

손미란 빅데이터융합학과/2023511050
엄다안 빅데이터융합학과/2025511040
유인호 빅데이터융합학과/2025511053
이 람 빅데이터융합학과/2024512011

2025.06.17

[Note.]

1) 새로운 장표를 만드실 경우 : 바로 만들어주세요.(1순위)

2) 기존 장표에 직접 수정을 원하실 경우 : 바로 반영하지 말고 원본 피피티를 똑같이 뒷장에 복사하신 후,
수정을 해주세요. 또 우측 상단에 사각형 붙여서, 이름이랑 변경항목 대략 써주세요.
수정을 할때는 예쁘게 하시려고 안해도 됩니다. 대략 해두시면 제가 손볼게요.

페이지 장표가 늘어나는 것에 대해 우려하지 마세요! 발표장표는 있되 skip하는게 오히려 더 낫습니다.
다만, 6분 이내 발표 완료 해야하므로, 상황에 따라 간소하게 전달할 순 있습니다.

유권자 나침반

대통령 후보 공약 기반 유권자 AI에이전트

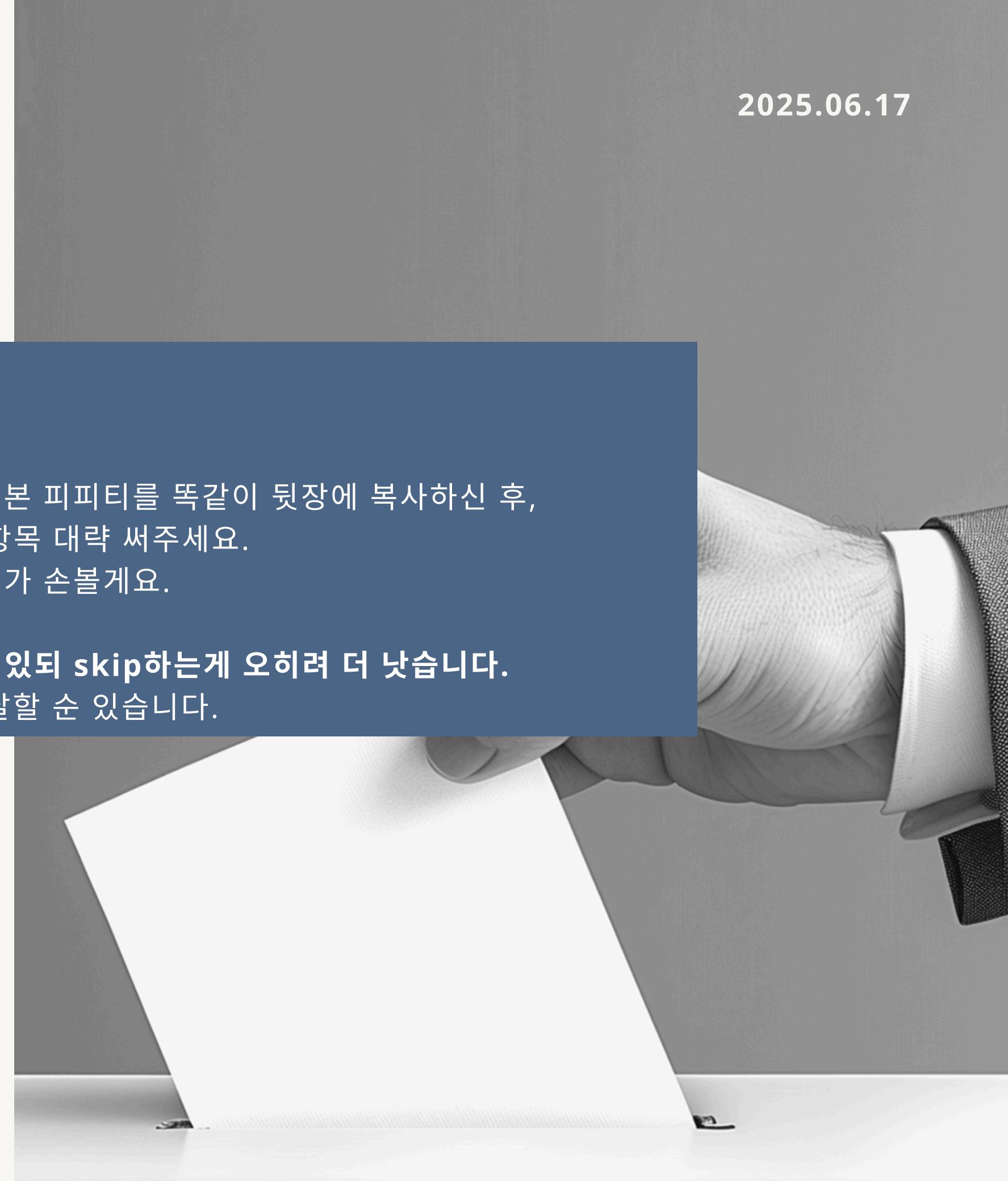


Table of Contents

- 01** 프로젝트 개요
- 02** 데이터 수집 및 전처리
- 03** 서비스 주요 기능
- 04** 시스템 개요 및 상세
- 05** 프롬프트 설계 전략
- 06** 실험 및 데모 결과
- 07** 성능 개선 및 한계점
- 08** 결론 및 향후 발전 방향

01 프로젝트 개요 | 대통령 후보 공약 기반 유권자 AI에이전트

서비스 목적

- 유권자가 쉽고 빠르게 후보의 공약을 이해할 수 있도록 지원
- 후보의 정책 별 질의응답, 정책 비교, 개인 상황 맞춤형 추천 통해 정보 격차 해소
- LLM 기반 에이전트 통해 복잡한 공약 문서를 쉽게 요약 및 설명

타겟 사용자

- 자신의 상황에는 어떤 공약이 유리한지 모르는 유권자
> 청년 취준생, 신혼부부, 무주택 중산층 등
- 정책을 확인할 시간이 부족한 유권자. 많은 내용 중 핵심을 빠르게 알고 싶은 사람.
> 바쁜 직장인, 하루 전날 후보를 결정해야하는 사람
- 후보 별 정책을 알고 싶으나, 자극적인 정치적 발언이나 이벤트로 매스미디어(유튜브 등)를 신뢰하지 않는 시민,

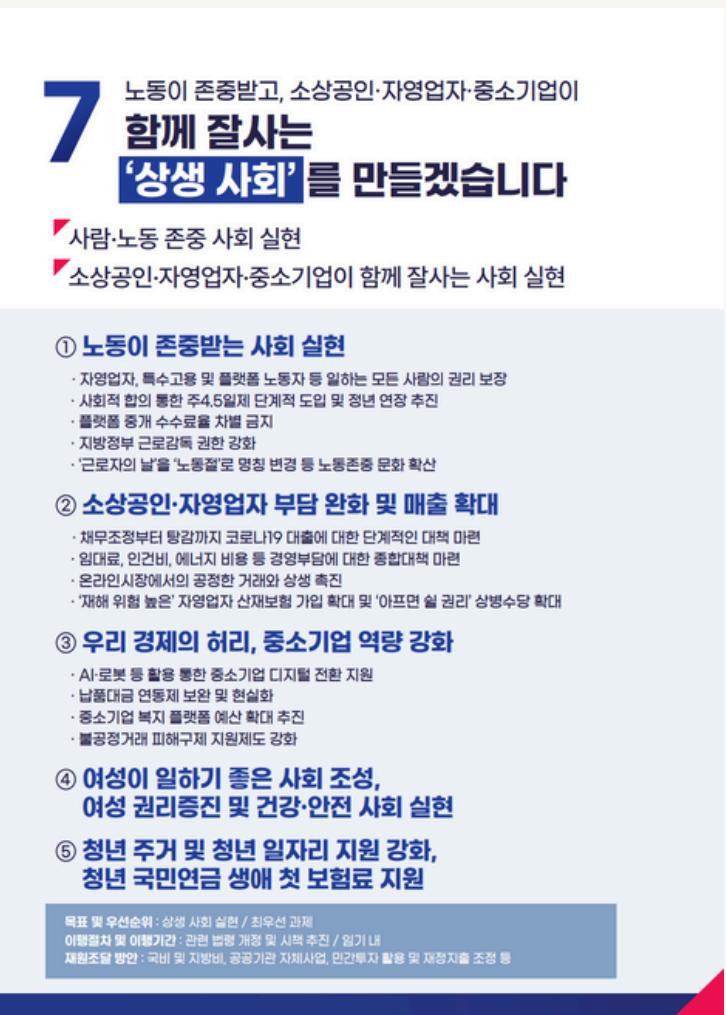
01 프로젝트 개요 | 팀원 구성 및 역할

구성원 (가나다순)	역할
손미란	
엄다안	
유인호 (팀장)	
이람	

02 데이터 수집 및 전처리

데이터

- 후보별 정책 문서(PDF)
 - 네이버 뉴스기사 API
- 권영국, 김문수, 송진호, 이재명, 이준석, (황교안) 선거공약서 및 정책



제21대 대통령선거					
후보자명	이재명	기호	1	소속정당명	더불어민주당
정책순위 : 9	저출생·고령화 위기를 극복하고 아이부터 어르신까지 함께 살보는 국가를 만들겠습니다.			분야	교육·복지

□ 목표
○ 저출생·고령화 해소 및 돌봄체계 구축

□ 이행방법
○ 저출생 대책 혁신 및 자녀양육 지원 확대
- 자녀 수에 비례한 신용카드 소득공제율·공제 한도 상향 추진
- 초등학생 예체능학원·체육시설 이용료를 교육비 세액공제 대상에 추가
- '우리아이자립펀드' 단계적 도입 및 신혼부부 결혼출산지원 확대
- 신혼부부 공공임대주택·공급 확대
- 난임부부 치료지원 강화
○ 아이 키우기 좋은 나라를 위한 돌봄·교육, 일·가정 양립 지원 강화
- 공공 아이돌봄 서비스 지원 강화
- 자자체 협력형 초등돌봄 추진
- 초등학교 방과후학교 수업료 지원 확대
- 교육·보육의 질을 높이는 정부 책임형 유보 통합 추진
○ 발달장애인 24시간 돌봄 등 장애인 맞춤형 지역돌봄체계 구축
○ 생애주기별 외로움(고독) 대응 정책 개발·추진
○ 고령사회 대응을 위한 통합적 지원체계 마련
- 치매·장애 등으로 재산 관리가 어려운 노인을 위한 공공신탁제도 도입
- 어르신 주거 문제 해결을 위한 고령자 친화 주택·은퇴자 도시 조성
- 간호·간병 통합서비스 확대 및 요양병원 간병비 건강보험 적용
- 노인 등이 집에서 의료·돌봄서비스를 받는 지역사회 통합 돌봄체계 구축
○ 지속 가능한 노후 소득 보장 체계 구축
- 국민연금 수급 연령에 맞춘 정년 연장, 사회적 합의를 통해 단계적 추진
- 주택연금 제도개선 등을 통해 노후 소득 보장을 위한 지원강화

Step 1. PDF 문서 로딩 및 후보 메타데이터 부여

- PyMuPDFLoadera로 공약 PDF에서 텍스트 데이터 추출
- 문서 별 후보자명을 metadata["candidate"]로 부여
>> 추후 RAG 검색 시 후보자 필터링 가능

Step 2. 문서 분할 Chunking

- 문맥 연결성을 유지하면서도 LLM이 처리 가능한 길이로 분할
- RecursiveCharacterTextSplitter 사용
- Setting : chunk_size=1000, chunk_overlap=200

Step 3. 임베딩 및 벡터화

- Ko-SBERT(jhgan/ko-sbert-nli) 모델로 문서 임베딩
- FAISS 벡터 DB에 저장하여, 향후 유사도 기반 검색 지원 *Trial1
- Chroma로 전환, "candidate"별 필터링 용이한 Retriever *Trial2

03 서비스 주요 기능

사용자가 복잡한 공약을 쉽게 이해하고, 상황에 맞는 적합한 후보를 선택하도록 4가지 피쳐를 제공합니다.



단일 후보자 정책 질의응답

'OOO 후보는 이 문제를
어떻게 바라볼까?



복수 후보자 정책 비교분석

'특정 주제에 대해 후보 별
공약은 어떤 차이가 있지?'



사용자 컨텍스트 별 질의응답

'내 상황을 개선해준다는
후보자가 대체 누구야?'



정책에 대한 여론 반응분석

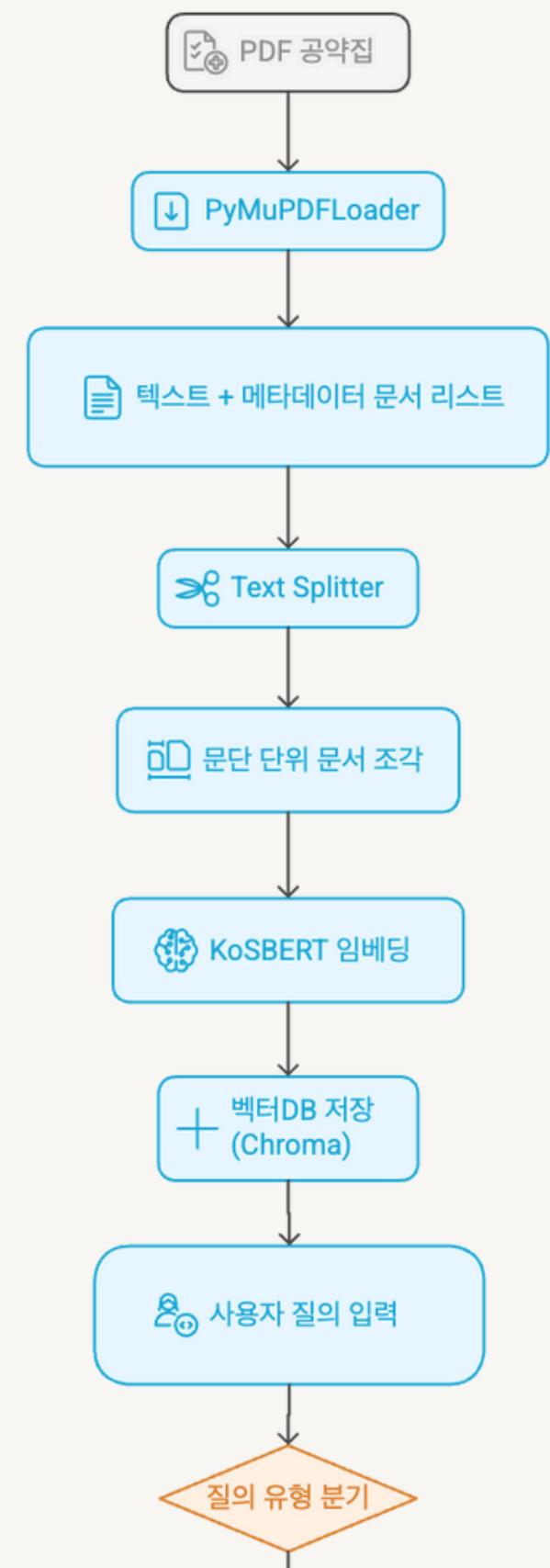
'대중의 합의가 잘 이뤄져야
빨리 추진할 수 있어.'

04 시스템 개요 | 활용 기술 스택

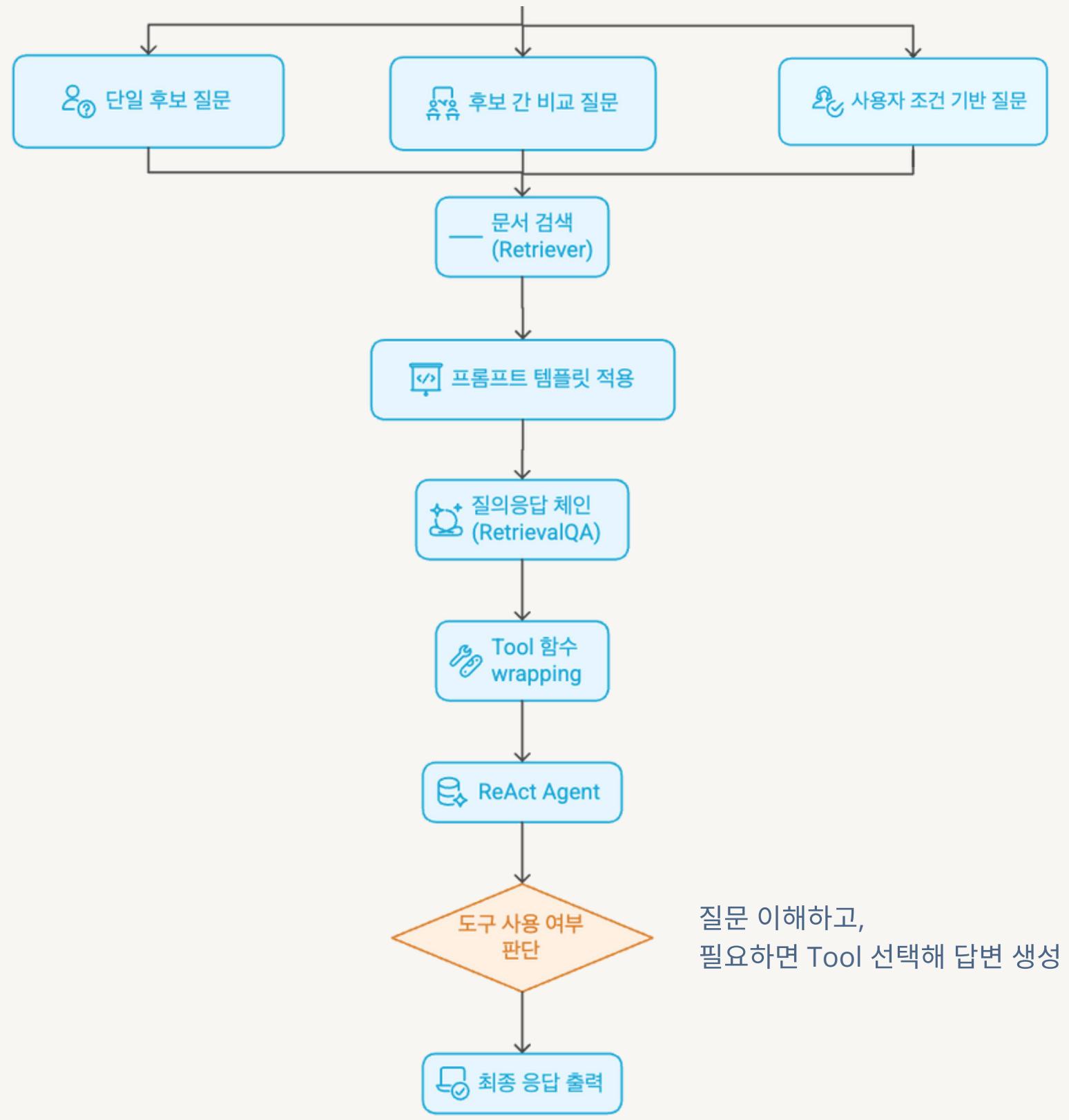
기술 스택	적용	프로젝트 내 적용 방식
RAG (검색 + 생성)	O	RetrievalQA로 검색 기반 응답 생성, 문서 → 검색 → 응답 연결 구조
Meta Prompt	O	PromptTemplate 사용해 {context}와 {question} 변수로 구성된 일반화 프롬프트 활용 * Trial1 PromptTemplate 사용해 단일 후보 질의, 후보 간 비교 질의, 사용자 맞춤형 추천 질의 형태로 구체화 * Trial2
Prompt Chaining	O	문서 검색 → 응답 생성을 단계적으로 처리하는 구조. 명시적 chaining 코드는 없지만 단계적 흐름.
Function Calling	O	Tool(func-qa_chain, run) 형태로 QA체인을 함수처럼 래핑
ReAct	O	Initialize_agent(...,agent=ZERO_SHOT.REACT_DESCRIPTION)로 LLM이 도구를 선택하고 실행
Role Playing Prompt	O	Naver뉴스 API 연동 시에만 사용 명시적 역할 부여 (예: “너는 뉴스 분석 전문가야.”)

04 시스템 개요 | 구조 및 아키텍처

데이터 전처리 및 인덱싱 파이프라인



사용자 질의응답 처리 파이프라인



04 시스템 개요 | 특장점 및 개발환경 w/Github

- a. 시스템의 특장점
- b. 시스템의 개발환경 및 설정
- e. 개발 소스코드 (gitup URL, 저장소는 public으로 설정)
 - a. API 목록 명세 (어차피 네이버 밖에 없어서, 여기에 같이 몰아넣어요)

04 시스템 상세 | 주요 핵심 모듈 알고리즘

어떤걸 써야할까요?

06 프롬프트 설계 전략 | 질의 유형 별 프롬프트

단일 후보 질의

```
single_candidate_policy_prompt = PromptTemplate(  
    input_variables=["summaries", "question"],  
    template="")
```

아래 문서는 대통령 후보의 공약이다.

질문에 대해 주의사항을 고려하고, 다음 요소를 포함하여 답변하라:

주의사항: 반드시 문서의 내용에 기반하여 답하라.

문서에 없는 내용은 추측하지 말고 답하지 마라. 반드시 한국어로 답변할 것

- 정책의 목적
- 구체적 수단 (시설, 제도, 법안 등)
- 실행 대상 또는 지역
- 문서상 등장한 구체적인 단어(용어)를 사용

문맥:
{summaries}

질문:
{question}

답변:
")

- 추측 금지
- 한국어 고정

후보 간 비교 질의

```
multi_candidate_comparison_prompt = PromptTemplate(  
    input_variables=["topic", "comparisons"],  
    template="")
```

다음은 '{topic}'에 대한 대통령 후보 공약 요약이다.

다음 기준에 따라 자세히 비교하라:

1. 정책의 목적 비교
2. 구체적 수단 비교
3. 실행 대상 또는 지역 비교
4. 문서상 등장한 구체적 용어 비교

아래 형식을 유지하고 문장을 요약하지 마시오. 반드시 문단 단위로 상세히 작성하고, 후보별 차이점을 구체적으로 명시하시오.

후보별 공약:
{comparisons}

반드시 한국어로 작성하시오.
")

- 문단 단위 비교 (정책목적, 수단, 실행대상, 용어)
- 차이점 강조

사용자 맞춤형 질의

```
#  사용자 프로파일 추출  
user_profile_extraction_prompt = PromptTemplate(  
    input_variables=["question"],  
    template="")  
다음 질문에서 사용자 프로파일을 추론하시오:  
{question}
```

- 연령:
 - 직업/소득:
 - 주거 상태:
 - 관심 정책 키워드:
- ")

```
#  추천용 프롬프트  
candidate_recommendation_prompt = PromptTemplate(  
    input_variables=["question", "profile", "summaries"],  
    template="")  
[질문]  
{question}  
[프로파일]  
{profile}
```

[후보별 공약 요약]
{summaries}

이 정보를 바탕으로 가장 적합한 후보를 한 명 추천하라. 한 문단으로 이유 포함.
- 추천 후보:
- 추천 이유:
")

- 프로파일 추론 : 연령, 직업, 주거상태, 관심 주제
- 후보 추천 : 사용자 프로파일 기반 1명 추천

06 프롬프트 설계 전략 | 질의 유형 별 프롬프트 w/실시간 정보 활용

현재 작성중
논의 필요

API



활용방식 변경

V1. * Trial 1

사용자 질의가 후보 공약집
내 부재 시 (RAG결과 부재)



V2. * Trial 2

사용자 질의가 신뢰성, 실
행 상태, 논란 포함 시

v특징1

- Ipsum ad enim aliqua et ex do excepteur
- Ipsum ad enim aliqua et ex do excepteur

특징2

- Ipsum ad enim aliqua et ex do excepteur
- Ipsum ad enim aliqua et ex do excepteur

특징3

- Ipsum ad enim aliqua et ex do excepteur
- Ipsum ad enim aliqua et ex do excepteur

```
def summary(text: str, keyword: str):
    prompt = f"""
너는 뉴스 분석 전문가야. 다음은 뉴스 기사 전문이야.

[뉴스 기사 전문]
{text}

이 기사에서 "{keyword}"와 관련된 내용이 있다면, 관련된 내용을 요약해서 알려줘.
만약 관련 내용이 없다면 부가 설명 없이 "관련 없음" 이란 단어만 말해줘.

"""

return gpt_prompt_action(prompt, 300)

def news_filter(news_list: list, search_word: str) :
    filter_ls = [x.replace('\"', '') for x in news_list if '관련 없음' not in x]
    count = len(filter_ls)
    concat_text = '\n'.join(filter_ls)
    prompt = f"""
아래 문장들을 보고 {search_word} 주제 기준으로 긍정적인지 부정적인지 알려줘.

{concat_text}

답변은 부가 설명없이 아래 list안에 json 형식을 담아서 답변해줘.
요소인 json 형식은 아래와 같아. 총 {count}개 문장이니 리스트에 요소 확실히 개수 맞춰서 대답해줘.

"""

    add_prompt = """
{
    'num' : '위에 나오는 문장의 순서',
    'sentiment' : '긍정 or 부정'
}

답변은 꼭 리스트로 해줘
"""

    return gpt_prompt_action(prompt+add_prompt, 4000)
```

06 프롬프트 설계 전략 | 질의 유형 별 프롬프트 (누락 건)

이람 : 보완 필요

하기 2가지 프롬프트 어떤 맥락으로 작성할지 논의 예정

```
#  비교용 상세 분해 프롬프트
policy_element_extraction_prompt = PromptTemplate(
    input_variables=["question", "summary"],
    template="")
```

다음은 어떤 대통령 후보의 공약 요약 내용입니다:

[질문]
{question}

[공약 요약]
{summary}

아래 항목들을 해당 공약 요약에서 가능한 한 구체적으로 추출하십시오:

- 핵심 공약: 핵심 아이디어 한 줄 요약
- 실현 방식: 구체적인 실행 수단, 제도, 구조
- 강점: 타 후보 대비 돋보이는 차별점이나 이점

출력 예시는 다음과 같습니다:

핵심 공약: 청년에게 생애 첫 집 공급 확대

실현 방식: 공공임대 및 분양 확대, 저리 대출 제공

강점: 청년 세대에 직접적이고 독립적인 주거 안정 방안 제시

위와 같은 형식으로 3줄로 정리하십시오.

.....

)

```
#  관련성 판단 프롬프트
relevance_prompt = PromptTemplate(
    input_variables=["question", "answer"],
    template="")
```

다음은 사용자의 질문과 PDF에서 추출된 응답입니다.

[질문]
{question}

[응답]
{answer}

다음 기준에 따라 이 응답이 질문에 '정책적으로 직접적인 답변'인지 판단하시오:

판단 기준:

- 질문에 등장하는 정책 주제가 응답에 구체적으로 다루어지지 않으면 → "관련 없음"
- 질문과 응답의 유사도, 관련성이 없으면 → "관련 없음"
- 단순히 단어가 포함된 경우가 아닌, 해당 주제를 명확히 다른 정책 목적과 수단이 있을 때만 → "관련 있음"

결론은 "관련 있음" 또는 "관련 없음" 중 하나로만 명확히 작성하십시오.

.....

)

```
translation_prompt = PromptTemplate(
    input_variables=["english_text"],
    template="다음 영어 텍스트를 자연스럽고 정확한 한국어로 번역하십시오:\n\n{english_text}\n\n번역:")
)
```

#후보자 이름 추출

```
candidate_detect_prompt = PromptTemplate(
    input_variables=["question"],
    template="다음 질문에서 언급된 대통령 후보의 이름을 정확히 추출하시오. 두 명 이상일 경우 모두 출력하시오.\n\n질문: {q}
)
```

07 실험 및 데모 결과

3가지 유형 별 스크립트 시각화하여 작성 예정 (단, 발표 시는 10초 내외. → 실제 데모로 넘어갈 확률 높음)

단, 최종 답변 정리 완료 시 작성 예정 (아직 남은것으로 알고 있음)
아닌 경우 바로 진행.

08 성능 개선 및 한계점

한계점 논의 필요

- 1) 제가 마지막에 실험 작성한 내용 중심으로 쓰면 될지?
- 2) + 챗 지피티가 틀린 pdf 쪽수 출처 예시 추가 (비교)

개선점 추가 또는 변경 필요

개선점 : chroma DB 변경, 네이버 api 사용방식 변경, 그 외

[Note]

각자 본인이 어필하고 싶은 부분(원래 이랬는데, 이걸 개선하기 위해 노력한 항목 및 결과)을 넣어도 좋겠어요.

09 결론 및 향후 발전 방향

자연어 처리

유권자 나침반 | 대통령 후보 공약 기반 유권자 AI에이전트

감사합니다

2025.06.17