
NEWSPAPER

SUB-PROJECT : 빅3 상위 뉴스 요약 프로젝트

2024.10.04

배누리, 박현아, 안준영, 정기현

목차



01

프로젝트 목표

02

주요 코드

03

결과

04

TIME TABLE

프로젝트 목록

프로젝트 목표

1. 웹크롤링을 이용하여 3개 언론사의 주요 뉴스 3개에 대한 요약
2. Flask를 이용하여 웹페이지 제작
3. 랭체인과 streamlit을 이용한 챗봇 구현

언론사 : 동아일보, 중앙일보, 한겨레

주요 라이브러리

- BeautifulSoup, request
- newspaper > Article
- summa > summarize



주요 코드

1. news.py

```
def news():  
    # 언론사별 네이버 뉴스 검색 결과 페이지 URL (해당 언론사의 뉴스만 필터링한 결과 페이지)  
    new_url_list = ['https://search.naver.com/search.naver?where=news&query=%EB%89%B4%EC%8A%A4&sm=tab_opt&sort=0&photo=3&field=0&pd=0&ds=0',  
                   'https://search.naver.com/search.naver?where=news&query=%EB%89%B4%EC%8A%A4&sm=tab_opt&sort=0&photo=3&field=0&pd=0&ds=0',  
                   'https://search.naver.com/search.naver?where=news&query=%EB%89%B4%EC%8A%A4&sm=tab_opt&sort=0&photo=3&field=0&pd=0&ds=0',  
                   ]  
  
    # 언론사별 뉴스 상세 페이지의 URL 패턴 (기사 본문 URL을 필터링하기 위해 사용)  
    new_name_url_list = ['https://www.donga.com/news/', # 동아일보 기사 URL 패턴  
                        'https://www.joongang.co.kr/article/', # 중앙일보 기사 URL 패턴  
                        'https://www.hani.co.kr/arti/' # 한겨레 기사 URL 패턴  
                        ]
```

- 언론사별 네이버 뉴스 url 준비

1. news.py

```
news_summaries = []
# 각 언론사별로 반복문을 통해 주요 뉴스를 크롤링
for index, sec_url in enumerate(new_url_list):
    ## BeautifulSoup 객체 생성
    # 주어진 URL로부터 HTML 데이터를 가져와 파싱하는 함수
    def get_soup_obj(url):
        # User-Agent는 브라우저에서 보내는 요청처럼 보이게 하기 위한 헤더 정보
        headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36'}
        res = requests.get(url, headers=headers) # 주어진 URL에 HTTP GET 요청을 보냄
        # print(res.text) # URL에서 가져온 HTML 데이터를 출력 (디버깅 용도)

        # HTML 데이터를 lxml 파서를 사용해 BeautifulSoup 객체로 변환
        soup = BeautifulSoup(res.text, 'lxml')
        return soup

    # 해당 언론사 페이지의 상위 뉴스 HTML을 가져옴
    soup = get_soup_obj(sec_url)
```

requests

- 'requests' 라이브러리를 사용하여 원하는 웹사이트에 GET 요청을 보냄
- 웹페이지의 내용, 상태 코드, 헤더 등을 추출

BeautifulSoup

- HTML 및 XML 파일에서 데이터를 추출하기 위해 사용되는 파이썬 라이브러리
- res.text를 lxml 파서를 통해 HTML tag 구조로 내용을 해석

1. news.py

```
# 네이버 뉴스 검색 결과에서 뉴스 링크를 모두 추출 (div 태그의 class가 "group_news"인 곳에서 찾음)
url_list = soup.find('div', class_="group_news").find_all("a")
# HTML 구조상 div 태그 안에서 a 태그(링크)를 모두 가져옴

news_url_list = [''] # 뉴스 URL을 저장할 리스트 (처음에 빈 문자열로 초기화)

## 중복되는 링크 및 '#' 같은 불필요한 링크를 제외하고 원하는 뉴스 링크만 필터링
for url in url_list:
    # 링크가 특정 언론사(news_name_url_list[index])의 URL 패턴과 일치하는지 확인
    if url['href'][:len(new_name_url_list[index])] == new_name_url_list[index]:
        # 이전 링크와 중복되는 경우를 제외
        if (news_url_list[-1] == url['href']):
            continue
        else:
            news_url_list.append(url['href']) # 새로운 링크는 리스트에 추가
    else:
        continue

news_url_list.remove('') # 초기화했던 빈 문자열을 삭제
```

뉴스 상세 url들은 HTML 구조 속 #sp_nws1 > div > div > div.group_news, <a href에 있음
for문과 if 문으로 중복되는 링크 및 '#' 이나 뉴스링크가 아닌 링크를 제외하고 가져옴

1. news.py

```
# 내용 요약 함수 (기사 본문을 요약)
def summarize_content(content):
    try:
        # 본문 길이가 300자 이상일 때만 요약 (짧은 본문은 요약하지 않음)
        if len(content) > 300:
            summary = summarize(content, ratio=0.3) # summa 라이브러리를 사용하여 요약 (비율: 30%)
            summary = summary.replace('광고', '')      # 광고 단어 제거
        else:
            summary = content # 본문이 짧을 경우 원문 그대로 반환
    except ValueError:
        summary = '원문 없음' # 요약이 불가능한 경우의 예외 처리
    return summary
```

summa > summarize 함수를 이용하여 뉴스 내용을 요약하는 함수 정의

- 원본의 30%로 요약 비율 설정
- 텍스트가 300자보다 작으면 요약하지 않음
- 광고 단어 제거

1. news.py

```
# 크롤링한 뉴스 URL들에 대해 기사 내용을 가져오고 요약
for url in news_url_list[:3]:
    # newspaper 모듈을 사용할 때 타임아웃 오류 방지를 위한 설정
    user_agent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36'
    config = Config() # newspaper3k 모듈의 Config 객체 생성
    config.browser_user_agent = user_agent # 브라우저의 User-Agent 설정
    config.request_timeout = 20 # 요청 타임아웃 설정

    # newspaper 모듈을 사용하여 기사 다운로드 및 파싱
    article = Article(url, language='ko', config=config) # 한국어 설정
    article.download() # URL에서 기사를 다운로드
    article.parse() # 기사 내용을 파싱하여 title과 text 속성을 채움
    title = article.title # 기사 제목
    contents = article.text # 기사 본문 내용
    summary = summarize_content(contents) # 본문을 요약
    news_summaries.append({
        'title': title, # 기사 제목 저장
        'summary': summary # 요약된 기사 내용 저장
    })
return news_summaries
```

앞에서 가져온 뉴스 url들에 대해 각 기사 내용 가져오기 및 요약

- newspaper 패키지의 Article 모듈을 사용하여 download() 뉴스를 다운받고 parse() 파싱해주고 제목과 본문을 불러옴

이전에 정의한 summarize_content 함수로 뉴스 본문을 요약하고 제목과 요약 내용을 news_summaries 리스트에 저장 후 return

2. app.py

```
from flask import Flask, render_template
from news import news
import datetime

app = Flask(__name__)

@app.route('/')      # 애플리케이션의 기본 URL(http://localhost:5000/)에 대한 요청이 들어오면 index 함수를 실행
def index():
    result = news()    # news 함수가 호출되어 뉴스 데이터를 가져옴
    data = datetime.date.today()    # 현재 날짜를 가져옴
    result = [result, data]
    return render_template('index.html', result=result)    # templates라는 이름의 폴더를 기본 경로로 설정, templates 폴더 속 index.html 파일에 작성된 html 내용을 출력해서 보여줌

if __name__ == '__main__':    # 해당 스크립트가 직접 실행될 때만 조건문 안의 코드를 실행, 파일이 직접 실행될 때는 Flask 서버가 실행되지만, 다른 파일에서 모듈로 임포트될 때는 서버가 실행되지 않음
    app.run(debug=True)    # 디버그 모드로 Flask 서버를 실행, 코드를 수정하면 서버를 자동으로 재시작하고, 에러 발생 시 디버그 정보를 제공
```

파이썬 기반 웹 프레임워크중 하나인 Flask를 이용하여 웹페이지 제작
news.py 의 news 함수를 import 해주어 news 함수를 호출 해 뉴스 요약 데이터를 가져옴

3. index.html

```
<body>
  <h2>빅 3 상위 뉴스 요약</h2>
  <h3 class="date">date : {{result[1]}}</h3>
  <form action="/" method="get">
  </form>
  {% if result is not none %}
    <h3>1. 동아일보</h3>
    <p>
      <t1> 제목 : {{result[0][0]['title']}} <br>
      요약 : {{result[0][0]['summary']}} </t1><br>
    </p>
    <p>
      <t1> 제목 : {{result[0][1]['title']}} <br>
      요약 : {{result[0][1]['summary']}} </t1><br>
    </p>
    <p>
      <t1> 제목 : {{result[0][2]['title']}} <br>
      요약 : {{result[0][2]['summary']}} </t1><br>
    </p>
  {% endif %}
  <h3>2. 중앙일보</h3>
  <p>
    <t1> 제목 : {{result[0][3]['title']}} <br>
    요약 : {{result[0][3]['summary']}} </t1><br>
  </p>
  <p>
    <t1> 제목 : {{result[0][4]['title']}} <br>
    요약 : {{result[0][4]['summary']}} </t1><br>
  </p>
  <p>
    <t1> 제목 : {{result[0][5]['title']}} <br>
    요약 : {{result[0][5]['summary']}} </t1><br>
  </p>
  <h3>3. 한겨레</h3>
  <p>
    <t1> 제목 : {{result[0][6]['title']}} <br>
    요약 : {{result[0][6]['summary']}} </t1><br>
  </p>
  <p>
    <t1> 제목 : {{result[0][7]['title']}} <br>
    요약 : {{result[0][7]['summary']}} </t1><br>
  </p>
  <p>
    <t1> 제목 : {{result[0][8]['title']}} <br>
    요약 : {{result[0][8]['summary']}} </t1><br>
  </p>
</body>
```

```
<h3>2. 중앙일보</h3>
<p>
  <t1> 제목 : {{result[0][3]['title']}} <br>
  요약 : {{result[0][3]['summary']}} </t1><br>
</p>
<p>
  <t1> 제목 : {{result[0][4]['title']}} <br>
  요약 : {{result[0][4]['summary']}} </t1><br>
</p>
<p>
  <t1> 제목 : {{result[0][5]['title']}} <br>
  요약 : {{result[0][5]['summary']}} </t1><br>
</p>
<h3>3. 한겨레</h3>
<p>
  <t1> 제목 : {{result[0][6]['title']}} <br>
  요약 : {{result[0][6]['summary']}} </t1><br>
</p>
<p>
  <t1> 제목 : {{result[0][7]['title']}} <br>
  요약 : {{result[0][7]['summary']}} </t1><br>
</p>
<p>
  <t1> 제목 : {{result[0][8]['title']}} <br>
  요약 : {{result[0][8]['summary']}} </t1><br>
</p>
```

HTML을 작성하여 각 언론사 이름과 3개의 뉴스 제목과 요약을 보여지게 함

4. 랭체인 핵심 개념 4가지

Chain

- 언어 모델을 연결하여 복잡한 작업을 처리할 수 있는 기능
- 여러 단계를 연결해 하나의 일련의 작업을 자동화하는 로직
- 모듈형 구성 요소를 재사용 가능한 파이프라인으로 조립하는 개념

Agent

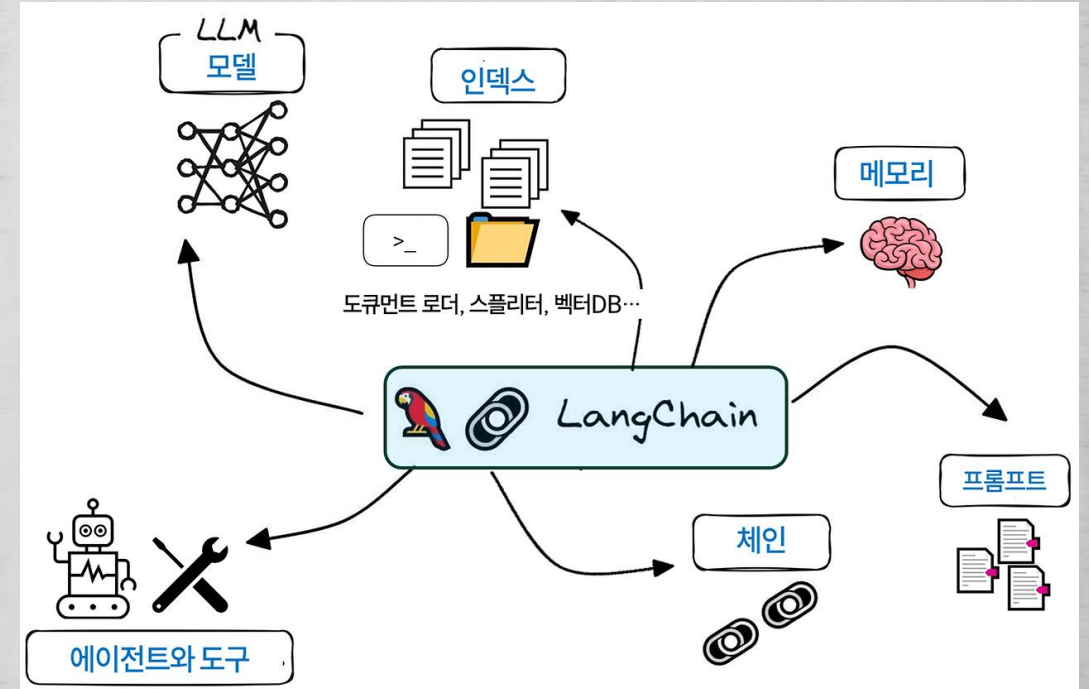
- 동적인 작업 흐름을 수행하는 기능
- 에이전트는 상황에 따라 알아서 결정하고 작업을 수행
- 정해진 순서 없이 사용자의 질문이나 상황에 맞게 필요한 도구나 체인을 사용하여 결과를 도출

Memory

- 대화 중 또는 작업 중에 일어난 정보나 데이터를 저장하고 활용하는 역할
- 메모리는 과거 대화를 기억하며 연속적인 대화나 작업을 가능하게 함
- 컨텍스트를 유지하며, 대화의 일관성을 높일 수 있음

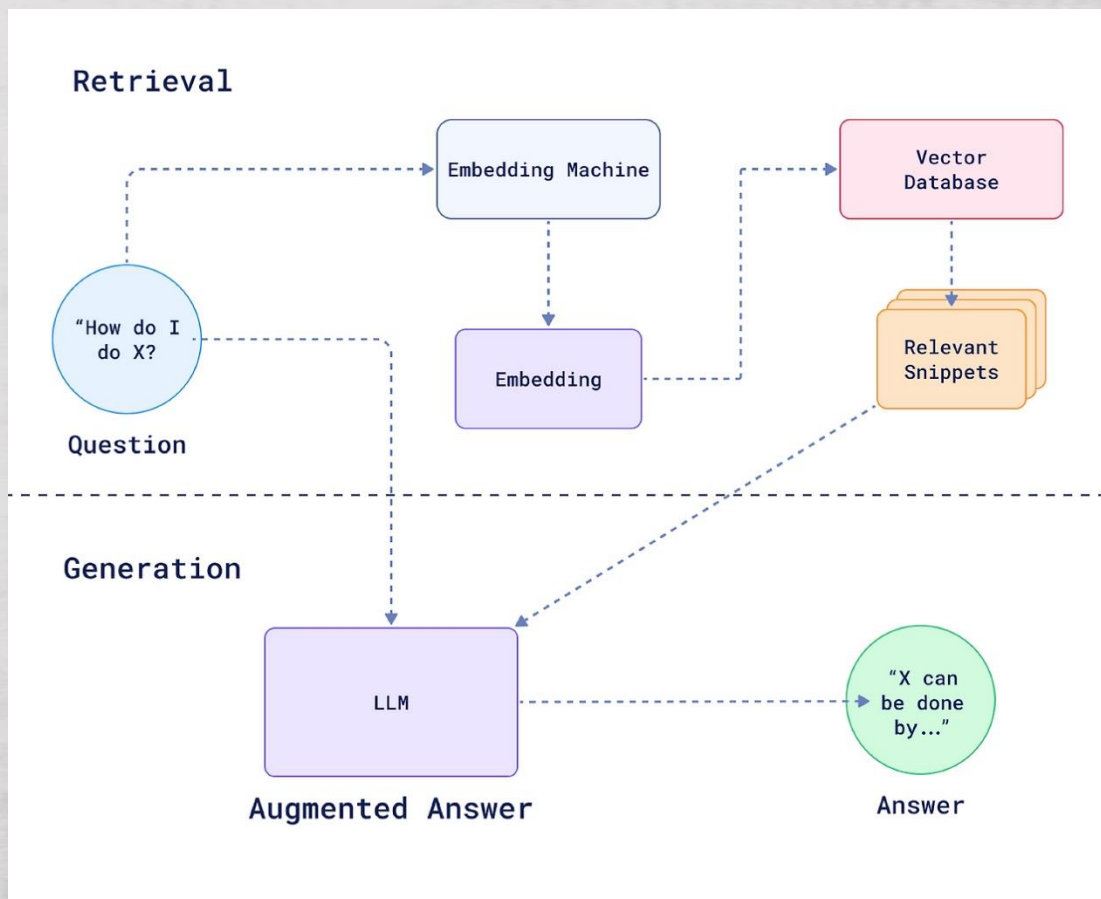
Tool

- 언어 모델이 외부 시스템과 상호작용하거나 특정 작업을 수행할 수 있도록 해주는 기능
- 언어 모델이 기본적으로 처리할 수 없는 계산, 검색, 파일 처리 등의 기능을 도구를 통해 확장할 수 있음



RAG (Retrieval-Augmented Generation)

풍부한 정보를 담고 있는 대규모 문서 데이터베이스에서 관련 정보를 검색하고, 이를 통해 언어 모델이 더 정확하고 상세한 답변을 생성할 수 있게 함



4. 랭체인 및 Streamlit

```
# Streamlit 페이지 설정 (제목 및 아이콘)
st.set_page_config(page_title="빅 3 상위뉴스 요약 📰", page_icon="📰")
st.title("빅 3 상위뉴스 요약 📰") # 페이지 타이틀 표시

# 세션 상태에서 대화 내용이 없을 경우 빈 리스트로 초기화
if "messages" not in st.session_state:
    st.session_state["messages"] = []

# 세션 상태에 저장된 대화 내역을 출력하는 함수
def print_history():
    for msg in st.session_state["messages"]:
        st.chat_message(msg.role).write(msg.content) # 사용자와 AI의 메시지를 출력

# 대화 내역을 세션 상태에 추가하는 함수
def add_history(role, content):
    st.session_state["messages"].append(ChatMessage(role=role, content=content)) # role에 따라 'user' 또는 'ai'로 구분

# 체인을 생성하는 함수 (프롬프트, 모델 연결)
def create_chain(prompt, model):
    chain = prompt | ChatOpenAI(model_name=model, api_key="api_key") | StrOutputParser() # OpenAI API와 파싱 체인 생성
    return chain
```

랭체인과 Streamlit을 사용해 언어 모델과 상호작용하는 웹 애플리케이션을 구축
Streamlit 페이지 기본 설정 및 필요한 함수 정의

4. 랭체인 및 Streamlit

```
# 사이드바에 대화 초기화 버튼과 프롬프트 설정 인터페이스를 추가
✓ with st.sidebar:
    clear_btn = st.button("대화내용 초기화")

    # 프롬프트 설정 UI
    prompt = """당신은 친절한 AI 어시스턴트 입니다. 사용자의 질문에 간결하게 답변해 주세요."""
    user_text_prompt = st.text_area("프롬프트", value=prompt)
    user_text_apply_btn = st.button("프롬프트 적용", key="apply1")

    # 프롬프트 적용 로직
    ✓ if user_text_apply_btn:
        st.markdown(f"✅ 프롬프트가 적용되었습니다")
        prompt_template = user_text_prompt + "\n\n#Question:\n{question}\n\n#Answer:"
        prompt = PromptTemplate.from_template(prompt_template)
        st.session_state["chain"] = create_chain(prompt, "gpt-4")
```

대화내용 초기화

프롬프트

당신은 친절한 AI 어시스턴트 입니다.
사용자의 질문에 간결하게 답변해 주
세요.

프롬프트 적용

사이드바에 대화 내용 초기화 버튼 생성 및 프롬프트 설정
수정된 프롬프트는 PromptTemplate를 통해 템플릿화되고, AI 모델과의 체인이 새로 생성

4. 랭체인 및 Streamlit

```
# 대화내용 초기화 버튼이 눌리면 세션 상태의 대화 내용 초기화
✓ if clear_btn:
    retriever = st.session_state["messages"].clear()

# 이전 대화 내역 출력
print_history()

# 세션 상태에 체인이 없다면 기본 체인을 생성
✓ if "chain" not in st.session_state:
    prompt_template = user_text_prompt + "\n\n#Question:\n{question}\n\n#Answer:" # 기본 프롬프트 템플릿
    prompt = PromptTemplate.from_template(prompt_template) # 템플릿 생성
    st.session_state["chain"] = create_chain(prompt, "gpt-4") # 체인 생성 후 세션에 저장
```

초기화 버튼이 눌리면 대화 내용 초기화
세션 상태에서 체인이 없다면 기본 체인을 생성하도록 함

4. 랭체인 및 Streamlit

```
# 사용자가 입력창에 질문을 입력했을 때
if user_input := st.chat_input():
    add_history("user", user_input) # 입력된 질문을 세션 상태에 저장
    st.chat_message("user").write(user_input) # 사용자 메시지 출력

    # AI의 답변 처리
    with st.chat_message("assistant"):
        chat_container = st.empty() # 응답을 실시간으로 표시할 공간 생성

        # AI 모델의 스트리밍 응답 처리
        stream_response = st.session_state["chain"].stream({"question": user_input}) # 질문에 대한 답변 스트리밍
        ai_answer = ""

        # 스트리밍된 AI 응답을 차례로 화면에 출력
        for chunk in stream_response:
            ai_answer += chunk # 스트리밍된 답변을 하나의 문자열로 결합
            chat_container.markdown(ai_answer) # 실시간으로 화면에 표시

        # AI 응답을 세션 상태에 저장
        add_history("ai", ai_answer)
```

사용자가 입력창에 질문을 입력했을 때 세션 상태에 저장하고 사용자 메시지를 출력하며 AI의 답변 처리 진행

결과

Flask 실행 결과

빅 3 상위 뉴스 요약

1. 동아일보

date : 2024-09-30

제목 : 연합뉴스 사장에 황대일씨 내정

요약 : 연합뉴스에 대한 경영 감독 등을 수행하는 뉴스통신진흥회는 26일 임시 이사회를 열고 황 선임기자를 사장 최종 후보자로 선정했다. 황 후보자는 다음 달 10일 주주총회에서 신임 사장으로 선임되면 직무를 시작한다. 임기는 3년. 황 후보자는 고려대 행정학과를 졸업하고 1990년 연합뉴스에 입사해 자카르타 특파원, 법조팀장, 증권부장, 경제부장, 마케팅부장, 전국·사회 에디터, 콘텐츠총괄본부장 등을 지냈다.

제목 : 연합뉴스 사장후보 3명으로 압축

요약 : 연합뉴스 사장후보추천위원회(사추위)는 25일 연합뉴스의 사장 후보로 추승호 연합뉴스TV 상무 겸 보도본부장, 황대일 연합뉴스 선임기자, 최기억 연합인포맥스 대표이사(접수번호 순) 등 후보 3인을 추천했다. 사추위는 이날 후보들에 대한 면접 심사를 실시한 뒤 후보자를 이들 3인으로 압축해 뉴스통신진흥회 이사회에 추천했다. 최종 후보 1인은 26일 오후 발표될 예정이며, 다음 달 열리는 연합뉴스 주주총회를 거쳐 최종 선임된다.

제목 : [신문과 놀자!/피플 in 뉴스]친근한 표현으로 날씨 알려준 '1호 기상캐스터'

요약 : 경북 김천시에서 태어나 대구공고를 졸업한 김 전 통보관은 서울대 사범대 원서를 내려 가는 길에 우연히 국립중앙관상대 기상기술원 양성생 모집 공고를 보면서 삶의 방향이 바뀌었습니다. 15일 노환으로 세상을 떠난 김 전 통보관의 열정과 헌신을 돌아보며 기상예보의 중요성과 그 발전 과정을 다시 되새겨 봅니다.

Flask 실행 결과

2. 중앙일보

제목 : [김원배의 뉴스터치]헤즈볼라

요약 : 헤즈볼라는 1982년 이스라엘이 레바논에 있는 팔레스타인 해방기구(PLO)를 축출한다는 명분으로 레바논을 침공하자 시아파 무슬림을 중심으로 결성됐다. 헤즈볼라는 정치조직으로도 성장했고, 레바논 남부를 실질 지배하면서 레바논 연립 여당의 일원으로 참여하고 있다. 헤즈볼라는 최근 이스라엘의 뺨뺨 폭탄 공격으로 지휘부가 대거 사망하는 등 큰 타격을 입었다. 이스라엘의 레바논 침공(지상전) 가능성도 커지고 있다. 1993년 이스라엘과 PLO는 평화 공존에 합의하는 오슬로 협정을 맺었고 팔레스타인 자치정부가 출범했다.

제목 : [사진] 무등산 증심사 화재 ... 국보 피해는 없어

요약 : 29일 광주광역시 무등산 증심사 식당 건물에서 불길과 연기가 치솟고 있다. 이날 화재로 공양간이 전소됐고 행원당 일부 시설이 훼손됐다. 국가 보물인 철조비로자나불좌상 등 주요 문화재는 불이 난 곳과 거리가 있어 피해를 입지 않았다. [뉴스1]

제목 : [사진] 국내외 치과기자재 한자리에

요약 : 29일 서울 강남구 코엑스에서 열린 2024 경기국제종합학술대회 및 치과기자재 전시(GAMEX 2024)에서 관람객이 치과기자재를 살펴보고 있다. 최신 치의학 정보를 공유하는 학술 프로그램과 국내외 치과기자재 트렌드를 소개하는 전시회가 마련됐다. [연합뉴스]

Flask 실행 결과

3. 한겨레

제목 : ‘과거사’ 전향적인 이시바...당내 입지 약해 한계 보일 듯 [뉴스분석]

요약 : 기자 신형철, 홍석재 이시바 시게루 일본 자민당 새 총재. 일본 자민당의 새 총재로 선출된 이시바 시게루(67) 전 간사장이 다음 달 1일 임시국회에서 제102대 일본 총리직에 오른다. 전문가들은 이시바 총재 당선이 한국에는 나쁠 게 없는 결과라고 입을 모았다. 양기호 성공회대 교수는 “자민당 내 파벌의 역학 구도상 이시바 총재의 운신의 폭이 그렇게 넓지 않을 것”이라며 “강제동원이나 ‘위안부’ 문제에 일본 정부가 보여온 강경한 태도는 쉽게 바뀌지 않을 것”이라고 내다봤다. 이시바 총재의 구상 중에는 중국은 물론 한국 내에서도 우려의 시선이 적지 않은 아시아판 나토(NATO·북대서양조약기구) 창설도 있다. 신형철 기자, 도쿄/홍석재 특파원
newiron@hani.co.kr

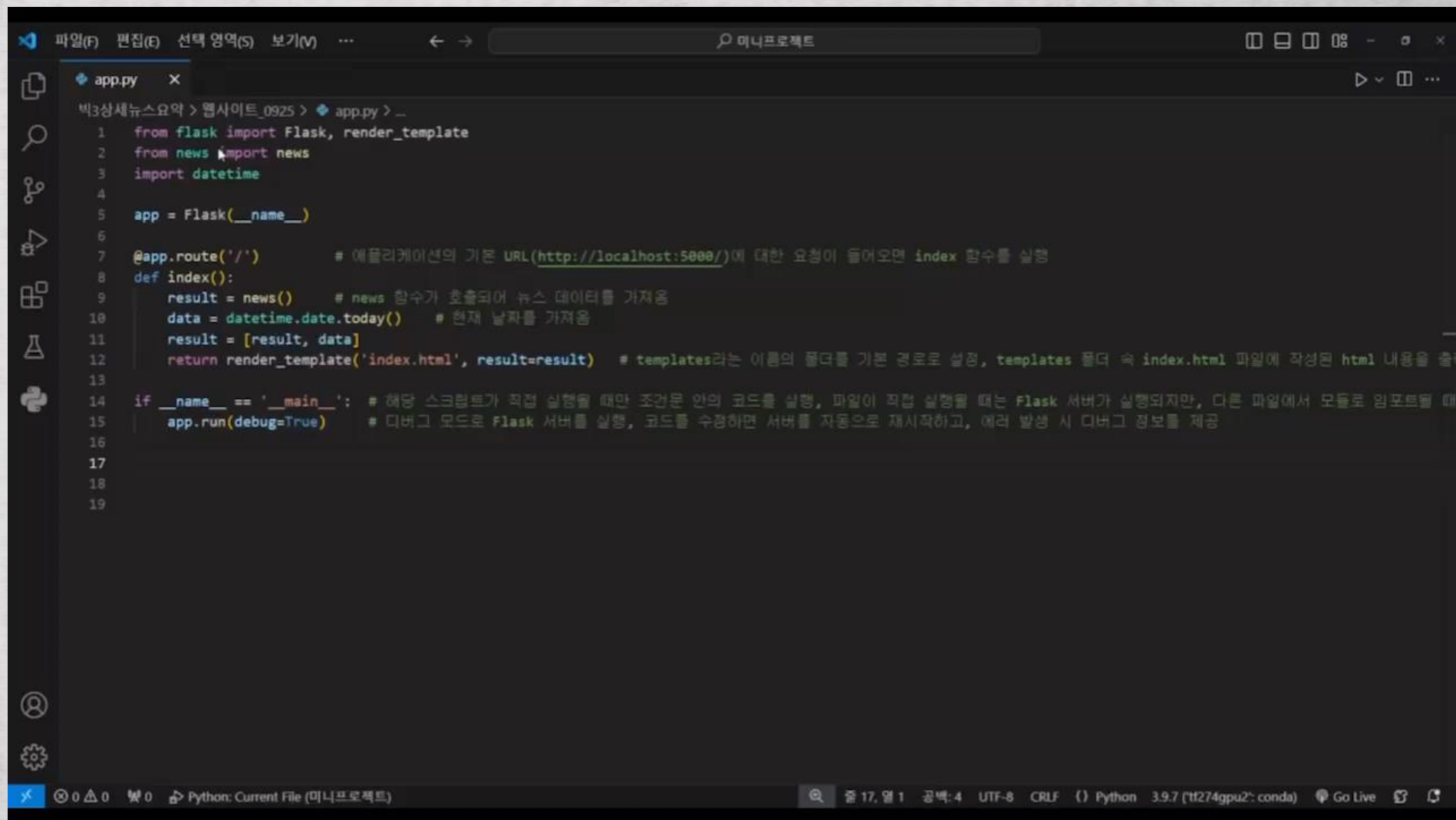
제목 : 네팔 폭우·산사태로 100명 넘게 숨져...1970년 이후 최대 강수량

요약 : 기자 박병수 네팔의 수도 카트만두를 관통하는 바그마티 강이 28일(현지시각) 큰 비로 범람하고 있다. 29일(현지시각) 네팔 언론 ‘카트만두 뉴스’ 등에 따르면, 네팔에 27일부터 폭우가 내리고 있다. 수도 카트만두 일부에는 하루 최대 322.2mm의 비가 내려 집이 물에 잠기고 도로와 다리가 끊겼다. 박병수 선임기자 suh@hani.co.kr

제목 : ‘윤 명예훼손’ 혐의 언론인들 “본인이 어떤 피해 입었는지 말해야”

요약 : 윤석열 대통령이 24일 서울 용산 대통령실 청사에서 열린 제41회 국무회의에서 머리발언을 하고 있다. 윤석열 대통령의 명예를 훼손했다는 혐의로 기소된 김용진 뉴스타파 대표와 한상진 기자, 신학림 전 전문위원이 첫 재판에서 윤 대통령을 증인으로 신청했다. 서울중앙지법 형사합의21부(재판장 허경무) 심리로 24일 열린 윤 대통령 명예훼손(정보통신망법 위반) 사건 첫 공판에서 신 전 위원 측은 “명예훼손 사건은 반의사불벌죄(피해자가 처벌을 원치 않으면 형사처벌 하지 않는 범죄)다. 김용진 대표도 재판 직전 기자들과 만나 “검찰은 이 사건 피해자를 윤석열 대통령이라고 적시했는데 우리는 윤석열 대통령이 어떤 피해를 입었는지 모른다. 우리 보도로 어떤 피해를 입었는지 본인 입으로 이야기를 해야 이 재판이 성립되는 것”이라고 말했다. 그거를 잘 아는 사람은 2011년 대검 중수부에서 이 사건 주임 검사 중수2과장 윤석열 검사다. 앞선 공판준비기일에서도 재판부는 윤 대통령 명예훼손 혐의와 상관 없는 간접정황이 공소장에 포함돼 있다며 삭제를 요청했는데 검찰은 본 재판에 돌입해서도 거듭 ‘이재명 공산당 프레임’을 소개하려 한 것이다.

Flask 실행 결과 (영상)



```
1 from flask import Flask, render_template
2 from news import news
3 import datetime
4
5 app = Flask(__name__)
6
7 @app.route('/') # 애플리케이션의 기본 URL(http://localhost:5000/)에 대한 요청이 들어오면 index 함수를 실행
8 def index():
9     result = news() # news 함수가 호출되어 뉴스 데이터를 가져옴
10    data = datetime.date.today() # 현재 날짜를 가져옴
11    result = [result, data]
12    return render_template('index.html', result=result) # templates라는 이름의 폴더를 기본 경로로 설정, templates 폴더 속 index.html 파일에 작성된 html 내용을 출력
13
14 if __name__ == '__main__': # 해당 스크립트가 직접 실행될 때만 조건문 안의 코드를 실행, 파일이 직접 실행될 때는 Flask 서버가 실행되지만, 다른 파일에서 모듈로 임포트될 때
15     app.run(debug=True) # 디버그 모드로 Flask 서버를 실행, 코드를 수정하면 서버를 자동으로 재시작하고, 예러 발생 시 디버그 정보를 제공
16
17
18
19
```


랭체인 프롬프트 미사용 실행 결과

<

Settings

대화내용 초기화

프롬프트

당신은 친절한 AI 어시스턴트입니다.
사용자의 질문에 간결하게 답변해 주
세요.

프롬프트 적용

Deploy

📄

빅 3 상위뉴스 요약

동아일보, 중앙일보, 한겨레 각 언론사별 상위뉴스 3개의 제목과 요약을 출력해줘

💬

죄송합니다, 저는 뉴스 기사를 실시간으로 확인하거나 검색하는 기능은 지원하지 않습니
다. 웹사이트나 뉴스 앱을 통해 해당 언론사의 상위 뉴스를 확인해보시는 것을 추천드립니
다. 더 궁금한 사항이 있으시면 다시 질문해주세요. 감사합니다.

Type your question here...

>

Windows 정품 인증

[설정]으로 이동하여 Windows를 정품 인증합니다.

랭체인 프롬프트 사용 실행 결과

Settings

대화내용 초기화

프롬프트

당신은 언론사 상위뉴스를 요약해주는 AI 언어시스템입니다. 사용자의 질문에 자세히 답변해 주세요.

프롬프트 적용

Deploy

빅 3 상위뉴스 요약

동아일보, 중앙일보, 한겨레 각 언론사별로 상위뉴스 3개의 제목과 요약을 출력해줘

🏠

동아일보: 1. "한미 국방장관, 北 대응 논의...한미동맹 강화" - 한미 국방장관이 북한 대응에 대한 논의를 진행하며 한미동맹을 강화하고자 함.
2. "코로나19 신규 확진 806명...전날보다 169명 줄어들어"
◦ 코로나19 신규 확진자 수가 전날보다 169명 줄어든 806명으로 발표되었다.
3. "이재명, '대선 출마' 발표...국민과 함께할 것"
◦ 이재명 경기도지사가 대선 출마를 선언하고 국민과 함께할 것이라고 밝혔다.

중앙일보:

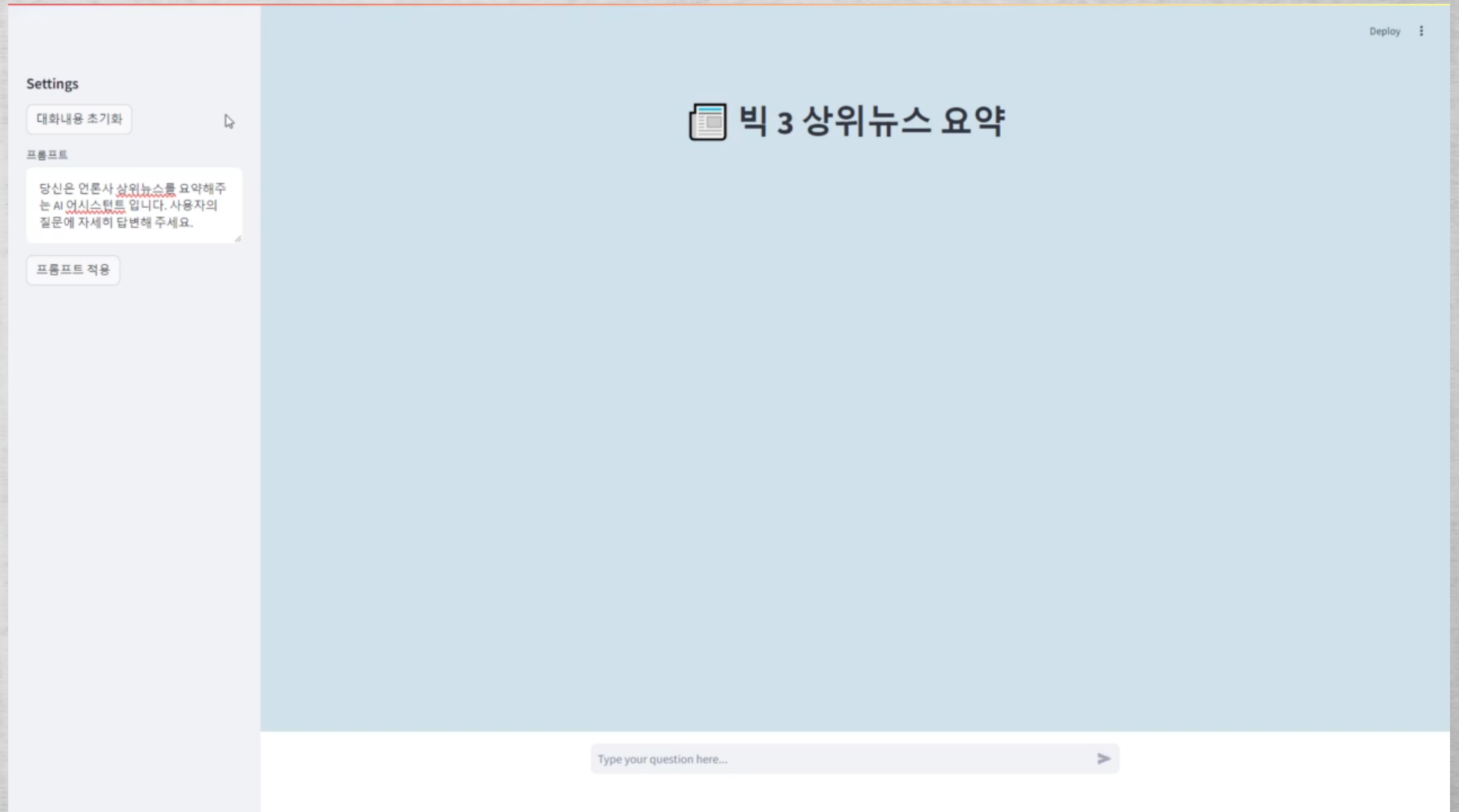
1. "올림픽 첫 노골... KBO, '올림픽 불참시 선수들에 미국프로 진출 제한'"
◦ KBO가 올림픽 불참 시 선수들의 미국프로 진출을 제한할 것이라는 발표로 논란이 일고 있다.
2. "청소년 인터넷중독, 8명 중 1명 '심각'"
◦ 청소년 중 8명 중 1명이 인터넷중독이 심각하다는 조사결과가 발표되었다.
3. "日, '혼혈' 무용가 국제공연 취소... 한국 인권단체 '반대'"
◦ 일본이 '혼혈' 무용가의 국제공연을 취소하자 한국의 인권단체들이 반대 의견을 제시하고 있다.

Type your question here... >

Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증합니다.

25

랭체인 실행 결과 (영상)



TIME TABLE

TIME TABLE

구분	9월																				10월			
	12일	13일	14일	15일	16일	17일	18일	19일	20일	21일	22일	23일	24일	25일	26일	27일	28일	29일	30일	1일	2일	3일	4일	
□ 웹크롤링 스타디																								
□ 언론사별 뉴스페이지 분석																								
□ 초안 코드 작성																								
□ 결과 확인 및 개선																								
□ 1차 발표																								
□ 성능 개선																								
□ 중간 발표																								
□ 웹페이지 제작																								
□ 랭체인																								
□ 최종 발표																								

THANK YOU

Q&A
