

# Dr. Navi

사용자 증상 기반 진료과 추천 서비스



# CONTENTS

목차



01	<b>프로젝트 개요</b>	팀원 소개 / 개발 배경 및 목적 / 기대 효과
02	<b>시스템 구성</b>	Architecture / tech stack
03	<b>주요 기능 및 사용자 경험</b>	
04	<b>데이터 수집 및 처리</b>	
05	<b>RAG 시스템</b>	Architecture / Vector DB / Prompt Engineering
06	<b>성능 평가 및 분석</b>	RAGAS / Sampling
07	<b>배포 및 개선 방향</b>	배포 / 한계점 / 개선방향

## 프로젝트 개요

# 팀원 소개



김민철

보고서 작성  
테스트 코드 설계 및 작성



임동성

보고서 작성  
무작위 샘플링 및 정성평가



지용석

보고서 작성  
README 작성  
그래프 도출



최유정

RAG 파이프라인 설계  
데이터 크롤링  
PPT



허현준

Streamlit Frontend  
RAG 파이프라인 보완  
배포

## 프로젝트 개요

# Dr. Navi

의료 서비스 접근성 향상을 위한 LLM 및 RAG 기반 진료과 추천 서비스

## Why Dr. Navi ?

Dr. Navi의 이름은 'Doctor'와 'Navigator'의 결합입니다.  
의료 전문성을 상징하는 'Dr.'와 길잡이를 의미하는 'Navi'를 통해  
"의료 서비스의 길잡이가 되어주는 전문가"라는 의미를 담았습니다.  
복잡한 의료 시스템 속에서 환자들을 적절한 진료과로 안내하는 믿음직한  
의료 내비게이터를 지향합니다.



## 프로젝트 개요

## 개발 배경 및 목적

류마티스 내과는 만성 근골격계 질환 및 자가면역질환을 진료하는 분과!

<https://www.mediafine.co.kr/news/articleView.html?idxno=10284>

한편, 통풍은 대사 질환에 의해 발생하는 내과 질환임에도 불구하고 주 증상이 손과 발 등의 관절에 나타나기 때문에 통풍치료를 위해 류마티스내과가 아닌 정형외과를 찾는 이들이 많다.

이는 잘못된 의료 상식으로, 통풍의 근본적이고 올바른 치료방법을 위해 통풍진료과인 류마티스내과를 통해 진료받는 것이 적합하다.

대사 질환인 통풍은 조기에 발견하고 치료하지 않으면 관절의 변형이 심해져 회복이 어려울

## 의료 정보 접근성 문제

- 전문용어 이해도 부족
- 불필요한 병원 방문으로 인한 시간/비용 낭비
- 증상 방치로 인한 합병증 발생 위험 증가

## 의료 서비스 효율성 개선 필요

- 잘못된 진료과 방문으로 인한 진료 지연
- 의료 자원의 비효율적 사용
- 의료 서비스 질 저하

접근성  
향상RAG  
활용

배경

의료자원  
효율화

차별화

## RAG 기술 활용의 필요성

- 신뢰할 수 있는 의료 정보 기반 응답
- 복합 증상 처리의 어려움
- LLM의 진료과에 대한 이해도 부족

## 기존 서비스의 한계

- 병원 홈페이지: 단순 목록 나열
- 의료 어플: 단순 정보 제공 방식



=== Test Case 1 ===

증상: 유방에 뭔가 만져지고, 분비물이 나와.

답변:

[증상 분석]

유방에 덩어리나 분비물이 느껴진다면 유방암이거나 다른 유방 질환일 수 있습니다.

[추천 진료과]

1. 유방외과
2. 산부인과 (유방외과가 없을 경우)

[주의사항]

- 본 정보는 참고용이며, 정확한 진단을 위해서는 반드시 의사의 진찰이 필요합니다.
- 유방암의 조기 발견과 치료는 매우 중요합니다. 따라서 가능한 한 빨리 전문의를 찾아
- 유방암의 가족력이 있는 경우, 정기적인 검진이 중요합니다.
- 분비물이 나오는 경우, 색깔, 양, 냄새 등 자세한 정보를 의사에게 알려주세요.

## 프로젝트 개요

# 기대 효과

### 사용자 측면

정확한 진료과 선택으로 불필요한  
진료 시간 및 비용 절감

### 기술적 측면

향후 다양한 의료 서비스로의 확장 가능성



### 의료기관 측면

의료자원 활용 최적화  
: 불필요한 진료과 이동 및 의료진 업무 부담 감소

### 사회적 측면

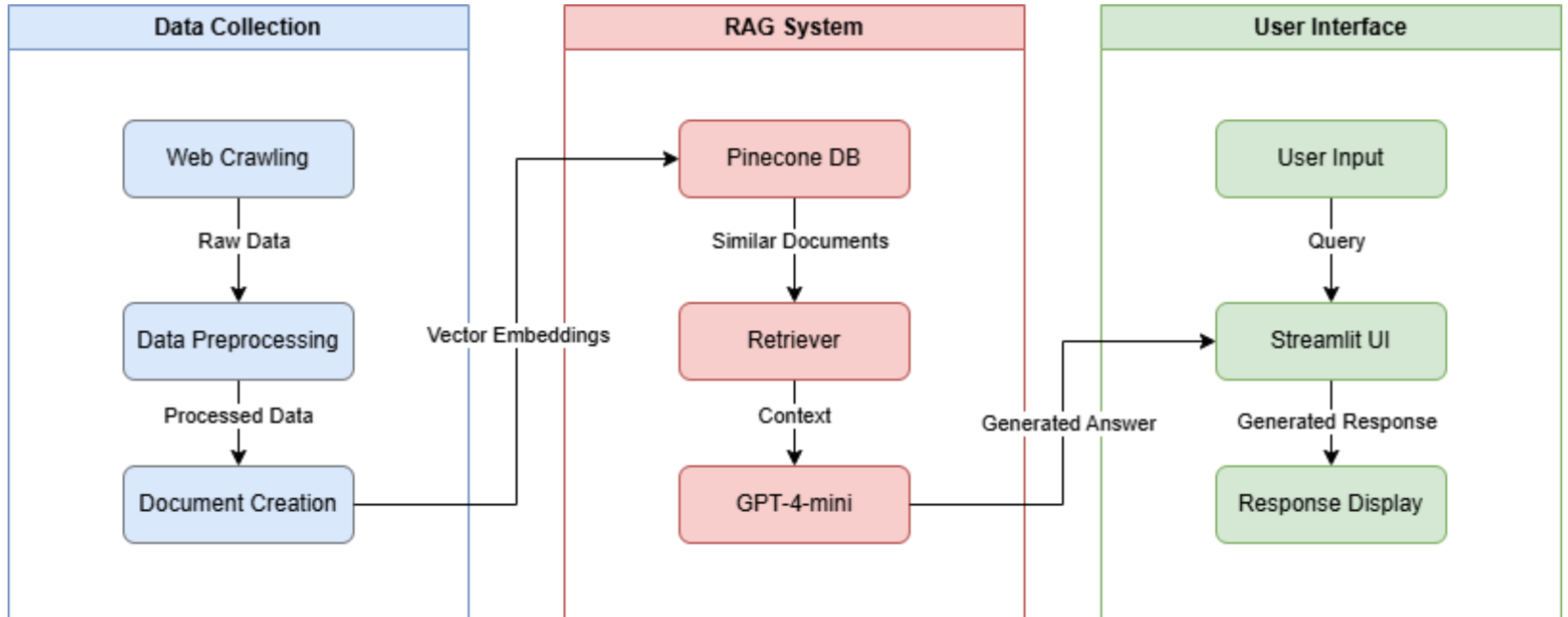
의료 서비스 질적 향상 및 의료 비용 절감

**적절한 진료과 선택 + 의료 자원 효율화**

의료 정보의 접근성 및 신뢰성을 향상시킴으로써 의료 서비스의 질적 향상 도모

시스템 구성

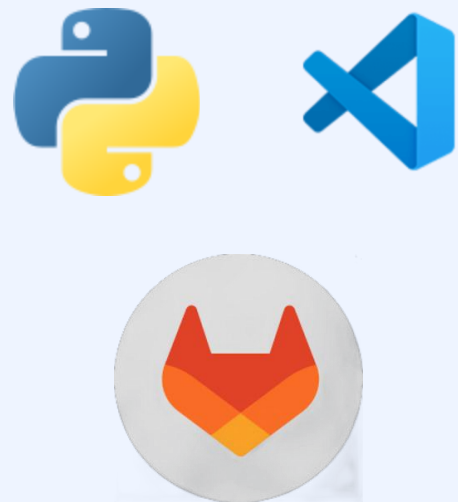
# Architecture



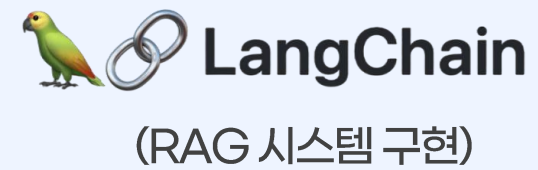
시스템 구성

# Tech Stack

## 개발 환경



## Library



## Database



## API

OpenAI GPT-4-mini

Upstage  
Embedding

## Frontend





## 주요기능 및 사용자 경험

# 주요 기능

### 01 | 증상 기반 진단



- 자연어 증상 입력
- GPT-4-mini 기반 분석

### 02 | 진료과 추천



- RAG 기반 정확도 향상
- 진료과 2개 추천
- 추천 근거 제공

### 03 | 맞춤형 응답



- 성별/나이 기반 개인화
- 의학용어의 쉬운 설명
- 단계별 응답 구조화

## 주요기능 및 사용자 경험

# 사용자 시나리오

### 상황 : 기침 있는 20살 남성

#### 1. 입력:

- 추가 정보: 성별 / 나이 입력
- 증상 입력: "기침과 가래가 나와요."

#### 2. 분석

- 기침, 가래와 관련된 의료 정보 검색
- 검색된 의료 정보 기반 분석
- 적합한 진료과 선정


#### 3. 진료과 추천

- 호흡기내과, 내과 추천
- 추천 근거 제시

증상  
입력


AI 분석

진료과  
추천

 **Dr. Navi**

의료 상담 챗봇입니다. 증상을 입력하면 진료과 추천, 주의사항 등을 안내해드립니다. 아래 정보를 입력해주세요.

팀원: 김민철, 임동성, 지용석, 최유정, 허현준

사용자 정보 입력 



안녕하세요! 의료 상담 챗봇입니다. 증상을 입력하면 진료과 추천, 주의사항 등을 안내해드립니다.

ex) 콧물과 기침이 나와요

증상을 입력해주세요...



[HTTPS://DRNAVI.STREAMLIT.APP/](https://drnavi.streamlit.app/)

## 주요 기능 및 사용자 경험

## Streamlit 기반 UI

The screenshot shows the initial user input form of the Dr. Navi app. The browser address bar displays 'https://drnavi.streamlit.app'. The app header includes the 'Dr. Navi' logo and a brief description: '의료 상담 챗봇입니다. 증상을 입력하면 진료과 추천, 주의사항 등을 안내해드립니다. 아래 정보를 입력해주세요.' Below this, the team members are listed: '팀원: 김민철, 임동성, 지용석, 최유정, 허현준'. The form is divided into two main sections. The first section, titled '사용자 정보 입력' (User Information Input), is highlighted with an orange border and contains a gender selection (radio buttons for '남성' and '여성', with '남성' selected) and an age input field (with '20' entered). The second section, titled '1. 증상 입력' (1. Symptom Input), is also highlighted with an orange border and contains a text input field with the placeholder '기침과 가래가 나와요' and a '2. 클릭!' (2. Click!) button next to it. A small '(옵션)' (Optional) label is visible next to the age input field.

The screenshot shows the results of the symptom analysis and recommendations. The browser address bar displays 'https://drnavi.streamlit.app'. The app header includes the 'Dr. Navi' logo and a brief description: '의료 상담 챗봇입니다. 증상을 입력하면 진료과 추천, 주의사항 등을 안내해드립니다. 아래 정보를 입력해주세요.' Below this, the team members are listed: '팀원: 김민철, 임동성, 지용석, 최유정, 허현준'. The results are displayed in a light gray box. At the top, it says '기침과 가래가 나와요' (Cough and phlegm). Below this, it says '[증상 분석] 기침과 가래는 호흡기 계통의 문제를 나타내는 일반적인 증상입니다. 이는 감기, 기관지염, 폐렴 등 다양한 호흡기 질환의 징후일 수 있습니다. 특히, 기침과 가래가 지속된다면 호흡기 질환의 가능성이 높습니다.' (Symptom Analysis: Cough and phlegm are common symptoms indicating a problem with the respiratory system. This can be a sign of various respiratory diseases such as cold, bronchitis, pneumonia, etc. In particular, if cough and phlegm persist, the possibility of respiratory disease is high). Below this, it says '[추천 진료과]' (Recommended Department) and lists two options: 1. 호흡기내과: 기침과 가래가 지속되는 경우, 호흡기 질환의 진단과 치료를 위해 호흡기내과를 방문하는 것이 적합합니다. 이 과에서는 폐렴, 기관지염 등 다양한 호흡기 질환을 전문적으로 다룹니다. 2. 내과: 일반적인 호흡기 증상에 대한 평가와 치료를 위해 내과를 방문할 수도 있습니다. 특히, 기침과 가래가 다른 증상(예: 열, 흉통 등)과 함께 나타나는 경우 내과에서 종합적인 진료를 받을 수 있습니다. Below this, it says '[주의사항]' (Precautions) and lists two items: • 본 정보는 참고용이며, 정확한 진단을 위해서는 반드시 의사의 진료가 필요합니다. • 기침과 가래가 심해지거나 호흡 곤란, 고열이 동반될 경우 즉시 응급실을 방문하세요. At the bottom, there is a text input field with the placeholder '증상을 입력해주세요...' (Please enter your symptoms...) and a right arrow button.

## 데이터 수집 및 처리

# 데이터 수집 및 처리

### Web Crawling

- 아산병원 질병백과 크롤링
- 질환 - 증상 - 진료과 데이터 수집
- BeautifulSoup, requests 활용

### Document 객체 변환

- Langchain Document 형식 변환
- 메타데이터 구성(질환명, 증상 목록, 진료과 목록)
- 벡터화 준비

데이터  
선정

#### 데이터 선정 이유

- 아산병원 질병백과 사이트
- 신뢰할 수 있는 의료 정보
- 체계적인 데이터 구조
- 질병-증상-진료과 연결 구조

아산병원 질병백과 링크

데이터  
수집

데이터  
전처리

#### 전처리

- 불필요한 공백 제거
- 중복 데이터 제거

데이터  
전처리

데이터  
분포

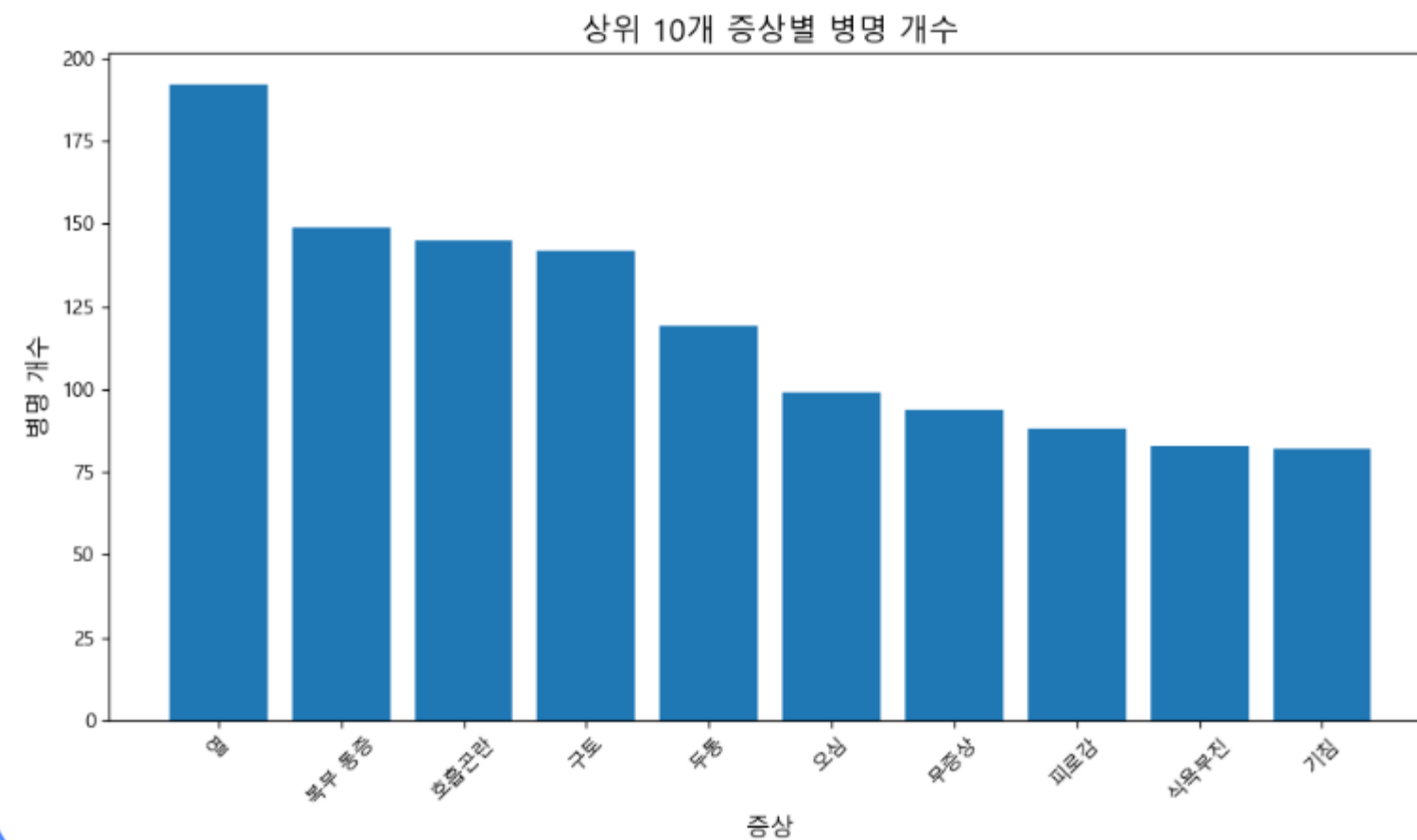
#### 통계

- 전체 데이터 수: 1577

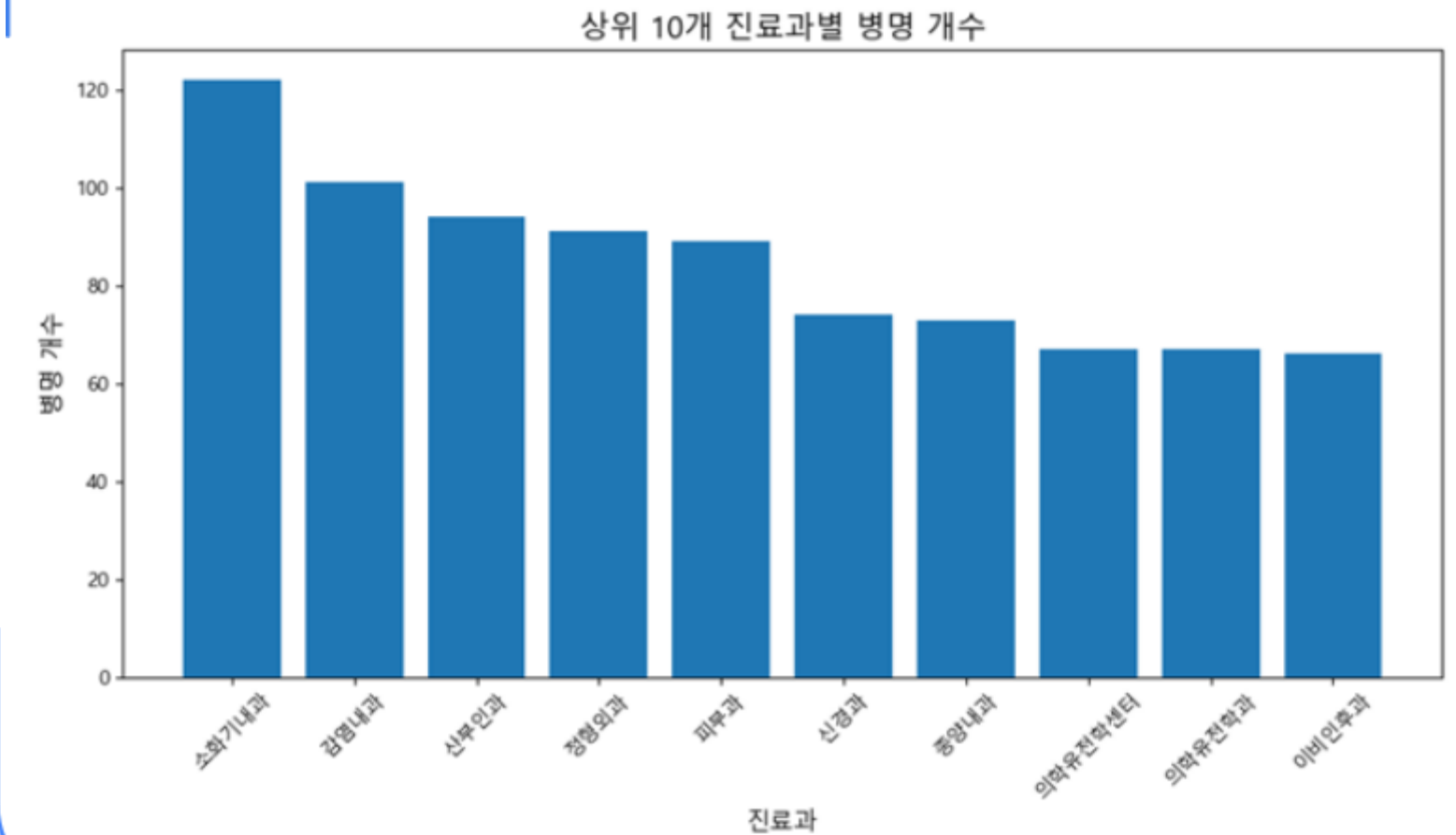
## 데이터 수집 및 처리

# 데이터 분포

### 상위 10개 증상을 포함하는 질환의 개수



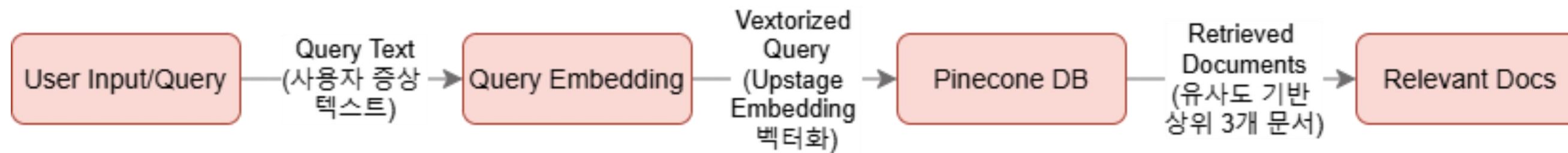
### 상위 10개 진료과에 포함되는 질환의 개수



## RAG 시스템

# Architecture

### Step 1 : Query Processing



### Step 2 : Response Generation



#### 주요 구성요소

- Retriever: 문서 검색
- Vector DB: 문서 저장
- GPT-4-mini: 응답 생성
- Context Builder: 맥락 구성

#### 시스템 특징

- 아산병원 데이터
- 실시간 벡터 검색
- 자연어 응답

# RAG 시스템 Vector DB

## Pinecone

### 1. 인덱스 설정

- 인덱스 명 : medical-chatbot
- Dimension : 4096
- embedding model : Upstage Embedding (embedding-query)
- Metric : cosine 유사도
- Serverless 환경 : AWS (us-east-1)

### 2. 데이터 구조

- Vector data : 문서 임베딩(4096)
- metadata : 질환명, 증상 목록, 진료과 목록

```
# Vectorstore
def get_vectorstore():
    index_name = 'medical-chatbot'

    # pinecone 초기화
    pc = Pinecone(api_key=os.getenv("PINECONE_API_KEY"))

    # 임베딩 모델 초기화
    embeddings = UpstageEmbeddings(
        api_key=os.getenv("UPSTAGE_API_KEY"),
        model="embedding-query"
    )

    # 테스트용 임베딩으로 차원 확인
    test_embedding = embeddings.embed_query("test")
    dimension = len(test_embedding)

    # pinecone index가 없다면 생성
    if index_name not in pc.list_indexes().names():
        print("새로운 인덱스 생성")
        pc.create_index(
            name=index_name,
            dimension=dimension, # 벡터 차원 수
            metric="cosine", # 벡터간 유사도 계산 방법
            spec=ServerlessSpec(cloud="aws", region="us-east-1") # 서버리스 사양 설정
        )
        # vectorstore 생성
        pinecone_vectorstore = PineconeVectorStore.from_documents(
            documents=crawl_disease_symptom(),
            index_name=index_name,
            embedding=embeddings
        )
    else:
        pinecone_vectorstore = PineconeVectorStore.from_existing_index(
            index_name=index_name,
            embedding=embeddings
        )
        print("기존 데이터 사용")

    return pinecone_vectorstore
```



## RAG 시스템

# Retriever

### Dense Retriever

#### Top-3 문서 선택 (k=3)

k=1 결과:  
평균 검색 시간: 5.869초

k=3 결과:  
평균 검색 시간: 4.772초

k=5 결과:  
평균 검색 시간: 6.407초

#### MMR(Maximum Marginal Relevance)

다양성과 관련성 동시 고려

```
# dense retriever 생성
retriever = vectorstore.as_retriever(
    # 유사도 정의
    search_type="mmr", # Maximum Marginal Relevance 검색 방식 사용
    # 검색할 쿼리 수 정의
    search_kwargs={"k": 3})

# 문서 체인 생성 (검색된 문서들을 결합하여 llm에 보낼 준비)
document_chain = create_stuff_documents_chain(llm, prompt)

# 최종 검색-생성 체인 생성
"""
검색 단계와 응답 생성 단계를 하나로 묶은 워크플로우
질문 -> 검색 -> 결합 -> 응답의 단계를 자동화하여 한 번에 수행
llm에 생성된 문서를 전달하고 답변을 생성
"""
retriever_chain = create_retrieval_chain(retriever, document_chain)
```



## RAG 시스템

## Prompt Engineering

## 최적화 전략

## 1. 환자 맞춤형 분석

- 성별, 나이 정보 활용
- 개인 특성에 따른 질병 가능성 고려

## 2. 사용자 친화적 응답 형식

- 증상분석, 추천 진료과, 주의사항 섹션 구분
- 전문 용어 쉽게 풀어서 설명

## 3. Few-shot Learning 활용

- 두 가지 예시 케이스 제공
- 적절한 응답 형식과 깊이 제공
- 모델의 응답 품질 향상

```
# system prompt
("system", ""다음 의료정보를 참고하여 답변해줘: {context}
너는 사용자의 성별, 나이, 증상을 종합적으로 고려하여 최적의 진료과를 추천하는 의료상담 챗봇이야.
다음 가이드라인을 따라 응답해줘:
1. [환자 특성 분석]:
  - 성별: {sex}
  - 나이: {age}세
  - 나이 및 성별에 따라 영향을 미치는 주요 질병을 고려해줘.

2. [증상 분석]:
  - 제시된 증상: {input}
  - 성별/나이를 고려한 종합적 질병 가능성 평가해줘.

3. [추천 진료과]:
  - 사용자 증상에 적합한 진료과를 최대 2가지 추천해줘.
  - 추천 이유와 근거 명확히 제시해줘.

4. [주의사항]:
  - 이는 참고용 정보이며, 정확한 진단은 의사의 진찰이 필요함을 명시해줘.
  - 응급 상황으로 판단되면 즉시 응급실 방문을 권고해줘.

5. [형식]:
  - 응답은 [증상 분석], [추천 진료과], [주의사항] 섹션으로 구분하여 제공
  - 전문 의학 용어는 사용자가 이해하기 쉽게 전문용어를 풀어서 설명해줘.
  - 검색된 문서에서 정보를 바탕으로 응답해줘.
""),
```

```
# few-shot prompting
("human", "기침과 열이 나고, 목이 간지럽고 아파."),
("ai", ""[증상 분석]
목의 불편감(간지러움 및 통증)과 함께 열이 동반된 상황입니다.
이는 상기도 감염(감기, 인후염 등)이나 목 관련 염증(편도염)일 가능성이 높습니다.

[추천 진료과]
1. 이비인후과: 목 통증 및 간지러움 증상에 대한 정확한 진단을 위해 방문하세요.
2. 내과: 열과 기침이 지속되거나 전신 증상이 있다면 내과 진료가 적합합니다.

[주의사항]
- 본 정보는 참고용이며, 정확한 진단을 위해서는 반드시 의사의 진료가 필요합니다.
- 고열(38도 이상)이 지속되거나 호흡 곤란이 있을 경우 응급실을 방문하세요.""

("human", "피부에 붉은 반점과 가려움이 있고, 발열이 있어."),
("ai", ""[증상분석]
붉은 반점과 가려움증은 피부염이나 알레르기 반응일 가능성이 있습니다.
발열이 동반된 경우, 감염성 질환(피부 봉와직염)이나 전신적인 알레르기 반응도 의심

[추천 진료과]
1. 피부과: 피부 반점과 가려움의 원인을 파악하고 적절한 치료를 받으세요.
2. 감염내과: 발열이 지속되거나 붉은 반점이 확산될 경우 감염 관련 진료가 필요합니다.

[주의사항]
- 발열이 심하거나 증상이 급격히 악화되면 즉시 응급실로 이동하세요.
- 알레르기 약 복용 후에도 증상이 개선되지 않으면 전문 진료를 권장합니다.""
```

# 성능 평가 및 분석 평가 방법

## ■ RAGAS (정량 평가)

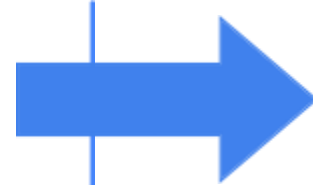
- context precision : 검색된 문서의 관련성
- context recall : 관련 정보 검색 비율
- faithfulness : 답변의 데이터 기반 신뢰도

## ■ Bert (정량 평가)

- F1 : 정답과 답변의 의미적 유사도

## ■ Sampling (정성 평가)

- 무작위 10개의 사용자 증상을 선택하여 답변 평가
- 다양한 의료 시나리오를 기반으로 질문 구성



## 평가 항목

정확성	생성된 답변이 저장된 의료 데이터와 얼마나 일치하는가?
관련성	답변이 검색된 의료 데이터와 관련이 있는가?
명확성	답변이 사용자가 이해하기 쉽고 논리적으로 명확한가?

# 성능 평가 및 분석

## RAGAS

정량평가

- Context percision : 0.9653
- Context recall : 0.5213
- Faithfulness : 0.4790
- Bert\_F1 : 0.9250

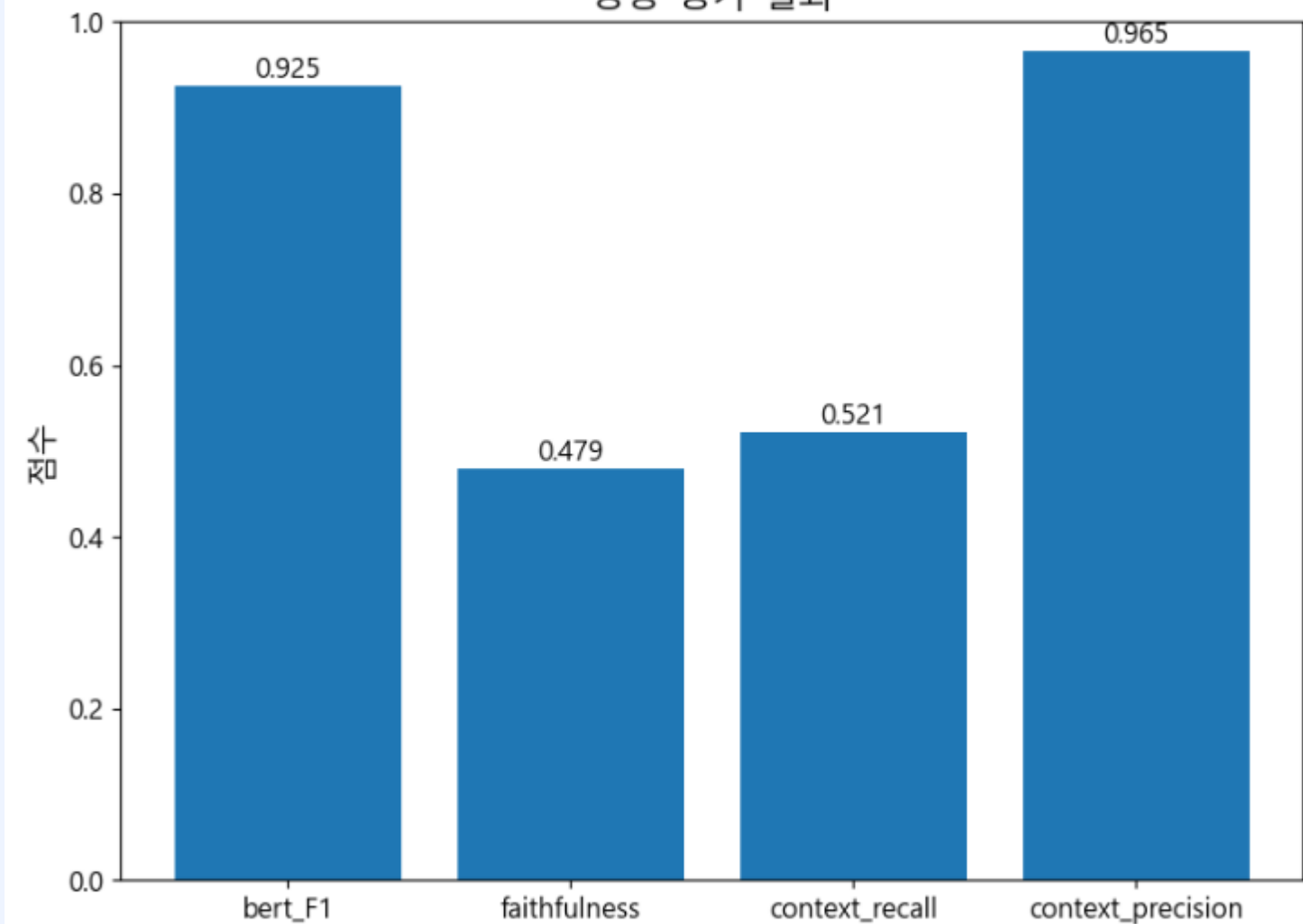
### 테스트 구성

- 테스트 케이스 : 12개 증상 조합 => 의사 자문 의뢰
- ground\_truth : 실제 정답
- 평가 방법 : RAG 평가 프레임워크

```
{
  "순서": 1,
  "성별": "남성",
  "증상 분석": "기침과 가래, 호흡 곤란은 호흡기 관련 질환에서 흔히 나타나는 증상입니다."
  "특히, 45세 남성의 경우 흡연, 알레르기, 만성 호흡기 질환 등의 가능성도 고려해야 합니다.",
  "추천 진료과 1": "호흡기내과 : 기침, 호흡곤란은 주로 호흡기관련 증상이므로,
  "호흡기내과에서 전문적인 진단과 치료를 받는 것이 적합합니다.",
  "추천 진료과 2": "알레르기내과: 호흡기 증상이 알레르기와 관련이 있을 가능성이
  "있으므로 알레르기내과 진료도 고려해 보시기 바랍니다.",
  "나이": 45,
  "증상": "기침, 가래, 호흡곤란"
}
```



정량 평가 결과



## 성능 평가 및 분석

# Sampling

정성평가

### 평가 절차

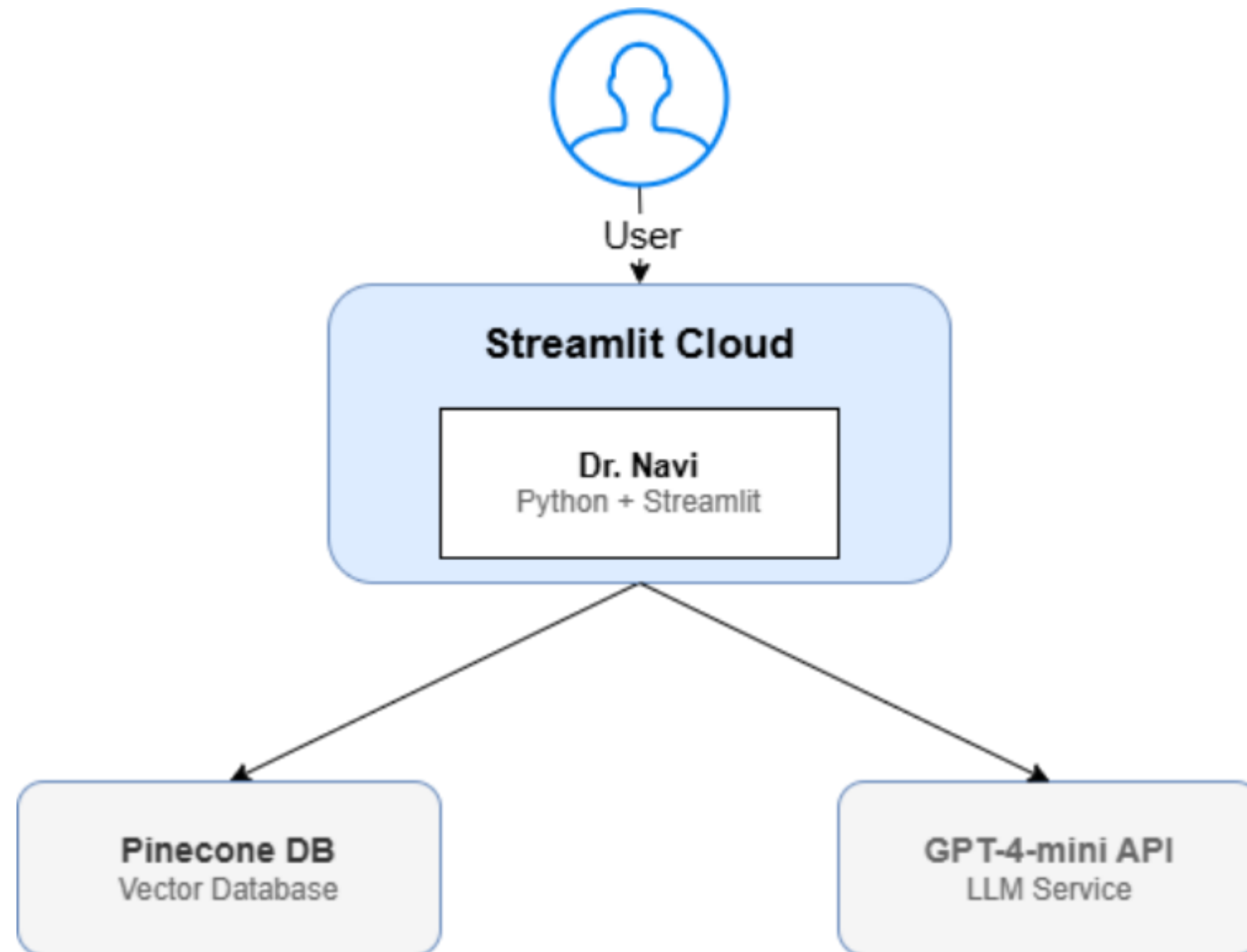
- 생성된 답변을 저장된 문서의 내용과 비교 검토
- 관련성이 낮은 답변은 피드백 기록 & 개선방안 도출
- 평가 결과를 바탕으로 생성된 답변의 장단점 정리 & 추가 최적화 방향 제시

### 평가 결과

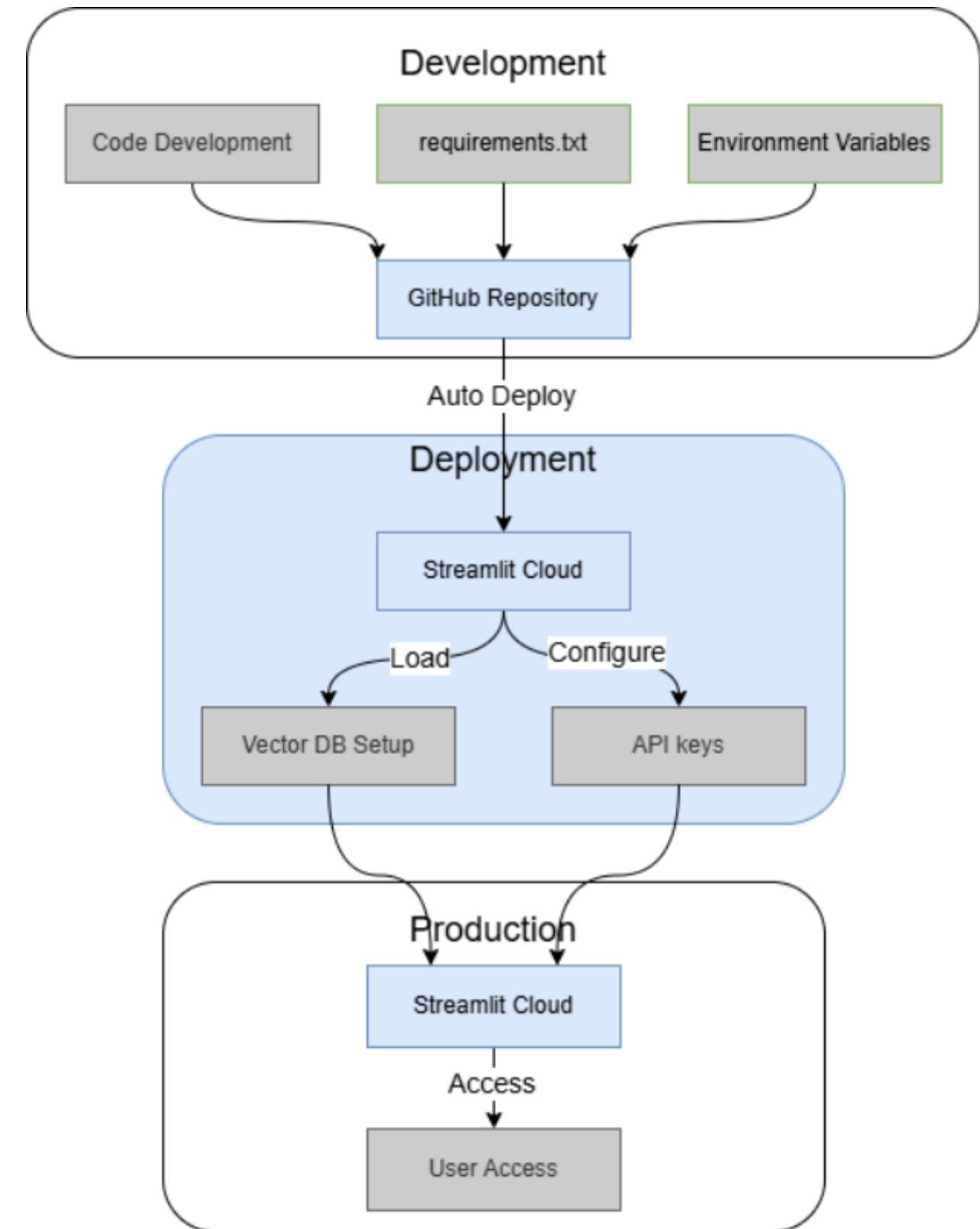
- 정확성: 10개의 질문 중 10개가 내용과 일치하는 답변 생성
- 관련성: 10개의 답변 중 10개가 검색된 문서와 밀접한 관련
- 명확성: 모든 답변이 문법적으로 정확하고, 이해하기 쉬운 표현으로 작성

## 배포 및 개선방향

## 배포



배포 링크 : <https://drnavi.streamlit.app/>



## 배포 및 개선방향 한계점



### 데이터 한계

- 단일 병원 데이터 사용
- 증상-진료과 매핑 정보 한계
- 실제 진단 사례 데이터 부재



### 시스템 한계

- 의료 전문가 검증 부재
- 단순 증상 위주의 매칭
- 복잡한 의학용어 처리 미흡



### 사용자 한계

- 응급상환 판단의 어려움
- 개인 맞춤형 추천 제한



### 검색 정확도 한계

- 단순 키워드 기반 검색의 한계
- 제한된 의료 데이터
- 복합 증상 처리 한계

## 배포 및 개선방향 개선방향



### 데이터 품질 개선

- 정기적인 의료 정보 업데이트
- 증상-질병 데이터 베이스 확장
- 의학 논문, 가이드라인 등 전문 자료 추가



### 시스템 고도화

- 나이/성별에 따른 가중치 적용
- 기저질환 고려
- 응급도 분류 시스템



### 사용자 경험 개선

- 진료과 정보 상세 설명
- 주변 병원 정보 안내 및 예약 시스템 연동
- 언어 지원 확대



### 검색 성능 고도화

- 검색 알고리즘 최적화
- 응답 품질 향상
- 복합 증상 처리 정확도 개선

*Thank You*  
**Q & A**