 class_mode='binary' # 이진 분류를 위해 라벨을 0 또는 1로 처리) # 결혼식 검증 데이터 제너레이터 생성 # 검증 데이터프레임에서 이미지 파일 경로와 라벨을 불러와 배치 단위로 검증 데이터 생성 wedding_val_generator = val_datagen.flow_from_dataframe(wedding_val_df, # 결혼식 검증 데이터프레임 x_col='filepaths', # 이미지 파일 경로가 저장된 열 이름 y_col='wedding_labels', # 결혼식 라벨이 저장된 열 이름 target_size=(150, 150), # 모든 이미지를 150x150 픽셀로 크기 조정 batch_size=32, # 배치 크기를 32로 설정 class_mode='binary' # 이진 분류를 위해 라벨을 0 또는 1로 처리) </pre>

기능	설명
모델 구성	<pre>from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, Flatten, Dense, Dropout def create_model(): # Sequential 모델을 사용하여 신경망을 층별로 순차적으로 쌓아 구성 model = Sequential([Input(shape=(150, 150, 3)), # 입력층: 150x150 크기의 RGB 이미지 (3 채널) # 첫 번째 Conv2D 레이어: 32개의 필터, 3x3 커널 크기, ReLU 활성화 함수 사용 Conv2D(32, (3, 3), activation='relu'), MaxPooling2D((2, 2)), # 첫 번째 MaxPooling 레이어: 2x2 크기로 풀링 # 두 번째 Conv2D 레이어: 64개의 필터, 3x3 커널 크기, ReLU 활성화 함수 사용 Conv2D(64, (3, 3), activation='relu'), MaxPooling2D((2, 2)), # 두 번째 MaxPooling 레이어: 2x2 크기로 풀링 # 세 번째 Conv2D 레이어: 128개의 필터, 3x3 커널 크기, ReLU 활성화 함수 사용 Conv2D(128, (3, 3), activation='relu'), MaxPooling2D((2, 2)), # 세 번째 MaxPooling 레이어: 2x2 크기로 풀링 Flatten(), # Flatten 레이어: 2D 특성 맵을 1D 벡터로 변환 Dense(512, activation='relu'), # 전결합층 (Dense 레이어): 512개의 뉴런, ReLU 활성화 함수 사용 Dropout(0.5), # Dropout 레이어: 과적합 방지를 위해 50%의 뉴런을 무작위로 비활성화 # 출력층 (Dense 레이어): 1개의 뉴런, 시그모이드 활성화 함수 사용하여 이진 분류 수행 Dense(1, activation='sigmoid')]) # 모델 컴파일: Adam 옵티마이저 사용, 손실 함수로 binary_crossentropy 사용, 평가 지표로 정확도 사용 model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']) return model # 생성된 모델 반환</pre>

기능	설명
모델 학습	<pre> # 장례식 모델 학습 # 장례식 데이터를 학습시키기 위해 create_model() 함수를 호출하여 모델을 생성합니다. funeral_model = create_model() # 모델을 10번의 에포크 동안 학습시킵니다. # funeral_train_generator를 통해 학습 데이터를 제공하고, # funeral_val_generator를 통해 검증 데이터를 사용하여 모델의 성능을 평가합니다. funeral_history = funeral_model.fit(funeral_train_generator, epochs=10, validation_data=funeral_val_generator) # 결혼식 모델 학습 # 결혼식 데이터를 학습시키기 위해 create_model() 함수를 호출하여 모델을 생성합니다. wedding_model = create_model() # 모델을 10번의 에포크 동안 학습시킵니다. # wedding_train_generator를 통해 학습 데이터를 제공하고, # wedding_val_generator를 통해 검증 데이터를 사용하여 모델의 성능을 평가합니다. wedding_history = wedding_model.fit(wedding_train_generator, epochs=10, validation_data=wedding_val_generator) </pre> <p>Epoch 1/10</p> <p>5/5 ----- 16s 3s/step - accuracy: 0.7198 - loss: 0.6219 - val_accuracy: 0.8429 - val_loss: 0.3389</p> <p>Epoch 2/10</p> <p>5/5 ----- 16s 1s/step - accuracy: 0.8518 - loss: 0.3573 - val_accuracy: 0.9457 - val_loss: 0.2368</p> <p>Epoch 3/10</p> <p>5/5 ----- 11s 2s/step - accuracy: 0.9645 - loss: 0.2396 - val_accuracy: 0.9829 - val_loss: 0.1017</p> <p>Epoch 4/10</p> <p>5/5 ----- 8s 1s/step - accuracy: 0.9910 - loss: 0.0328 - val_accuracy: 0.9929 - val_loss: 0.0625</p> <p>Epoch 5/10</p> <p>5/5 ----- 10s 2s/step - accuracy: 0.9951 - loss: 0.0187 - val_accuracy: 0.9944 - val_loss: 0.0371</p> <p>Epoch 6/10</p> <p>5/5 ----- 9s 2s/step - accuracy: 0.9976 - loss: 0.0090 - val_accuracy: 0.9957 - val_loss: 0.0338</p> <p>Epoch 7/10</p> <p>5/5 ----- 8s 1s/step - accuracy: 0.9976 - loss: 0.0062 - val_accuracy: 0.9960 - val_loss: 0.0323</p> <p>Epoch 8/10</p> <p>5/5 ----- 10s 2s/step - accuracy: 0.9996 - loss: 0.0043 - val_accuracy: 0.9974 - val_loss: 0.0298</p> <p>Epoch 9/10</p> <p>5/5 ----- 8s 1s/step - accuracy: 0.9998 - loss: 0.0036 - val_accuracy: 0.9984 - val_loss: 0.0289</p> <p>Epoch 10/10</p> <p>5/5 ----- 10s 2s/step - accuracy: 0.9999 - loss: 0.0015 - val_accuracy: 0.9990 - val_loss: 0.0181</p>

기능	설명
모델 평가	<div><pre># 평가 # 장례식 모델을 평가하여 검증 데이터에 대한 손실 값과 정확도를 반환합니다. funeral_val_loss, funeral_val_acc = funeral_model.evaluate(funeral_val_generator) # 장례식 모델의 검증 정확도를 소수점 둘째 자리까지 출력합니다. print(f'Funeral Validation accuracy: {funeral_val_acc:.2f}')</pre><pre># 결혼식 모델을 평가하여 검증 데이터에 대한 손실 값과 정확도를 반환합니다. wedding_val_loss, wedding_val_acc = wedding_model.evaluate(wedding_val_generator) # 결혼식 모델의 검증 정확도를 소수점 둘째 자리까지 출력합니다. print(f'Wedding Validation accuracy: {wedding_val_acc:.2f}')</pre></div> <div>2/2 _____ 1s 82ms/step - accuracy: 0.9984 - loss: 0.0289 Funeral Validation accuracy: 0.99</div> <div>1/1 _____ 1s 829ms/step - accuracy: 0.9984 - loss: 0.0289 Wedding Validation accuracy: 0.79</div> <div>약 99% 정확도</div>

기능	설명
예측 (옷 색상, 노출 정도)	<div data-bbox="485 342 890 824"> <p>Funeral: Inappropriate, Wedding: Appropriate</p>  <p>1/1 ————— 0s 37ms/step 1/1 ————— 0s 44ms/step</p> </div> <div data-bbox="898 342 1321 824"> <p>Funeral: Appropriate, Wedding: Appropriate</p>  <p>1/1 ————— 0s 37ms/step 1/1 ————— 0s 36ms/step</p> </div>
	<div data-bbox="485 857 890 1317"> <p>Funeral: Inappropriate, Wedding: Inappropriate</p>  <p>1/1 ————— 0s 45ms/step 1/1 ————— 0s 37ms/step</p> </div> <div data-bbox="898 857 1321 1317"> <p>Funeral: Inappropriate, Wedding: Inappropriate</p>  <p>1/1 ————— 0s 35ms/step 1/1 ————— 0s 38ms/step</p> </div>
	<div data-bbox="485 1350 890 1809"> <p>Funeral: Inappropriate, Wedding: Inappropriate</p>  <p>1/1 ————— 0s 46ms/step 1/1 ————— 0s 62ms/step</p> </div> <div data-bbox="898 1350 1321 1809"> <p>Funeral: Inappropriate, Wedding: Inappropriate</p>  <p>1/1 ————— 0s 61ms/step 1/1 ————— 0s 88ms/step</p> </div>

Funeral: Inappropriate, Wedding: Appropriate



1/1 ————— 0s 64ms/step
1/1 ————— 0s 72ms/step

Funeral: Inappropriate, Wedding: Appropriate

