

발표 자료 참고 문서

1. 프로젝트 개요

프로젝트

AI 기반 가짜뉴스 탐지 시스템

프로젝트 목적

- 가짜뉴스의 급속한 확산에 대응하는 AI 솔루션 개발
- 사용자가 쉽게 뉴스의 진위를 판별할 수 있는 웹 서비스 제공
- 네이버 뉴스 크롤링을 통한 자동화된 분석 시스템 구축

주요 기능

- 네이버 뉴스 **URL 자동 분석**: URL 입력 시 자동 크롤링 및 분석
- 직접 텍스트 분석**: 사용자가 직접 입력한 뉴스 내용 분석
- AI 기반 진위 판별**: 6가지 기준으로 체계적 분석
- 실시간 결과 제공**: 즉시 분석 결과 확인 가능

타겟 사용자

- 일반 시민 (뉴스 소비자)
- 언론인 및 편집자
- 교육 기관 (미디어 리터러시 교육)
- 연구자 및 팩트체커

2. 시스템 아키텍처

전체 시스템 구조

사용자 → 웹 인터페이스 → Flask 서버 → 크롤링 모듈 → AI 분석 API → 결과 반환

주요 컴포넌트

- 프론트엔드**: React, HTML5, CSS3, JavaScript, GSAP
- 백엔드**: Flask 웹 애플리케이션
- 크롤링 엔진**: BeautifulSoup 기반 네이버 뉴스 파서
- AI 분석**: text-generation-webui API 연동

데이터 흐름

- 1. 사용자 입력 (URL 또는 텍스트)
- 2. 입력 타입 판별 및 전처리
- 3. 크롤링 (URL인 경우) 또는 직접 처리
- 4. 텍스트 정제 및 검증
- 5. AI 모델 분석 요청
- 6. 결과 파싱 및 사용자 제공

3. 사용된 핵심 기술

백엔드 기술 스택

기술	버전	용도	주요 특징
Python	3.8+	메인 개발 언어	풍부한 라이브러리 생태계
Flask	2.x	웹 프레임워크	경량화, 빠른 개발, 확장성
Nginx	1.x	웹 서버/리버스 프록시	고성능, 로드 밸런싱, 정적 파일 서빙
requests	2.x	HTTP 클라이언트	크롤링 및 API 호출
BeautifulSoup4	4.x	HTML 파싱	강력한 웹 스크래핑
Jinja2	3.x	템플릿 엔진	Flask 내장, 동적 HTML 생성

프론트엔드 기술

기술	용도	특징
React	UI 라이브러리	컴포넌트 기반, 가상 DOM
GSAP	애니메이션 라이브러리	고성능 웹 애니메이션
HTML5	웹 페이지 구조	시맨틱 마크업
CSS3	스타일링	반응형 디자인, 애니메이션
JavaScript	사용자 인터랙션	햄버거 메뉴, 폼 검증

AI/ML 기술 스택

기술	설명	장점
text-generation-webui	모델 서빙 플랫폼	OpenAI API 호환

핵심 기술 맛보기

백엔드의 핵심 - Flask

python

*Flask*는 이렇게 간단하게 웹 서버를 만들 수 있어요

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/analyze', methods=['POST'])
```

```
def analyze_news():
```

```
    return "뉴스 분석 결과"
```

- **역할:** 사용자 요청을 받아서 처리하고 응답을 돌려주는 웹 서버
- **특징:** 파이썬으로 몇 줄만으로도 웹 서비스 구축 가능

크롤링의 핵심 - BeautifulSoup

python

*HTML*에서 원하는 정보만 쓱쓱 뽑아내는 도구

```
from bs4 import BeautifulSoup
```

```
soup = BeautifulSoup(html_content, 'html.parser')
```

```
title = soup.select_one('h2.media_end_head_headline').get_text()
```

- **역할:** 웹페이지의 HTML을 분석해서 제목, 본문 등 필요한 정보만 추출
- **특징:** 복잡한 HTML 구조에서도 CSS 선택자로 정확히 원하는 부분만 가져올 수 있음

프론트엔드의 핵심 - React

jsx

// 컴포넌트로 *UI*를 블록처럼 조립하는 방식

```
function NewsAnalyzer() {
```

```
    const [result, setResult] = useState(null);
```

```
    return (
```

```
        <div>
```

```
            <InputForm onSubmit={analyzeNews} />
```

```
            {result && <ResultCard data={result} />}
```

```
        </div>
```

```
    );
```

```
}
```

- **역할:** 사용자가 보는 웹 페이지의 화면을 블록(컴포넌트) 단위로 만들고 관리
- **특징:** 데이터가 바뀌면 화면이 자동으로 업데이트되어 부드러운 사용 경험 제공

웹 서버의 핵심 - Nginx

nginx

빠르고 안정적인 웹 서버 설정

```
server {  
    listen 80;  
    location / {  
        proxy_pass http://127.0.0.1:5005; # Flask 앱으로 요청 전달  
    }  
}
```

- **역할:** 사용자의 웹 요청을 받아서 Flask 애플리케이션으로 전달하는 문지기
- **특징:** 동시에 많은 사용자가 접속해도 빠르고 안정적으로 처리

애니메이션의 핵심 - GSAP

javascript

// 부드럽고 멋진 애니메이션을 쉽게 만드는 도구

```
gsap.from(".result-card", {  
    duration: 1,  
    y: 50,  
    opacity: 0,  
    ease: "power2.out"  
});
```

- **역할:** 웹 페이지에 부드럽고 전문적인 애니메이션 효과 추가
- **특징:** CSS로는 구현하기 어려운 복잡한 애니메이션도 간단한 코드로 가능

AI 연결의 핵심 - text-generation-webui

python

AI 모델과 대화하는 방식

```
response = requests.post('http://127.0.0.1:5000/v1/chat/completions',  
                          json={'messages': [{'role': 'user', 'content': '이 뉴스 분석해줘'}]}))
```

- **역할:** 복잡한 AI 모델을 간단한 API 호출로 사용할 수 있게 해주는 다리
- **특징:** OpenAI와 같은 방식으로 AI 모델을 호출할 수 있어 개발이 쉬움

4. AI 분석 방식

가짜뉴스 탐지 기준 (6가지)

1. **사실 오류 (High):** 과학적·객관적 사실과의 불일치
2. **출처 불명확 (High):** 명확한 출처 없는 제3자 정보

3. **통계 왜곡 (High)**: 데이터의 의도적 조작
4. **제목 낚시 (Medium)**: 본문과 불일치하는 자극적 제목
5. **공포 조장 (Low)**: 사회적 불안감을 유발하는 표현
6. **감정적 표현 (Low)**: 주관적이고 감정적인 언어 사용

판단 우선순위

- **중요도 순**: High → Medium → Low
- **최종 판단**: 가장 높은 중요도 기준을 우선 적용
- **근거 제시**: 구체적인 판단 이유와 분석 내용 제공

출력 형식

```
json

{
  "진위": "가짜 뉴스 | 진짜 뉴스 | 판단 불가",
  "근거": "구체적인 판단 근거",
  "분석": "상세한 분석 내용"
}
```

5. 시스템 동작 과정

상세 처리 흐름

1. **입력 접수**: 사용자가 URL 또는 텍스트 입력
2. **타입 판별**: http/https 패턴으로 URL 여부 확인
3. **크롤링 실행** (URL인 경우):
 - 네이버 뉴스 도메인 검증
 - HTML 요청 및 다운로드
 - BeautifulSoup으로 파싱
 - 제목, 본문, 날짜, 언론사 추출
4. **텍스트 전처리**:
 - 광고 문구 제거
 - URL 및 이메일 제거
 - 기자 정보 정리
 - 공백 정규화
5. **AI 분석**:
 - 내용 길이 검증 (50자 이상)

- 시스템 프롬프트 구성
- API 요청 전송 (최대 5분 대기)
- JSON 응답 파싱

6. 결과 제공: 웹 페이지에 분석 결과 표시

6. 프로젝트 진행 현황

완료된 주요 기능

- ☒ 기본 웹 애플리케이션 구조 완성
- ☒ 네이버 뉴스 크롤링 엔진 개발
- ☒ AI 분석 API 연동 완료
- ☒ text-generation-webui 연동 성공
- ☒ 기본 UI/UX 디자인 완성
- ☒ Nginx 웹 서버 도입
- ☒ 오류 처리 및 예외 상황 대응
- ☒ API 타임아웃 및 안정성 개선

진행 중인 작업

- ☐ HTML 애니메이션 추가 (GSAP 라이브러리)
- ☐ 코드 정리 및 최적화
- ☐ UI/UX 개선 작업

향후 개발 계획

- ☐ React 또는 Vue.js 프론트엔드 전환
 - ☐ 가짜뉴스 탐지 전용 파인튜닝 모델 개발
 - ☐ 다중 언론사 크롤링 지원 확대
 - ☐ 사용자 피드백 시스템 구축
 - ☐ 실시간 분석 성능 최적화
-

7. 사용자 인터페이스

메인 페이지 (index.html)

- 깔끔한 입력 폼: 큰 텍스트 영역으로 사용성 극대화
- 반응형 디자인: 모바일과 데스크톱 모두 지원
- 직관적인 UX: 플레이스홀더로 사용법 안내

결과 페이지 (result.html)

- 시각적 결과 표시: 진위 판별 결과를 명확하게 표시

- **상세 분석 제공:** 근거와 분석 내용을 구조화하여 표시
- **사용자 편의성:** 새로운 분석 버튼으로 연속 사용 지원

디자인 특징

- **어두운 테마:** 눈의 피로 감소
- **고대비 색상:** 가독성 최우선
- **애니메이션:** 부드러운 사용자 경험
- **접근성:** ARIA 속성으로 스크린 리더 지원