

# 채찍피티랑 논문 요약하고 아이디어 찾기

## 1. 논문 요약하기

 연구 배경

 MLLMU-Bench란?

 주요 기여

 주요 실험 결과

 결론 및 한계

## 2. 프로젝트 아이디어 뽑아내기

 **AI 보조 기능에 개인정보 망각 로직 적용**

 개요

 구현 예시

 확장 포인트

 **관리자용 Unlearning 도구**

 개요

 구성 예시

 핵심 포인트

 **프로젝트 제목 제안**

 한글

 영어

 **프로젝트 기획 요약 (포트폴리오용 소개)**

 **기술 스택 제안**

 **주요 기능**

1. 요약 요청 기능

2. 개인정보 삭제 요청

3. 삭제 전/후 결과 비교 뷰

4. 관리자/유저용 로그 보기 (선택 기능)

 **기본 데이터 흐름**

 예시 API

 포트폴리오에 쓸 수 있는 어필 포인트

 확장 아이디어 (옵션)

## 3. 구체적인 주제 골라보기

 민감한 서비스 + GPT + 개인정보 보호 컨셉

 주제 예시 1. 심리 상담 일지 요약 서비스

 주제 예시 2. 의료 기록 기반 요약 서비스

 주제 예시 3. 익명 고백 게시판 요약 시스템

 이런 방향이 좋은 이유

 포트폴리오용 소개 문장 예시

# 1. 논문 요약하기

## 연구 배경

- 대규모 언어 모델(LLM) 및 멀티모달 대규모 언어 모델(MLLM)은 막대한 웹 데이터를 학습하며 개인정보를 기억하고 노출할 수 있어 **법적·윤리적 문제**를 야기함.
- LLM에 대한 '기계 망각(Machine Unlearning)' 연구는 활발하지만, MLLM에서는 상대적으로 **연구가 부족함**.

## MLLMU-Bench란?

- **MLLM 전용 기계 망각 벤치마크**로, 민감한 정보를 학습한 후 이를 선택적으로 "잊게" 만드는 알고리즘의 성능을 평가.
- 총 653개의 인물 프로필 (가상 인물 500명 + 유명인 153명) 에 대해 2만 개 이상의 질문을 제공함.
  - 각 인물에 대해 이미지+텍스트 및 텍스트 단독 질문 포함
- 벤치마크는 다음과 같이 4개 데이터셋으로 구성됨:
  - **Forget Set**: 모델이 "잊어야 할" 정보 포함
  - **Test Set**: Forget Set 변형 버전 (일부 문장/이미지를 바꿔서 일반화 성능 테스트)
  - **Retain Set**: 학습을 유지해야 할 정보 포함
  - **Real Celebrity Set**: 실제 유명인 정보로, 망각이 의도하지 않은 영향을 미쳤는지 평가

## 주요 기여

1. **MLLMU-Bench 개발**: 텍스트+이미지를 사용하는 MLLM에서 민감정보 제거 평가 가능
2. **모달리티별 성능 비교**: 텍스트만 사용하는 Unlearning이 생성 및 채우기(cloze) 작업에선 우수, 반면 이미지+텍스트 함께 사용하는 Unlearning은 분류 작업에 강함
3. **기존 알고리즘 평가**: GA(Gradient Ascent), Gradient Difference, KL Minimization, NPO, 프롬프트 기반 방법 등을 실험

## 주요 실험 결과

- \*Unimodal 방식(텍스트만)\*\*은 생성 및 cloze 작업에서 더 뛰어난 성능을 보임.
- \*Multimodal 방식(이미지+텍스트)\*\*은 분류 작업에서 더 뛰어난 성능을 보임.
- **Unlearning과 모델 유용성 간의 트레이드오프 존재:**
  - 성능 좋은 unlearning은 모델의 일반적인 능력(이웃 정보, reasoning 등)에 부정적 영향을 줄 수 있음.

## 결론 및 한계

- 기계 망각 알고리즘이 텍스트와 이미지를 동시에 고려할 필요 있음.
- 단일 모달만 잊게 해서는 **다른 모달에 정보가 남아 있을 수 있음.**
- 향후에는 **특정 속성만 선택적으로 망각**하는 기술, 그룹 이미지에서의 망각, 인컨텍스트 러닝을 통한 재학습 방지 연구가 필요함.

위 논문 주제를 기반으로, 프로젝트에 쓰일 아이디어들을 지피티와 함께 구상해보았습니다!

## 2. 프로젝트 아이디어 뽑아내기

### AI 보조 기능에 개인정보 망각 로직 적용

#### 개요

- 예: 웹에서 챗봇, 게시글 요약, 유저 추천, 검색 등의 AI 기능이 있는데, **특정 유저의 데이터가 포함된 응답을 모델이 "잊도록" 만드는 기능**을 붙이는 프로젝트
- 실제로 GPT API나 Hugging Face 모델 연동 가능

#### 구현 예시

- `/forget/{userId}` : Forget Set 등록
- `/generate-summary` : 게시글 요약 요청
- 프롬프트 전처리 시 Forget Set에 포함된 유저 정보 제거

- 고급 구현 시 자체 텍스트 분석 모델에 **Unlearning** 기법 적용 시도

### 💡 확장 포인트

- **Before/After 비교 UI**: "삭제 전에는 이런 응답이 나왔는데, 지금은 정보가 빠졌어요!"
- 사용자가 자신의 정보가 들어간 응답을 검출하면 "이건 지워주세요" 요청 기능
- 개인정보 탐지 모델 (ex. 이름/전화번호 regex + NER 등) 연동

## 🔧 관리자용 Unlearning 도구

### 🔑 개요

- 웹 관리자 페이지에서 특정 유저의 데이터를 완전히 **삭제/망각 요청**하고, 그 영향을 시각화하는 도구를 만드는 프로젝트
- AI 모델에서의 Forget/Retain 효과를 직접 확인 가능

### 🧱 구성 예시

- **Forget 버튼**: 유저 삭제 요청 → 백엔드에서 Forget Set 저장
- **Test Panel**: 질문 입력 시 AI 응답이 Forget/Retain에 따라 어떻게 달라지는지 보여줌
- **Unlearning 로그**: 언제 어떤 유저가 지워졌고, 어떤 데이터에 영향을 주었는지 로그
- **성능 변화 시각화**: ex. "Unlearning 이후 요약 정확도가 몇 % 낮아짐"

### 💡 핵심 포인트

- 관리자 입장에서 **프라이버시 보호와 AI 품질을 동시에 확인**할 수 있는 체험 플랫폼
- MLLMU-Bench처럼 "Forget/Test/Retain Set"을 작게나마 직접 구현해볼 수 있음

저는 1번이 더 마음에 드네요. 해당 주제로 구체화했습니다!



## 프로젝트 제목 제안



### 한글

"개인정보 보호를 위한 GPT 기반 요약 서비스: Right to be Forgotten 적용 웹 프로젝트"



### 영어

"Privacy-Aware GPT Summarizer: Web App with Right to Be Forgotten"



## 프로젝트 기획 요약 (포트폴리오용 소개)

이 프로젝트는 사용자가 작성한 게시글 또는 텍스트를 GPT 기반 모델로 요약해주는 웹 서비스입니다.

하지만 일반적인 AI 서비스와 달리, **특정 사용자의 정보가 포함되지 않도록 하는 '기계 망각(Unlearning)' 개념**을 도입했습니다.

사용자는 **개인정보 삭제 요청**을 통해 자신의 이름, 직업, 취미 등이 모델 응답에 반영되지 않도록 설정할 수 있으며,

시스템은 이를 반영하여 **요약 결과에서 해당 정보를 제거하거나 모호하게 처리**합니다.



GPT API 활용



민감 정보 감지 + 제거 로직



사용자의 '삭제 요청' 반영 기능



삭제 전/후 결과 비교



## 기술 스택 제안

구성	기술
프론트엔드	React + Tailwind + Axios

백엔드	Spring Boot (Kotlin or Java) + REST API
AI 연동	OpenAI GPT API (요약 기능)
개인정보 필터링	Regex + 기본 NER (예: 이름, 직업, 취미)
DB	MySQL (ForgetSet 저장)
기타	Swagger (API 문서), Postman (테스트)

## 주요 기능

### 1. 요약 요청 기능

- 사용자가 입력한 텍스트를 GPT에 전송
- Forget Set(삭제 요청 유저 데이터)에 있는 정보는 제거하거나 마스킹

### 2. 개인정보 삭제 요청

- 유저가 자신의 정보 삭제 요청 시 ForgetSet에 등록
- 예: 이름 = "홍길동", 직업 = "의사" → 모델 입력/출력 시 제거

### 3. 삭제 전/후 결과 비교 뷰

- 같은 텍스트에 대해:
  - 삭제 전 요약 → "홍길동은 서울의 병원에서 의사로 일하며..."
  - 삭제 후 요약 → "해당 인물은 병원에서 근무 중이며..."

### 4. 관리자/유저용 로그 보기 (선택 기능)

- 어떤 정보가 삭제되었고, 요약 결과에 어떤 변화가 있었는지 기록 표시

## 기본 데이터 흐름

사용자 입력 → 백엔드 (삭제 요청 포함 여부 확인) → GPT 요청 전 프롬프트 수정 → 응답 수신 → 민감 정보 후처리 → 프론트에 응답 표시

## 예시 API





```
POST /api/summarize
{
  "userId": 42,
  "text": "홍길동은 서울대 병원에서 근무하며 ... "
}
```

```
POST /api/forget
{
  "userId": 42,
  "keywords": ["홍길동", "서울대 병원", "의사"]
}
```

## 포트폴리오에 쓸 수 있는 어필 포인트

- “GPT API를 단순 호출하는 것이 아닌, 개인정보 보호라는 현실적 문제를 해결하는 필터링·프롬프트 전처리·응답 후처리 로직까지 설계했습니다.”
- “유저의 삭제 요청이 실제로 어떻게 응답 결과에 영향을 주는지를 Before/After로 시각화해 직관적인 체험을 제공했습니다.”
- “기계 망각(Machine Unlearning) 개념을 웹 서비스에서 구현하며, LLM 안전성과 활용성 사이의 균형을 고민했습니다.”

## 확장 아이디어 (옵션)

-  Hugging Face 모델 연동하여 커스텀 요약기 구축
-  JWT 인증 기반 사용자 시스템 도입
-  실제 사용자를 위한 데이터 예시 생성기 (가상 인물 자동 생성)
-  관리자 대시보드 (삭제 요청 통계, 로그 기록 등)

서비스에다가 이 주제를 붙이고 싶어서, 조금 더 다양하게 만들어봤습니다!

새로운 보안 주제를 적용하는 거라 서비스를 무겁게 가면 시간이 너무 오래 걸릴 거 같습니다. 무거운 주제 말고 간단히 할 수 있는 주제를 중심으로 골라봤습니다.

### 3. 구체적인 주제 골라보기

#### 민감한 서비스 + GPT + 개인정보 보호 컨셉

##### 주제 예시 1. 심리 상담 일지 요약 서비스

사용자가 자신의 심리 상태를 일기처럼 적으면, GPT가 간단히 요약해 주고 감정 상태를 정리해줌.

그런데 민감한 정보(이름, 장소, 구체적인 사건 등)를 지워서 응답에 포함되지 않도록 처리!

##### Unlearning 포인트:

- 사용자는 나중에 "내가 말한 A라는 사건은 지워주세요" 요청 가능
- 또는 민감 키워드(이름, 병명 등)를 자동 감지해서 제거 후 요약

##### 주제 예시 2. 의료 기록 기반 요약 서비스

사용자가 병원 진료 기록/증상 내용을 넣으면 GPT가 증상 요약, 의사에게 보여줄 간단한 정리 제공

민감 정보(이름, 주소, 특정 병원명 등)는 사전 제거

##### Unlearning 포인트:

- "내 진료 정보는 더 이상 저장하거나 분석하지 말아주세요" 요청 가능
- Unlearning 로그 남기기



### 💡 주제 예시 3. 익명 고백 게시판 요약 시스템

익명 글이 너무 길어질 경우, GPT가 요약해서 감정을 정리해줌  
but, 실명이나 학과, 장소 등은 응답에서 자동 제거

#### Unlearning 포인트:

- "내가 예전에 올린 글에 있는 [홍길동]은 지워주세요" → 즉시 적용됨
- 게시판에서 삭제되어도 AI는 기억하고 있을 수 있기 때문에 Unlearning 필요

### 🧭 이런 방향이 좋은 이유

- 너무 "AI 연구적"이지 않아서 **사용자 공감**이 쉽고
- 포트폴리오에서 "**보안 + 실용성 + 트렌디함**" 모두 설명 가능
- 실제 서비스로 발전 가능성 있음!

### 💬 포트폴리오용 소개 문장 예시

"심리 기록 요약 서비스를 만들면서, 사용자의 민감한 사건이나 이름이 요약 결과에 포함되지 않도록 보호하는 기능을 구현했습니다. 사용자는 나중에 자신이 입력한 정보 중 일부를 '삭제 요청'할 수 있으며, 시스템은 이에 따라 GPT 응답에 해당 정보를 포함하지 않도록 처리합니다. 단순한 AI 연동을 넘어, 개인정보 보호를 고려한 UX 설계를 실현한 점이 핵심입니다."

1, 2, 3 번 다 너무 재밌을 거 같습니다!!