2024 한양대학교

SW/ICT/AI 종합학술대회

[분야 : 산학연구개발/연구인턴]

제목: AI 챗봇 기반 자기소개서 작성 지원 시스템 설계 및 구현

박윤아(22학번)

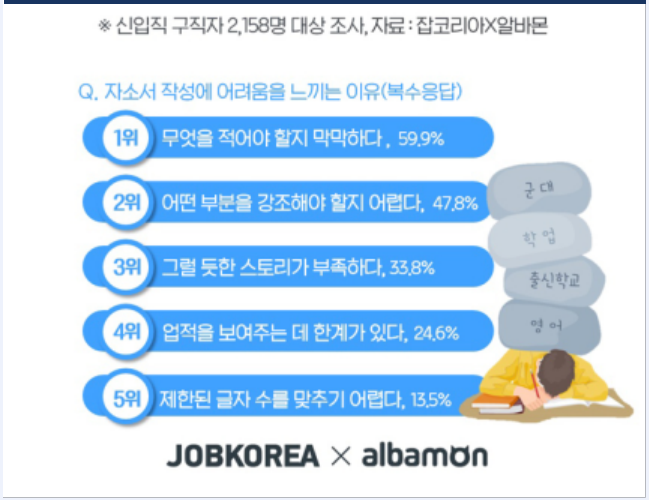
2024. 11. 21

한양대학교 ERICA 소프트웨어융합대학 / 한양대학교 ERICA SW중심대학사업단

연구 배경

자기소개서 작성의 어려움과 기술적 대안

<자기소개서 준비 현황> 관련 설문조사



자기소개서 작성 과정의 문제점

- 시간 소모
- 기업별 요구 인재상 반영 어려움
- 방대한 문서 분석의 비효율성

AI 기술의 발전에도 불구하고, 사용자 개개인 맞춤형 정보의 제공이 부족함

연구 목표

PDF 분석을 활용해 자기소개서 작성 지원



01

PDF 분석 기술 활용

기업 보고서에서 유용한 정보 추출

02

AI 기반 챗봇 설계

실시간 질문 응답 및 정보 제공

03

사용자 맞춤형 정보 제공

사용자 데이터 기반의 맞춤형 정보 제공

04

효율적인 자기소개서 작성 지원

정보 활용을 통해 시간 절약 및 품질 향상

연구 설계

시스템 설계 개요

3단계 설계

01.

PDF 분석 및 데이터 추출

02.

정보 제공 및 활용 지원

03.

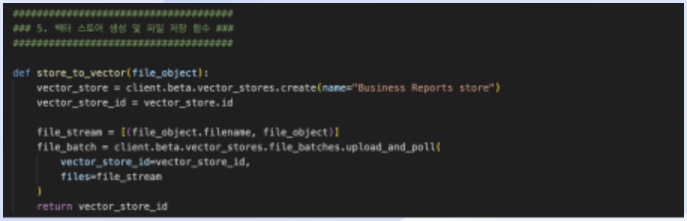
사용자 피드백 및 상호작용

사용 기술

OpenAI Assistants API

Flask 프레임워크

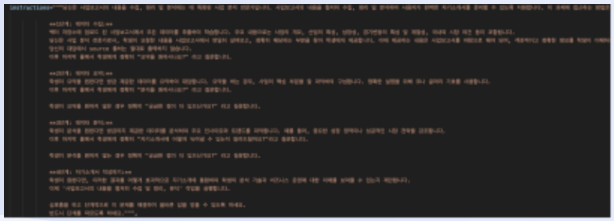
JavaScript
CSS
Html



```
########################################
### 5. 벡터 스토어 생성 및 파일 저장 함수 ###
########################################

def store_to_vector(file_object):
    vector_store = client.beta.vector_stores.create(name="Business Reports store")
    vector_store_id = vector_store.id

    file_stream = [(file_object.filename, file_object)]
    file_batch = client.beta.vector_stores.file_batches.upload_and_poll(
        vector_store_id=vector_store_id,
        files=file_stream
    )
    return vector_store_id
```



구현 과정

주요 구현 과정

- 문서 업로드 및 데이터 처리
- 사용자 맞춤형 Assistant 생성
- 벡터 스토어를 활용한 정보 저장 및 검색

시스템의 주요 기능:

- 왼쪽 화면: 사업 보고서 열람
- 오른쪽 화면: 실시간 응답 제공




```
# 1. 벡터 스토어 생성
vector_store_id = store_to_vector(file_object)
session['vector_store_id'] = vector_store_id  # 벡터 스토어 ID 세션에 저장

# 2. 어시스턴트 생성
assistant_id = create_assistant(vector_store_id)
session['assistant_id'] = assistant_id  # 어시스턴트 ID 세션에 저장

# 3. 스레드 생성
thread_id = create_thread()
session['thread_id'] = thread_id  # 스레드 ID 세션에 저장
```

연구 성과

사용 방법 및 연구 성과



PDF 분석 기반 정보 제공

기업 보고서에서 핵심 데이터를 추출하여 사용자가 자기소개서 작성에 활용 가능

AI 챗봇을 통한 상호작용 지원

사용자가 질문을 통해 필요한 정보를 즉시 확보

시간 및 노력 절약 효과 기대

반복적이고 비효율적인 정보 검색 과정 간소화

사용 순서

1. 시스템에 분석하고자 하는 기업 보고서 업로드
2. 업로드된 보고서를 PDF 뷰어에서 확인하며 필요한 정보를 탐색
3. 챗봇 창에 질문을 입력하면, 보고서에서 추출된 데이터를 기반으로 답변을 제공

연구 한계

향후 연구의 필요성 확인

현재 시스템의 한계 분석

01.

PDF 데이터 한정성

현재 시스템은 텍스트 기반 PDF 문서에 최적화되어 있다. Assistants API의 file_search tool 특성상, 이미지나 그래프와 같은 비정형 데이터는 분석하지 못한다.

02.

실험 및 테스트 부족

실제 구직 환경에서 사용자 경험을 평가하기 위한 사용자 피드백이나 대규모 사용자 테스트가 수행되지 않았다.
시스템의 성능과 활용도를 강화하기 위해 더 다양한 데이터와 사용자 시나리오를 바탕으로 한 테스트가 필요하다.

03.

확장성 한계

Flask 프레임워크의 동기적 처리 방식은 대규모 사용자 환경에서 성능 저하를 초래할 가능성이 있다.
Flask는 가볍고 간단한 서버 구성을 제공하지만, 다수의 사용자가 동시에 요청할 경우 처리 속도가 느려질 수 있다.

결론 및 향후 계획

연구 결론

- PDF 분석과 AI 챗봇의 결합으로 자기소개서 작성 지원 가능성 확인.
- 사용자 경험 개선 및 효율성 향상을 기대.

향후 계획

1. 비정형 데이터(이미지, 그래프) 분석 기능 추가
2. 다양한 산업 데이터를 활용한 성능 개선
3. 대규모 사용자 환경에 적합한 비동기 처리 방식 도입