

법률 조언 웹 서비스 'LawBot'

윤슬 팀 / NLP 08

강민재_T5005 | 김주원_T5056 | 김태민_T5067 | 신혁준_T5119 | 윤상원_T5131

목차



01/ Project Overview

프로젝트 기획의도, 주제 선정 배경, 기대효과

02/ Service Architecture

유저 플로우, System Architecture

03/ Data Pipeline

데이터 크롤링, EDA, 전처리

04/ Model Architecture

Model Architecture

05/ Project Procedure

협업 방식, 프로젝트 타임라인, 팀원 역할

06/ Result

시연 영상, 후속 개발 및 연구, 결과 및 고찰



전 세계 리걸 테크 업체 7000여개
투자 규모 113억 달러(14조 8900억)
최근 2년 투자금액 48억 달러

- 글로벌 시장조사기관 트랙스(Tracxn) 2022년 9월 기준 -

국내 31개
리걸테크 기업
-문화일보-

법률 용어는 국민이 이해하기 쉽다
(부정 반응 76.3%)

법률 문장은 국민이 이해하기 쉽다
(부정반응 78.4%)

- 한국 법제연구원 실태조사 -

30분 법률 상담시 5만원~ 10만원
많게는 30만원의 변호사 상담비용

01 Legal Tech 의 성장

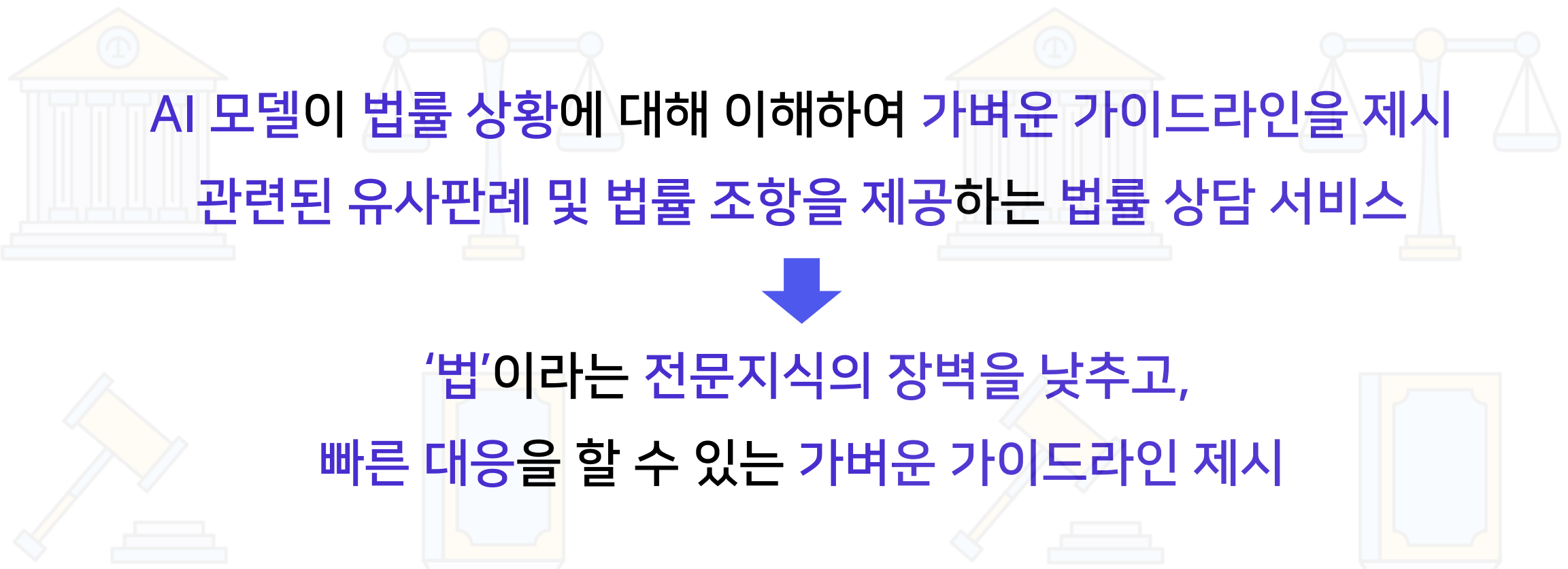
국제적인 성장, 국내의 소요 증가

02 전문 지식의 진입장벽

일반인들에게는 법률 용어와 법조문 문맥의 난해함
→ 법률 용어나 문장에 대한 낮은 이해도

03 어려운 접근성

상담에 많은 돈 또는 시간 소비



AI 모델이 법률 상황에 대해 이해하여 가벼운 가이드라인을 제시
관련된 유사판례 및 법률 조항을 제공하는 법률 상담 서비스



'법'이라는 전문지식의 장벽을 낮추고,
빠른 대응을 할 수 있는 가벼운 가이드라인 제시



개발사	리걸 테크 서비스	기능
로아팩토리	모두싸인	온라인 계약서 체결 서비스
헬프미 법률사무소	헬프미	변호사 찾기, 지급명령, 등기, 상속 온라인 법률 서비스
로앤컴퍼니	로톡	변호사+사례 검색 및 온라인 법률 상담 및 변호사 매칭
아미스쿠렉스	로폼	문서작성, 변호사 상담 및 예약
보리움 법률사무소	머니백	가압류 신청

차별점

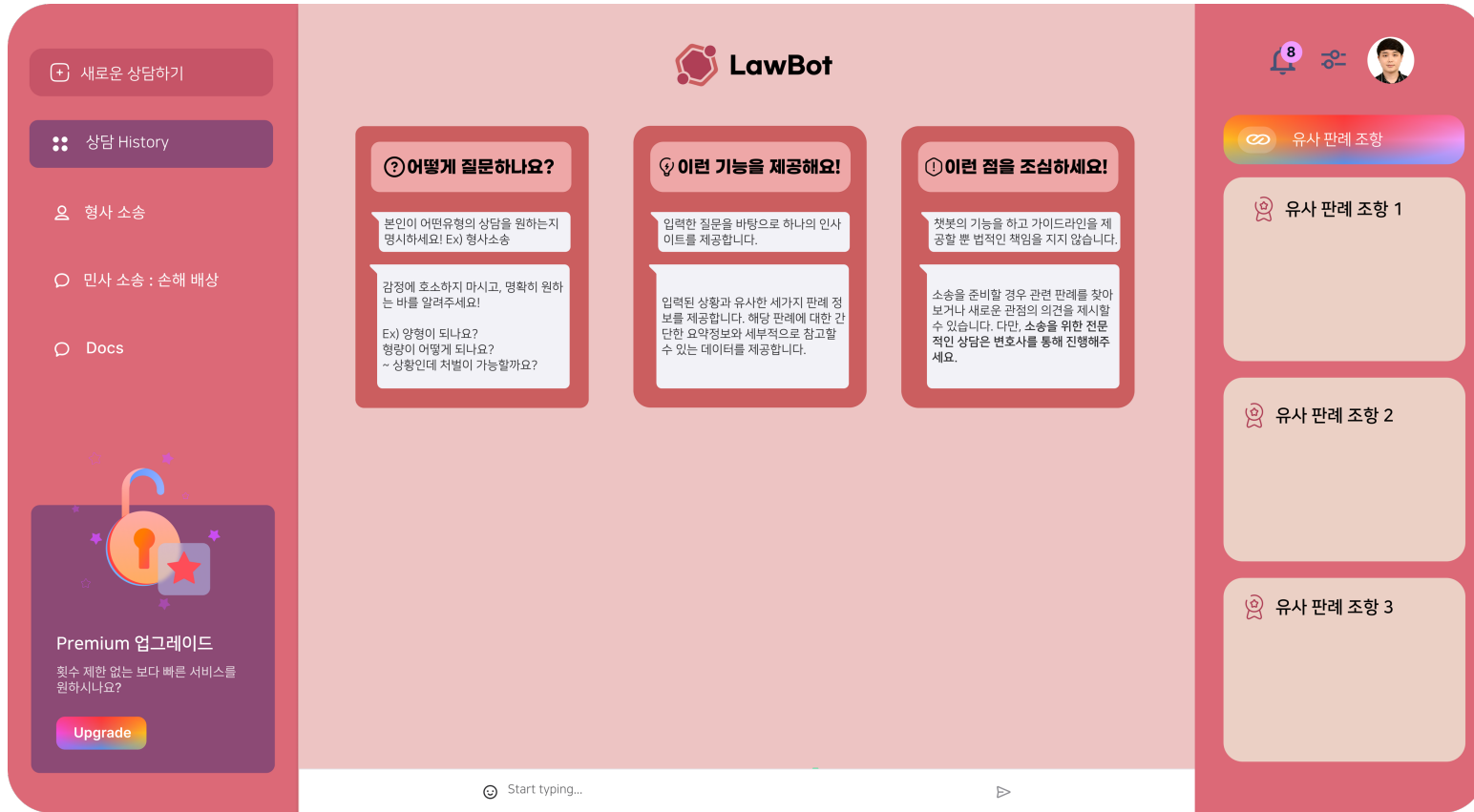
대부분의 리걸 테크 서비스

➔ 변호사 매칭, 법률 검색 플랫폼 (여전히 낮은 접근성)

법률 상담 LawBot

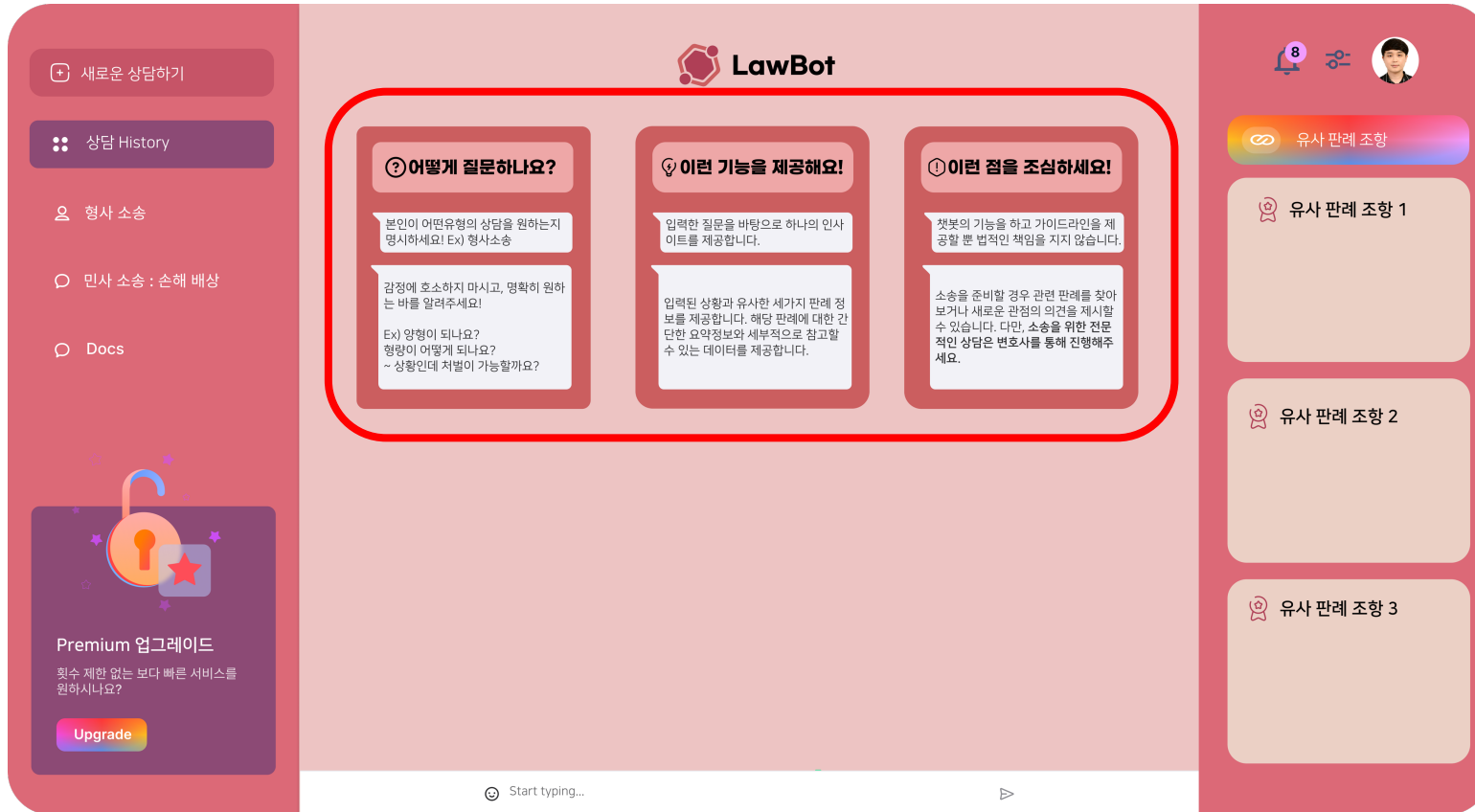
➔ AI 모델을 이용하여 법률적인 문제 상황에 대한

유사 판례 + 가벼운 가이드라인 빠른 시간내에 제공



메인 페이지(SPA)

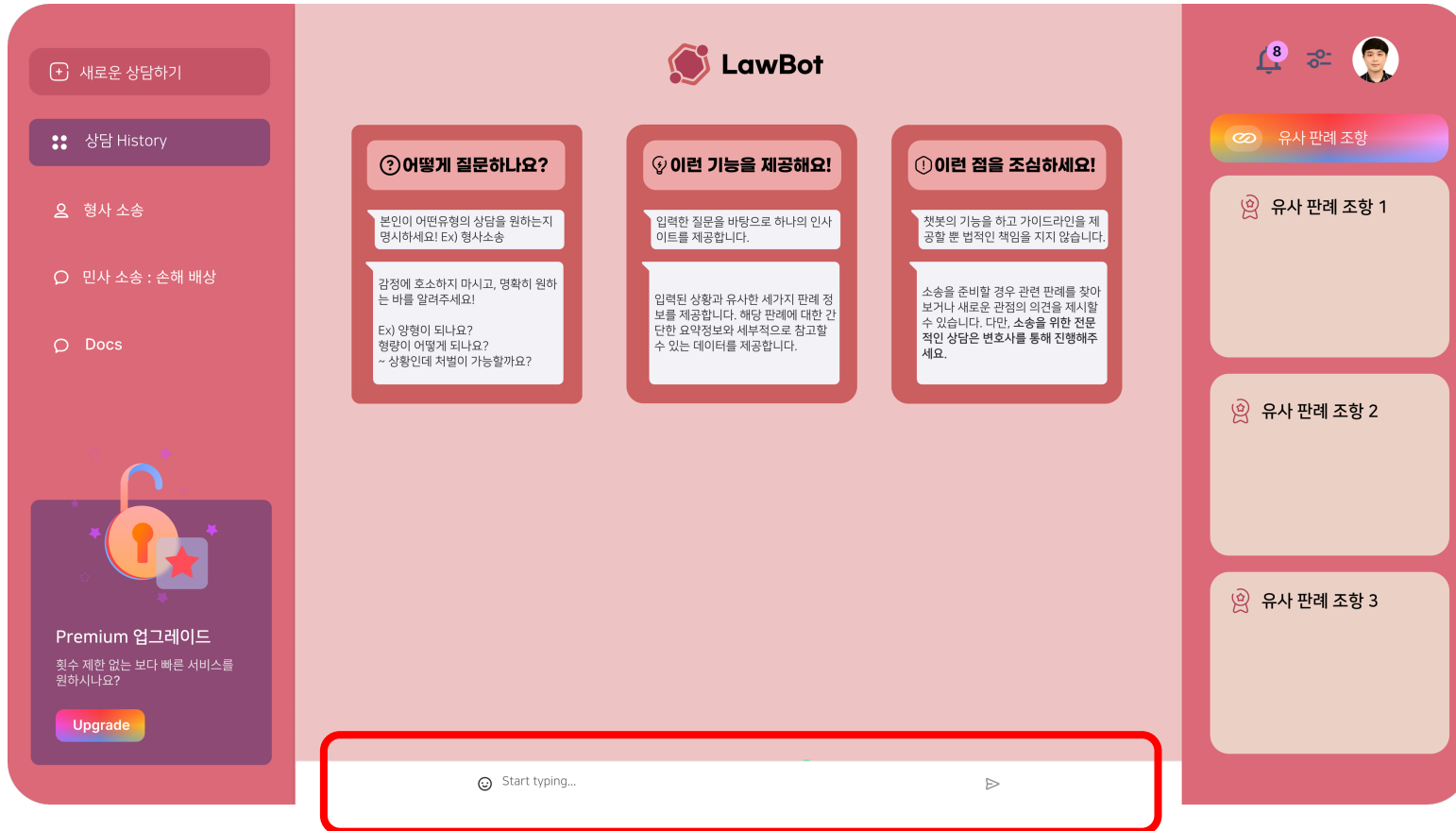
1. 웹 서버 접속
2. 배경화면을 통해 사용방법과 사용 예시 제공
3. 메시지 프롬프트 창에 자신이 처한 법률적인 상황 입력



메인 페이지(SPA)

1. 웹 서버 접속
2. 배경화면을 통해 사용방법과 사용 예시 제공
3. 메시지 프롬프트 창에 자신이 처한

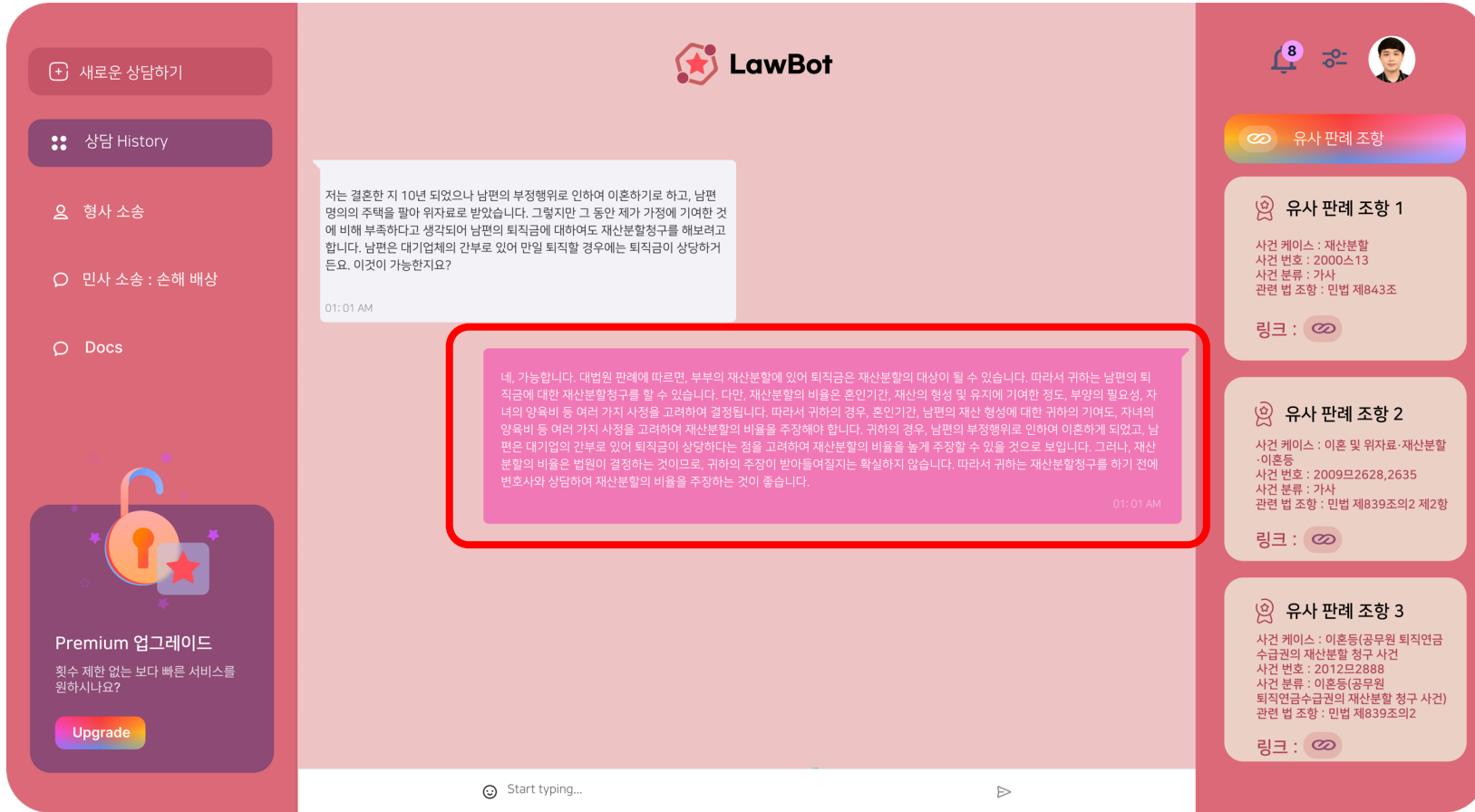
법률적인 상황 입력



메인 페이지(SPA)

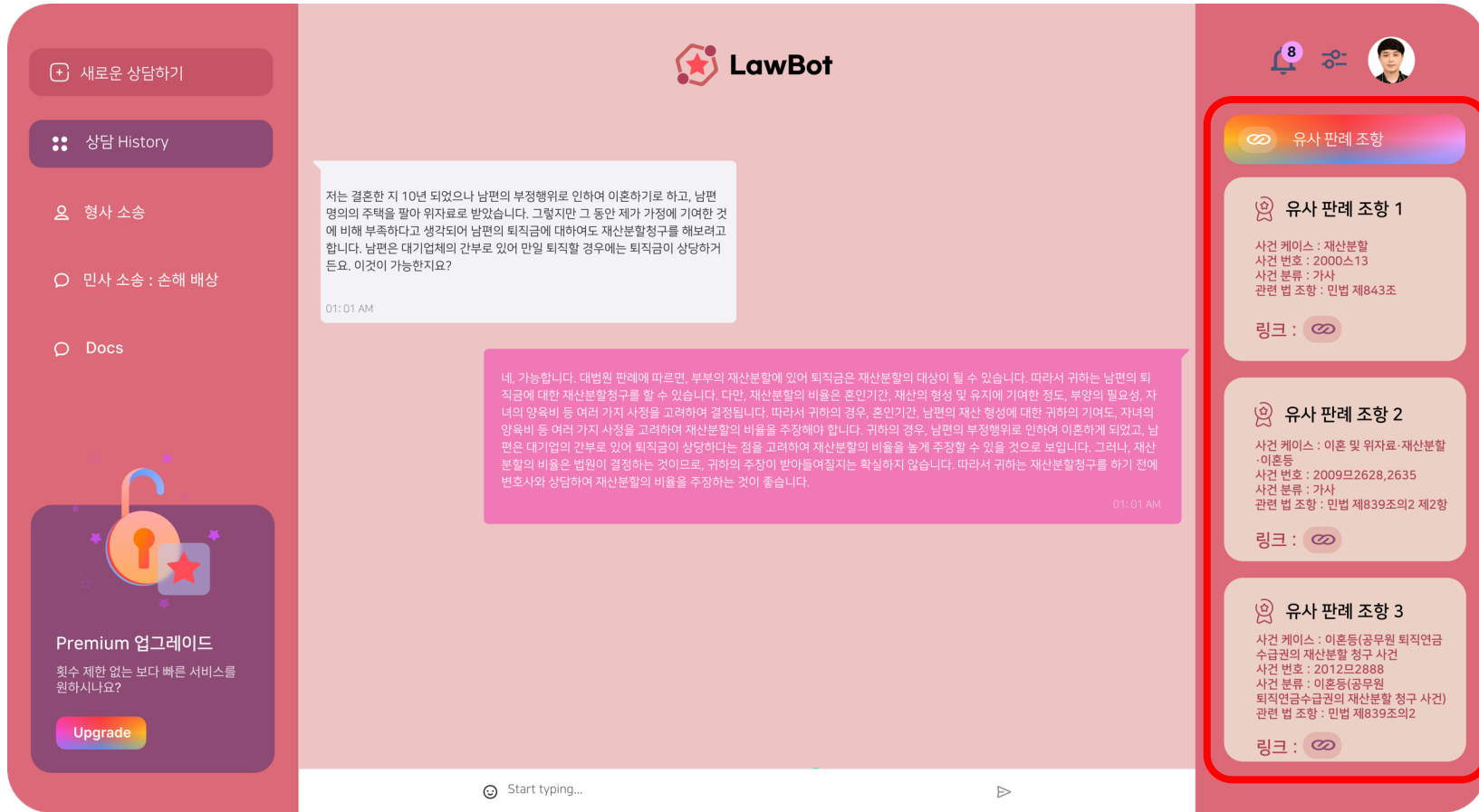
1. 웹 서버 접속
2. 배경화면을 통해 사용방법과 사용 예시 제공
3. 메시지 프롬프트 창에 자신이 처한

법률적인 상황 입력



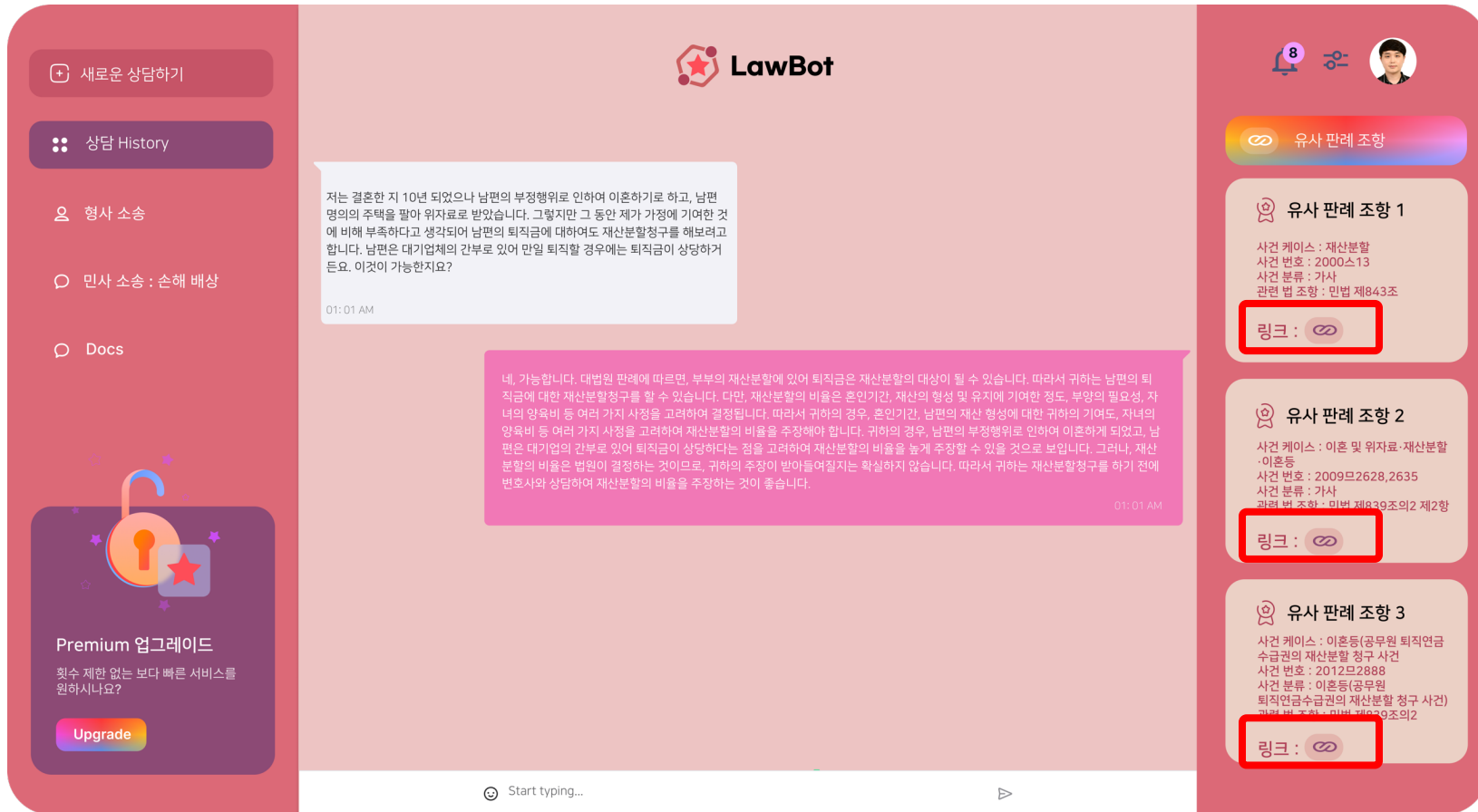
메인 페이지(SPA)

1. AI 모델이 상황 맥락을 이해하여 법적인 가이드라인 메시지 형태로 제공
2. 우측 사이드바에서 사용자가 입력한 상황과 비슷한 최대 3가지의 유사 판례 제공
3. 링크를 통해 법률 정보를 직접 확인 가능



메인 페이지(SPA)

1. AI 모델이 상황 맥락을 이해하여 법적인 가이드라인 메시지 형태로 제공
2. 우측 사이드바에서 사용자가 입력한 상황과 비슷한 최대 3가지의 유사 판례 제공
3. 링크를 통해 법률 정보를 직접 확인 가능



메인 페이지(SPA)

1. AI 모델이 상황 맥락을 이해하여 법적인 가이드라인 메시지 형태로 제공
2. 우측 사이드바에서 사용자가 입력한 상황과 비슷한 최대 3가지의 유사 판례 제공
3. 링크를 통해 법률 정보를 직접 확인 가능



Product Serving에 고려했던 부분들



확장성
Scalability

서비스 확장이 쉬운
마이크로 서비스.



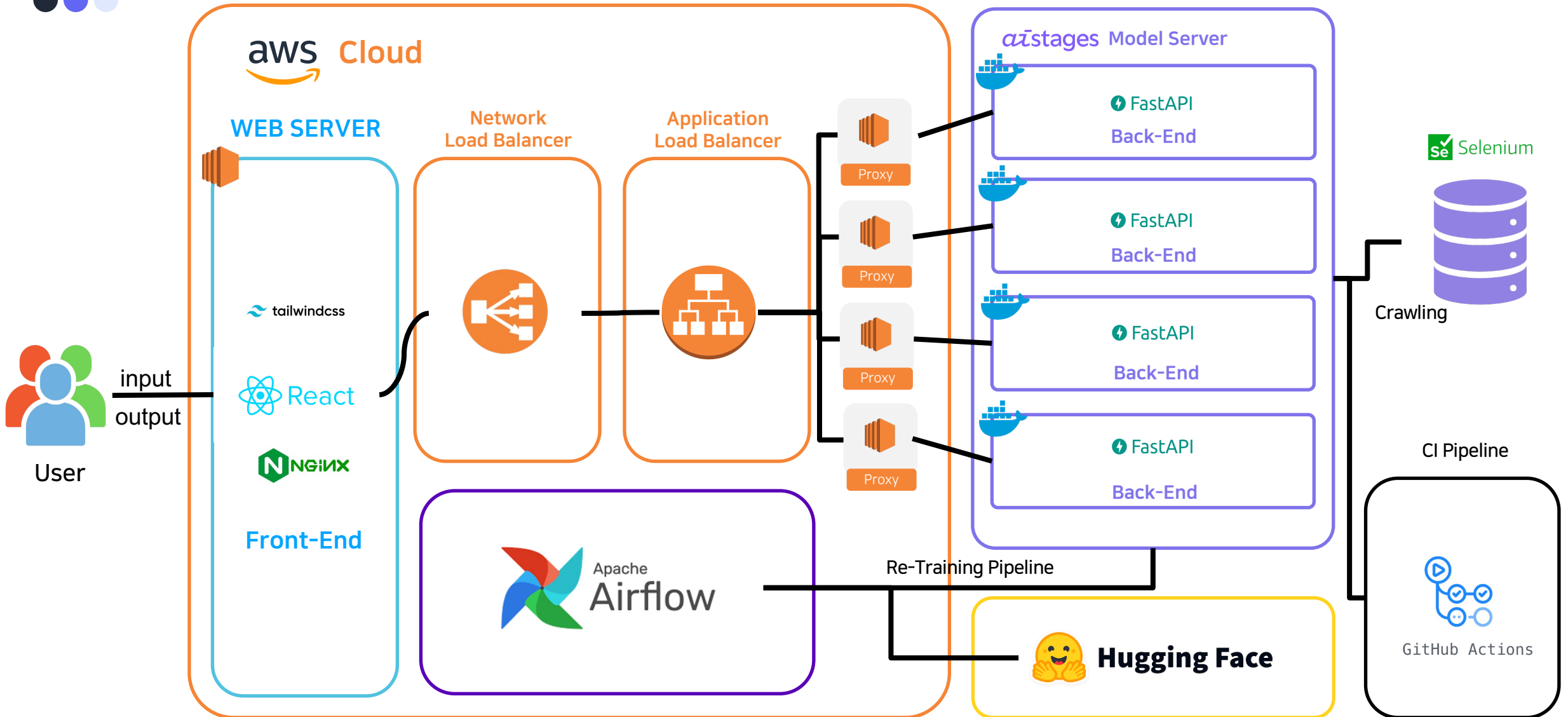
효율성
Efficiency

사용자에게 빠른 서비스가
제공될 수 있도록 효율적인 서버관리



자동화
Automation

데이터를 최신화하고, 온라인으로
재학습하는 MLOps 파이프라인 구축





AWS EC2 Proxy Web Server 구축

- React.js를 기반으로 한 FE 구축
- Proxy Server와 통신할 수 있는 API 구축
- Nginx를 이용한 웹 어플리케이션 배포

CI 파이프라인 구축

[사용 이유]

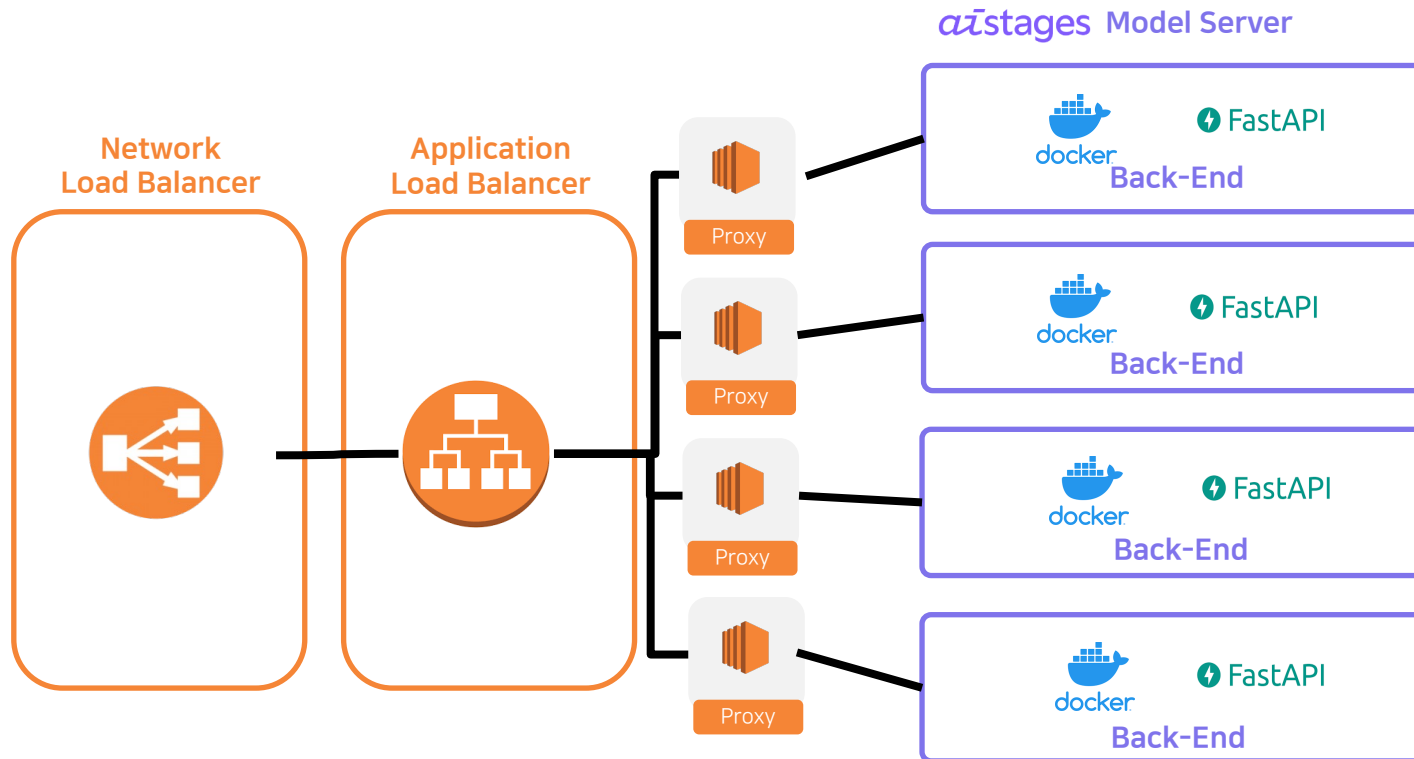
- 소스 코드 수정시 문제를 뒤늦게 발견할수록 시간적인 비용이 더 들기 때문에 테스트 자동화 필요

[구현 방법]

- GitHub Actions, Pytest 활용
- V100 서버를 이용한 Self-Hosted Runner

AI Stage V100 Backend Server 구축

- Fast API를 기반으로 한 BE 구축
- Proxy Server와 통신할 수 있는 API 구축
- 모델 서버와 router간 ssh 터널링 구축



Load Balancing

[사용 이유]

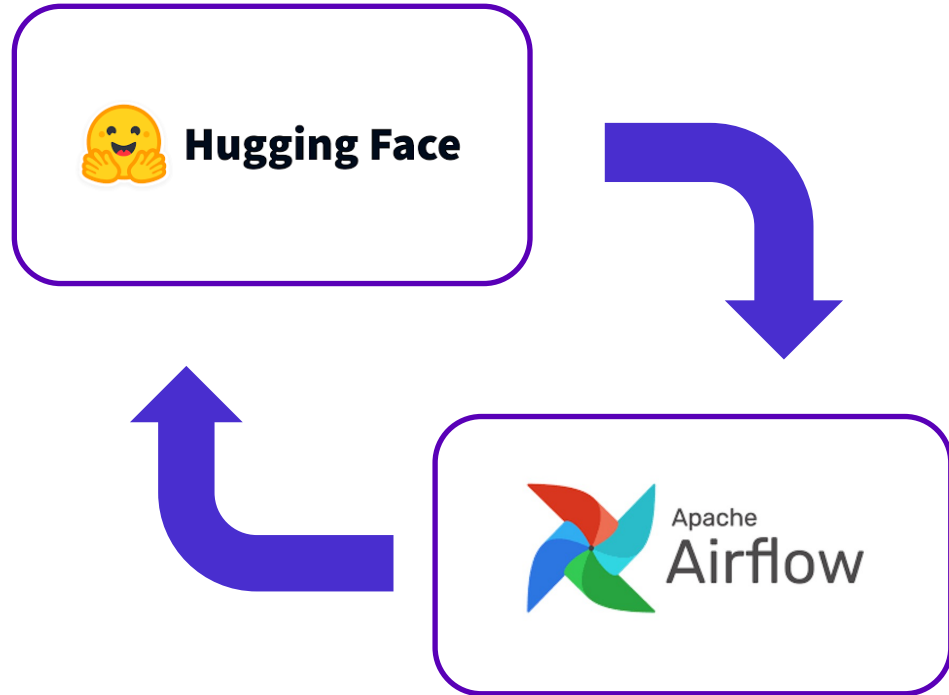
트래픽 분산을 위해

[구현 방안]

V100 서버 각각에 proxy 서버의 역할을 하는 EC2를 두고, Application Load Balancer를 5대의 proxy 서버와 연결하여 로드 밸런싱을 적용

[효과]

health check를 통해 모델 서버에 장애가 발생할 경우, 해당 서버로 트래픽을 보내지 않음으로써 서버의 장애가 서비스의 장애로 이어지는 것을 방지



모델 학습 파이프라인

[사용 이유]

매주 새롭게 추가되는 법률 사례들을 데이터셋에 추가하여 재학습

[구현 방안]

Airflow를 이용하여 HuggingFace에 일주일 마다 올라온 데이터를 재학습

Hugging Face 데이터 로드 → 전처리 → 모델 학습 → Hugging Face 모델 업로드



01 데이터 정의

서비스 목적에 맞는 데이터는 무엇인지
모델 입출력을 고려한 데이터의 형태는
어떠한지

02 데이터 크롤링

데이터 수집 과정에서 고려한 사항은
무엇인지

03 데이터 EDA

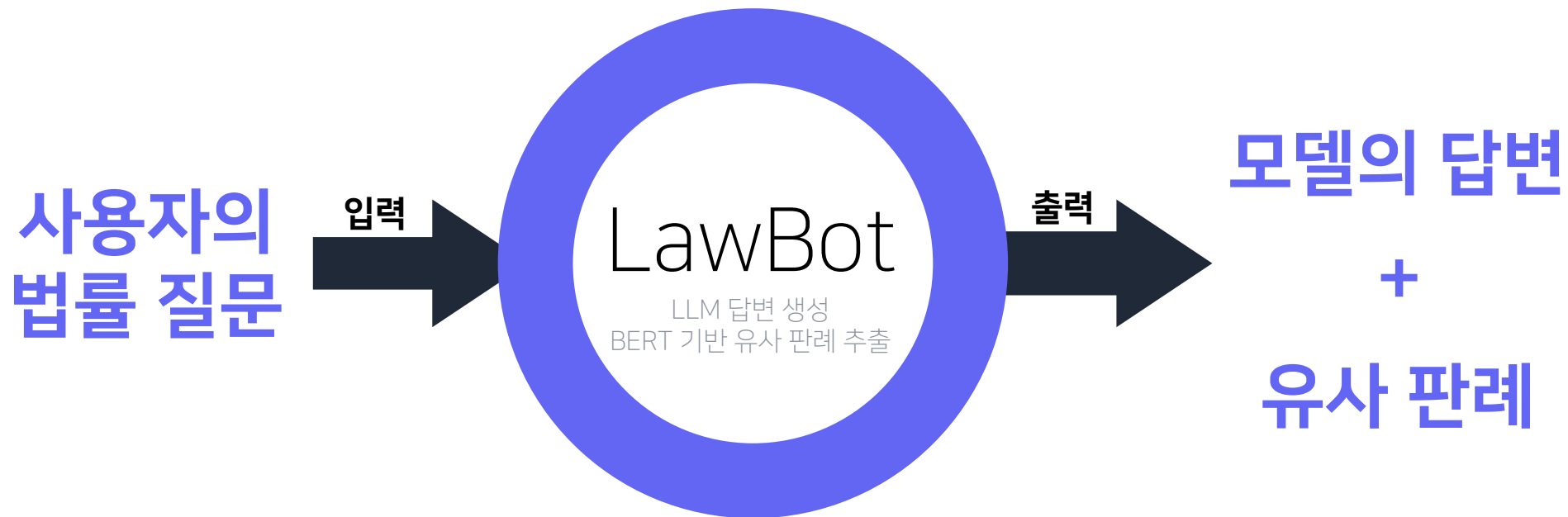
수집한 데이터의 특징은 어떠한지
데이터가 갖고 있는 문제점이 어떠한지

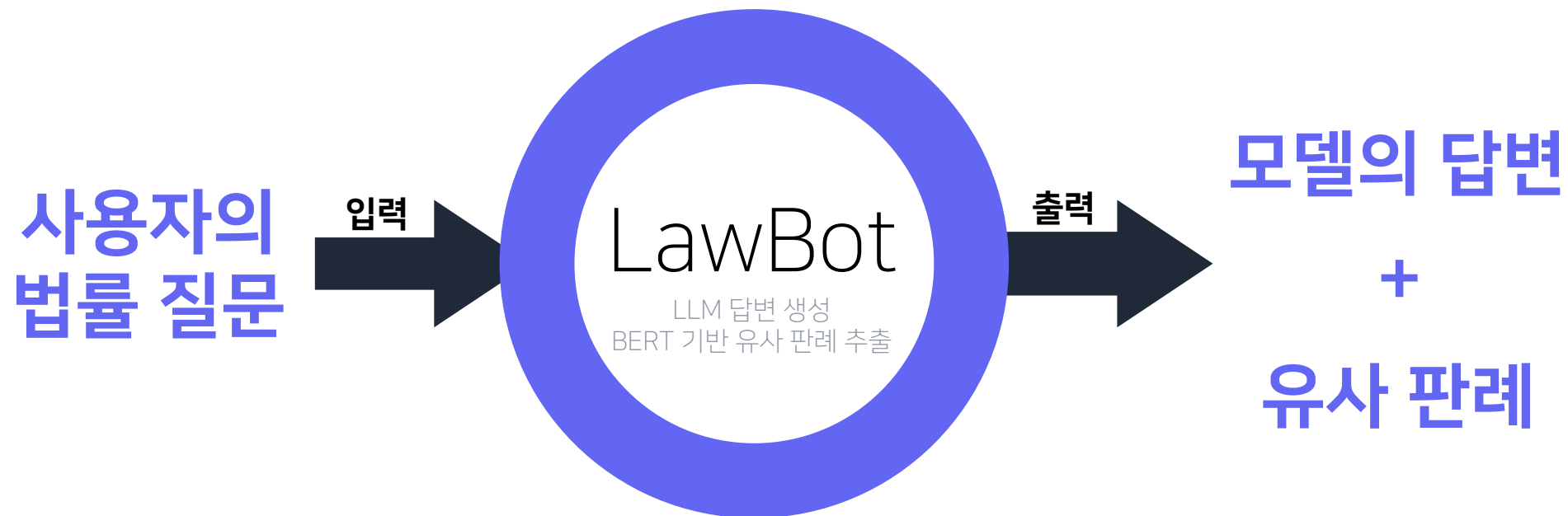
04 데이터 전처리

분석에 기반하여 데이터가 어떻게
변형되어야 하는지

05 데이터 생성

학습용 데이터의 개수가 충분한지





- 모델 입출력을 고려
1. 법률 QA 데이터
 2. 판례 데이터



데이터 탐색



판례 데이터

AI HUB: 법률/규정 (판결서, 약관 등) 텍스트 분석 데이터(77,382개)



법률 QA 데이터

HuggingFace : juicyjung/easylaw_kr (2,195개)
Github : haven-jeon/LegalQA (1830개)

상대적으로 부족한 QA 데이터를 크롤링하여 학습데이터를
구축하기로 결정



데이터 크롤링을 할 때 세웠던 원칙



1. 공개된 데이터만 수집하며, 보호된 영역에 무단으로 접근 **X**
2. 영리적인 목적으로 사용하지 않으며, 학습 및 연구 용도로만 사용.
3. 무단으로 재배포 **X**

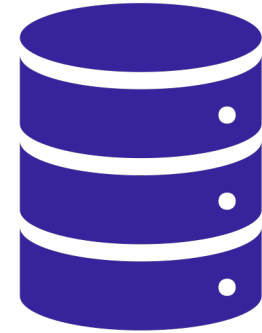


 Selenium

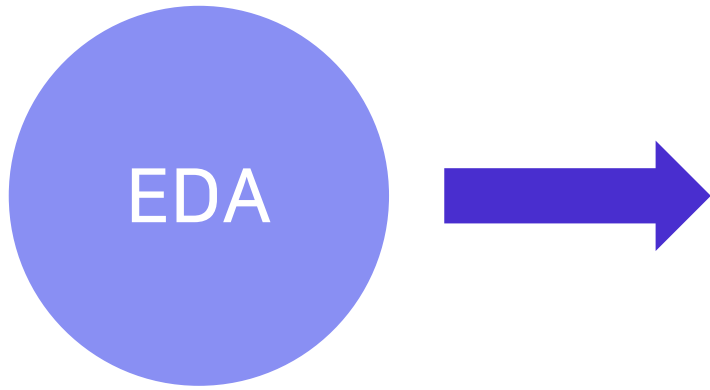
동적 크롤링



대한법률구조공단
사이버상담 데이터
2,463개



대한법률구조공단
법률상담사례 데이터
9,994개



학습시 의도치 않은 편향이 발생하지 않도록
전처리를 하기 위해 데이터를 확인하며 EDA 진행

수집한 데이터를 분석하며 발견한 특징과 문제점,

온라인 상담 게시판에서 수집한 데이터의 텍스트 형식이 일정하지 않음.

맞춤법, 띄어쓰기 등 문법 오류가 문장 내에 다수 포함.

법적 분쟁 상황에 대한 설명이 부족하거나, 문장이 부자연스러워 이해하기 어려운 경우가 존재.

법률 질문이 아니거나 심리 상담에 가까운 경우가 존재.

실명, 나이, 주소 등 개인정보가 포함되며 개인이나 단체를 특정할 수 있는 정보가 포함.

Hallucination을 유발할 것으로 생각되는 URL, 전화번호 등의 정보가 포함.

크롤링 과정에서 추가된 유니코드 문자와 개행 문자 등이 포함.

질문 또는 답변 앞뒤에 상담 내용과 관계없는 인사말이 존재.

하나의 게시글에 여러 개의 질문이 포함.

답변자가 여러 명이기 때문에 존칭, 인사말 등의 형식이 다름.



전처리 모듈 개발

1. EDA에서 근거하여 전처리에 앞서 띄어쓰기와 맞춤법 검사를 수행 → 모델의 토큰나이징 특성 고려
2. 유니코드, 개행 문자를 제거하였고, URL과 전화번호를 제거하였다 → Hallucination 완화 + 개인정보 제거
3. 질문이나 답변 앞뒤에 포함되어 있는 인사말을 제거

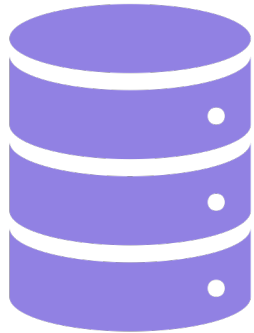
Completed

1. 규칙으로 처리해줄 수 있는 내용은 전처리 모듈을 생성하여 마무리 함.
2. 이외의 규칙은 Heuristic하게 적



향후 계획

- ChatGPT 3.5 모델에 여러개의 법률 QA 데이터를와 이후 프롬프트를 입력하는 few shot prompts를 이용한 데이터 생성



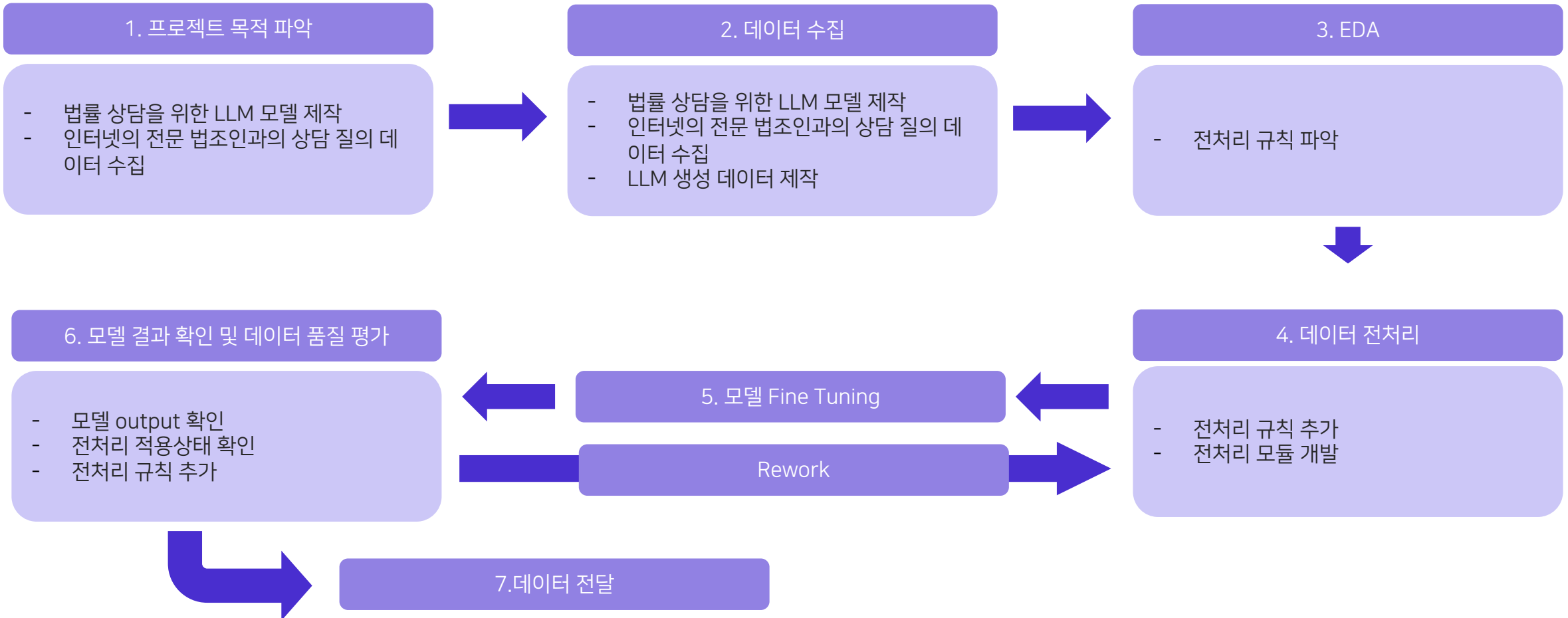
법률 QA 데이터 + Prompt



새로운 QA 데이터
(8616개)

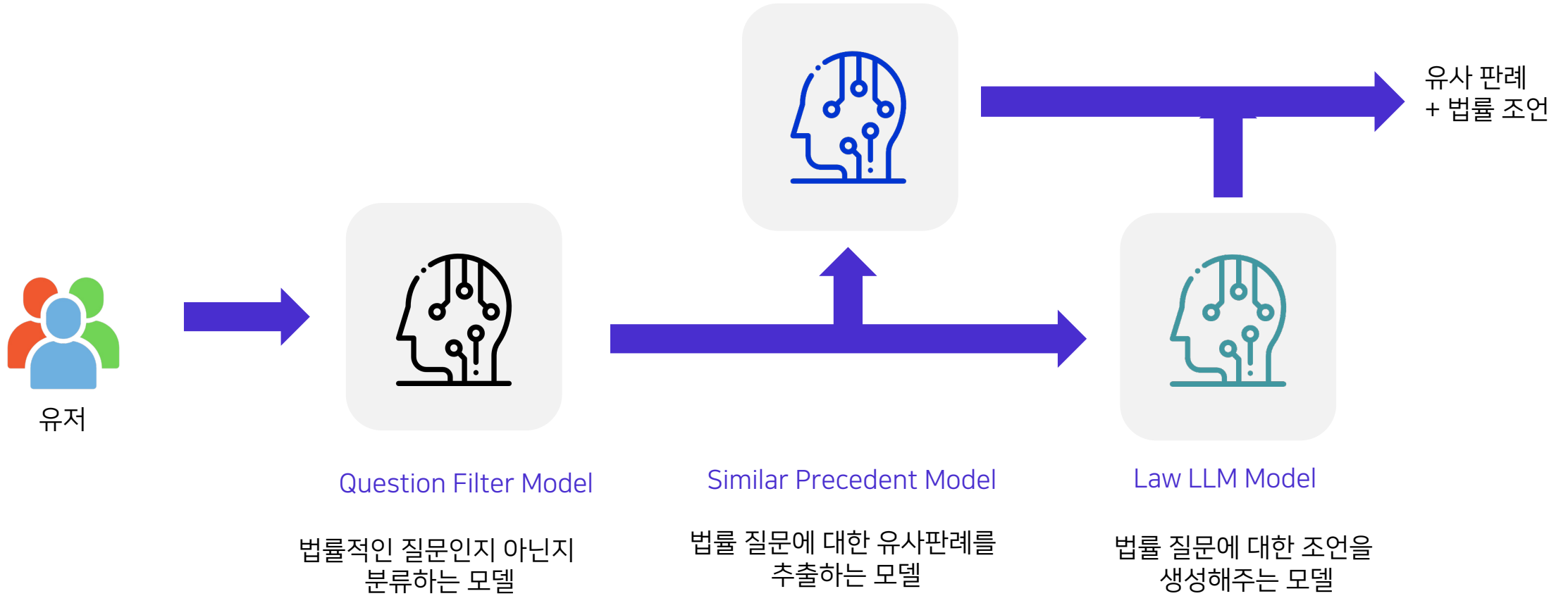


DMOps(Data Management Operation and Recipes)



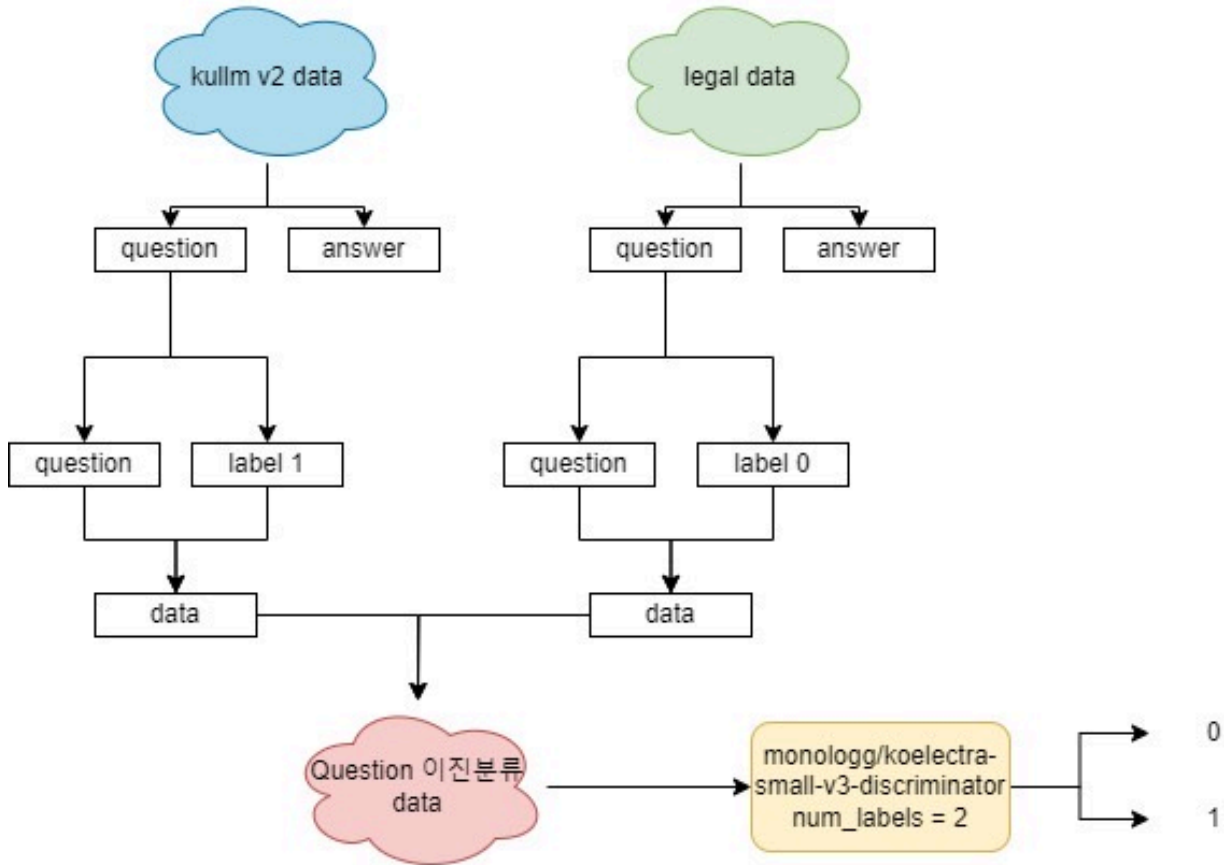
기준 통과

Eujeong Choi, Chanjun Park, DMOps: Data Management Operation and Recipes 구조도 참고하여 작성





Question Filtering Model



모델 설명

[모델 활용 이유]

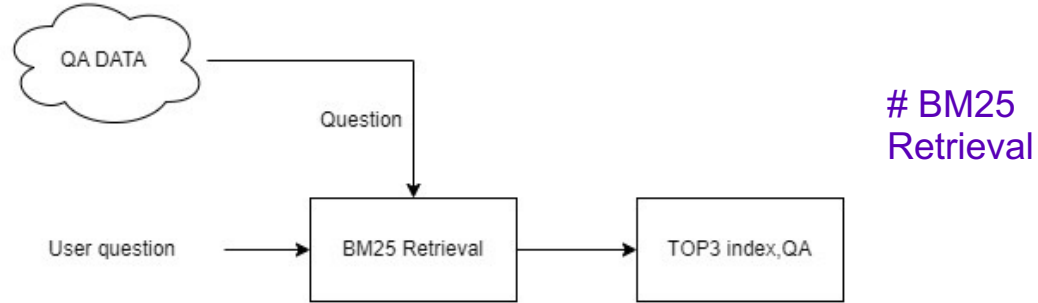
- 법률 질문이 아닌 답변에 대답하지 않기 위해
- 큰 LLM 모델을 사용하여 사전에 분류함으로써
법률질문이 아닌 데이터에 대한 Latency (적은 GPU사용)

[학습 방법]

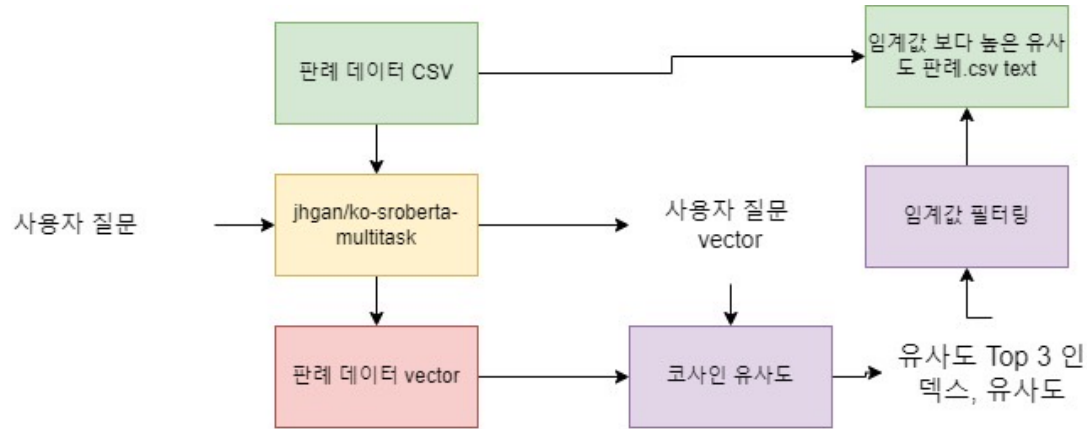
- 법률 질문을 0, 법률 질문이 아닌 질문을 1로 라벨링
- 이진 분류 진행
- Accuracy 98%



Similar Precedent Model



BM25 Retrieval



Cosine Similarity

모델 설명

[모델 사용]

- 유사 판례와 유사 QA를 가져오기 위해 Sentence BERT기반의 모델을 사용
- LLM 답변이 정확하지 않더라도 정확한 유사 판례를 가져오기 위함

[학습 방법]

- 사용자의 법률 질문을 법률 QA 데이터셋에서 유사한 질문을 추출
- 해당 질문의 답변과 사용자의 입력을 concat하여 판례 데이터셋에서 유사도가 높은 상위 판례데이터 Top3를 추출
- 사용자에게 유사한 판례로 제시



법률 조언에 생성 모델이 필요한 이유

[논거]

- 법률 분쟁 상황의 경우 일반화하기 어려움
- 개인마다 매우 특수한 상황을 겪는 일이 대부분이기 때문에 Retrieval을 사용한 기계 독해 같은 태스크로 문제를 해결하기 어려운 상황
- LLM의 경우 기학습된 방대한 정보와 더불어 fine-tuning에 사용한 법률 지식을 사용하여 다양하고 특수한 상황에 유연하게 대응하여 답변을 생성가능

[결론]

따라서 법률 조언을 제공하는 모델은 LLM 모델을 활용하게 되었습니다.



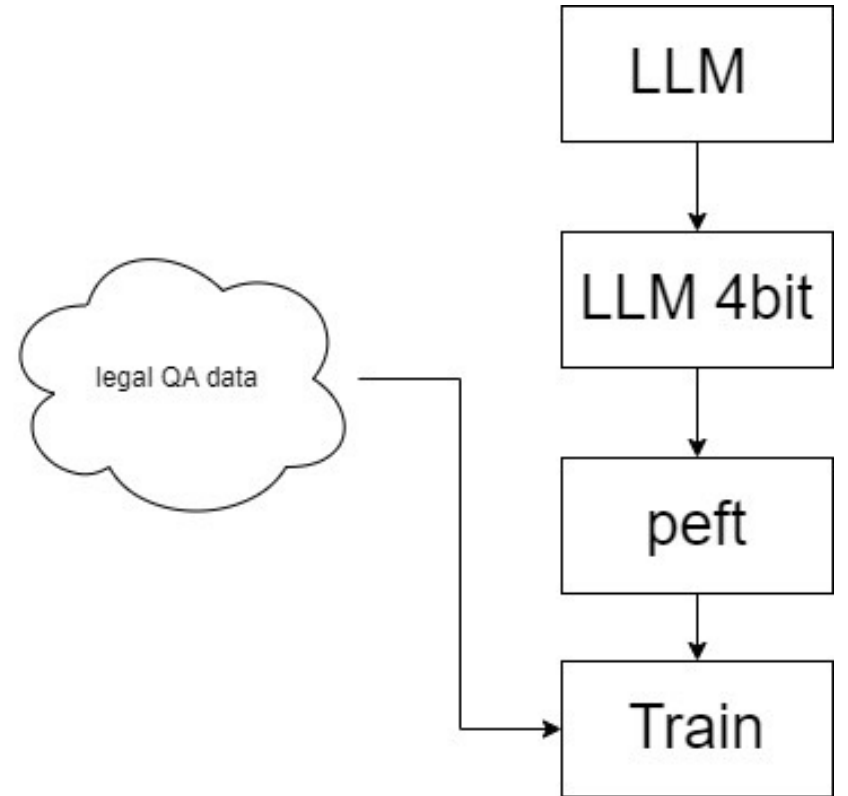
활용한 프롬프트

- **KULLM** : 아래는 작업을 설명하는 명령어입니다. 요청을 적절히 완료하는
 응답을 작성하세요. \n\n### 명령어:\n{x['question']}\n\n###
 응답:\n{x['answer']}

모델 학습 방법

- 현재 훈련은 **LoRA: Low-Rank Adaptation of Large Language Models** 논문을
참고하여 전체 훈련 파라미터의 0.1%만 Fine Tuning 하는 방식으로 학습
- 채팅 형식으로 학습된 모델에 법률 QA 데이터를 학습하여 모델을 구성
- bitsandbytes(4bit) : 32bit에서 4bit로 계산하여 모델의 크기 감소
- PEFT(Parameter-Efficient Fine-Tuning) : 기존의 원본 LLM의 가중치를 동결 → 각 레이어
마다의 추가 선형 레이어(LoRA)를 삽입 → 이를 훈련 시켜 VRAM의 감소와 훈련
계산량의 감소로 모델을 Full Fine-Tuning에 가까운 학습 가능
- 모든 파라미터를 학습하는 방식이 아니기에 기학습에 사용된 프롬프트를 그대로 학습에
사용.

Law LLM Model





학습한 모델 리스트

	모델	데이터	세부 내용
1	kfkas/Legal-Llama-2-ko-7b-Chat	GPT-3.5,easy_law, legal_QA, 크롤링 3차 전처리 데이터	7b / kfkas/Llama-2-ko-7b-Chat
2	level-3-KuLLM-5.8B-tae-2epoch	GPT-3.5,easy_law, legal_QA, 크롤링 3차 전처리 데이터	5.8b/ nlpai-lab/kullm-polyglot-12.8b-v2
3	uomnf97/LawBot-level2-final-preprocessing-v3	easy_law, legal_QA, 크롤링 3차 전처리 데이터	5.8 billion / nlpai-lab/kullm-polyglot-5.8b-v2
4	uomnf97/LawBot-level1-preprocessed	easy_law, legal_QA, 크롤링 전처리 데이터	5.8 / nlpai-lab/kullm-polyglot-5.8b-v2
5	kfkas/LawBot-level1	legal_QA, easy_law, 크롤링 원본 데이터	5.8 / nlpai-lab/kullm-polyglot-5.8b-v2
6	kfkas/LawBot-v1_koalpaca_legalQA_easylaw_cro	koalpaca_, legal_QA, easy_law, 크롤링 원본 데이터	5.8 billion / nlpai-lab/kullm-polyglot-5.8b-v2
7	LawBot-v1_koalpaca_legalQA_easylaw	legal_QA,easylaw	5.8 billion / nlpai-lab/kullm-polyglot-5.8b-v2



모델 평가지표 1

[모델 평가지표 선정기준]

- 언어의 자연스러움도 중요하지만 **도메인 특성상 법률적인 정확도가 가장 중요한 평가요소**
- Legal Tech 에서 작업을 할 때에 **변호사들과 협업**으로 진행

[Dialogue Evaluation Metric]

- **따라서 변호사 현직자에게 모델의 정확도 채점을 요청**
- 채점 데이터는 20개 문항의 학습되지 않은 데이터로 평가하며 다음과 같은 기준을 가지고 직접 validation data에서 뽑음
 1. 모델이 답변하기 어려운 데이터
 2. 전형적인 법률 상담 질문
 3. 고른 분포로 한 분야(예: 사기)에 치우치지 않도록 선택
- 이전 Kullm LLM 모델의 **Dialogue Evaluation Metric**을 참고하여 **도메인에 맞게 변형하여 활용**



💡 두 사람 간의 대화가 주어집니다. 다음의 지시문(Instruction), 입력(Input)을 받게 될 것입니다. 그리고 지시문과 입력에 대한 응답(Response)이 제시됩니다. 당신의 작업은 응답을 평가 단계에 따라 응답을 평가하는 것입니다. 이 평가 기준을 꼼꼼히 읽고 이해하는 것이 중요합니다. 평가하는 동안 이 문서를 계속 열어두고 필요할 때 참조해 주세요.

평가 기준:

- 이해 가능성 (0 - 1): Input에 기반하여 Response를 이해 할 수 있나요?
- 자연스러움 (1 - 3): 사람이 자연스럽게 말할 법한 Instruction 인가요?
- 맥락 유지 (1 - 3): Input을 고려했을 때 Response가 맥락을 유지하나요?
- 흥미롭기 (1 - 3): Response가 지루한가요, 아니면 흥미로운가요?
- Instruction 사용 (0 - 1): Instruction에 기반하여 Response를 생성 했나요?
- 전반적인 품질 (1 - 5): 위의 답변을 바탕으로 이 발언의 전반적인 품질에 대한 인상은 어떠한가요?

평가 단계:

1. Instruction, Input, 그리고 Response를 주의깊게 읽습니다.
2. 위의 평가 기준에 따라 Response를 평가합니다.

Instruction:

{{instruction}}

Input:

{{input}}

Response:

{{response}}

[구름의 대화 평가 metric]



안녕하세요, 변호사님.

법률 상담 챗봇 'Lawbot'을 개발하고 있는 '윤슬'팀입니다. 본 설문조사에서는 20문항의 상담질문에 대한 언어모델의 답변을 평가해주시면 됩니다. 평가를 하실 때에는 아래의 평가 기준을 참고하여 점수를 매겨주세요.

100 평가 기준:

1 언어 모델에 대한 평가

답변 적절성 (0 - 4): 질문 상황에 맞는 답변을 제공하나요?

자연스러움 (0 - 4): 사람처럼 자연스러운 표현을 사용했나요?

맥락 유지 (0 - 4): 답변이 문장 간의 맥락을 유지하고 있나요?

2 법률적 전문성에 대한 평가

정확성 (0 - 4): 법률적 사실을 근거로 한 정확한 답변인가요?

근거 적합성 (0 - 4): 필요한 법률적 근거 (예: 조항, 법률적 근거, 예시 등)를 제공했나요?

전반적인 품질 (0 - 4): 답변의 전반적인 품질은 어떠한가요?

[LawBot의 평가 metric]



모델	언어 모델에 대한 평가			법률적 전문성에 대한 평가		전반적인 품질
	답변 적절성	자연스러움	맥락유지	정확성	근거 적합성	
ChatGPT	3.1	2.95	3.05	2.2	1.1	1.35
BARD	2.95	3.0	3.0	1.25	0.65	1.0
uomnf97/LawBot-level1-preprocessed	1.95	2/35	2.05	0.8	1.1	0.2
kfkas/LawBot-v1_koalpaca_legalQA_easylaw	2.3	2.9	2.6	1.2	1.05	1.15
YoonSeul/LawBot-level-3-KuLLM-5.8B-tae-2epoch	1.9	3.3	2.85	1.2	0.95	0.9
uomnf97/LawBot-level2-final-preprocessing-v3	2.05	2.5	2.65	1.0	0.8	0.65
kfkas/legal-llama-2-ko-7b-Chat	1.9	3.3	3.35	1.7	1.7	1.5

- 각 모델을 모델 A, B, C ... 로 가려 블라인드 테스트 진행
- 법률 모델의 전문성+ 전반적인 품질을 최우선으로 고려, 비슷할 때에 언어 모델에 대한 점수로 비교
- 개발했던 모델 뿐만 아니라 ChatGPT, BARD의 모델의 평가와 비교하여 Benchmark로 활용
- 생성모델 데이터 + 전처리 데이터로 학습한 llama모델이 제일 높은 성능을 보였음.



모델 평가지표 2

[모델 평가지표]

- 정성적인 평가와 동시에 **정량적인 metric**을 통한 언어 모델에 대한 평가가 필요
- **법률적인 정확도를 반영할 수 있어야 함**

[Perplexity]

- PPL은 LM 모델을 평가하는 지표로서 Language Models are Unsupervised Multitask Learners(GPT-2)의 논문에서도 PPL을 사용
- 다음 단어를 맞추는 형식으로 만약 법률 데이터로 학습이 안되면 법에 관련한 단어들만 적어 확실히 맞추는 확률이 감소
→ 법의 관련한 단어들만 많이 출현할 수록 이 모델이 법에 대한 정보와 지식을 많이 가지고 있다라고 가정하여 평가 진행



모델 평가지표 2

[결과]

- PPL의 결과로써 법적인 단어가 많이 포함 될수록 PPL의 성능은 높아지는 경향성을 보여주었다.

Table 1: Performance Evaluation of LawBot Models

Model	PPL 1024	PPL 512	PPL 256
kfkas/legal-llama-2-ko-7b-Chat	7.84	5.7	9.31
YoonSeul/LawBot-Level-3-KuLLM-5.8B-tae-2epoch	5.01	5.21	5.89
uomnf97/LawBot-Level2-Final-Preprocessing-v3	4.9	5.1	5.61
uomnf97/LawBot-Level1-Preprocessed	4.9	5.1	5.6
kfkas/LawBot-level1	5.06	5.29	5.89
kfkas/LawBot-v1_koalpaca_legalQA_easylaw_cro	5.33	5.56	6.16
kfkas/LawBot-v1_koalpaca_legalQA_easylaw	5.33	5.57	6.18

05/ Project Procedure - 협업 규칙 1



윤슬 / Level 3 최종 프로젝트

Kanban Board

- 각 카테고리에서 수행할 Task를 간단히 **키워드로 정리**해주세요.
- 세부 내용은 Issue에 작성중이라 작성할 필요는 없지만, 추후 고도화를 위해서 참고할만한 자료가 있다면 추가해주세요.
- 카테고리 업무에서 수행중인 부분에서 어떤 **Level**, **담당자**, **진행상황** 을 추가해주세요.
- 진행 상황은 스크림, 피어세션 업데이트해주세요.

Kanban Board | 데이터 | 내 작업보기 | 타임라인 | Detail KanbanBoard | 표 보기 +

Project Kanban Board

레벨 + 필터 추가

할 일 11 | 진행 중 8 | 완료 27

- 모델 13
- 데이터 11

</> 모델 입력데이터 포맷 지정
태민 김, 강민재, 신혁준

</> 검수
신혁준

</> 법률 QA API를 통한 데이터 생성
강민재

+ 새로 만들기

</> Generative AI로 학습 데이터
강민재

</> DMOps 구조도 제작
신혁준

+ 새로 만들기

</> 대한법률구조공단 법률상담사레 웹사이트 크롤링
신혁준

</> 데이터 띄어쓰기 교정, 맞춤법 교정
신혁준

법률구조공단 크롤링 허락 메일 보내기
신혁준

</> 데이터 전처리 1차
강민재, 신혁준

</> 대한법률구조공단 사이버 상담 웹사이트 크롤링

윤슬 / Level 3 최종 프로젝트

할 일 11 | 진행 중 8

+ 그룹 추가

회의록

회의

생성일 | 참석자 | 유형 | 생성자

- 07.21 회의록 | 피어세션 | 5월 24일
- 07.20 회의록 | 피어세션
- 07.19 회의록 | 피어세션
- 07.18 회의록 | 피어세션
- 07.17 회의록 | 피어세션
- 07.14 회의록 | 피어세션
- 07.13 회의록 | 피어세션
- 07.12 회의록 | 피어세션
- 07.11 회의록 | 피어세션
- 07.10 회의록 | 피어세션
- 07.07 회의록 | 피어세션
- 07.06 회의록 | 피어세션
- 07.05 회의록 | 피어세션
- 07.04 회의록 | 피어세션
- 07.03 회의록 | 피어세션

+ 새로 만들기

Ground Rule 중요 사항 Wrap-Up

- Ground Rule 읽어볼 것
- Agile Process는 각 날짜의 아침 데일리 스크럼까지

Feature Branch는 어디서 생성?

- develop branch에서 git pull 을 통해 로컬 브랜치에 원격 브랜치 내용을 업데이트 한 뒤, git branch 를 통해서 로컬에 issue번호를 붙인 새 브랜치를 생성한다 이후 git checkout , git push -u origin 명령어를 통해서 원격브랜치를 생성한 뒤 진행해주면 됩니다.

Feature Branch를 삭제할까?

- 사유 : 모든 작업이 마무리 되면 main , develop 브랜치만 남게되는데 외부 인원이 우리의 프로젝트를 볼 때 어떤식으로 git flow를 관리 했는지 확인하기 어려우니 feature 브랜치도 남겨서 확인할 때 체크할 수 있도록 하면 좋겠다.(포트폴리오를 위해)
- 고려할 부분 : 포트폴리오 + 작업효율(feature가 너무 많아지지 않을까)
- 어차피 commit 메시지에 로그가 남지 않나?(의문점) - main로그에는 남지 않고, 브랜치 삭제여부는 외부에서 알 수 없음. 하지만 closed issue 나 develop branch 에서 merge할 때 어떤 브랜치에서 오는지 확인 가능
- 결론 : 우선 보류. Level 1이후에 결정

디렉토리 구조 정하기

- 디렉토리는 아래와 같은 방식으로 진행되고 모든 디렉토리 내부 내용은 레벨 담당자가 정리한다
 - datapipeline 담당자들은 모듈 이름이 겹치지 않도록 조심해주세요.
 - model 담당자는 유사도 검사 모델과 LLM 모델을 나눠주는걸 권장합니다.
 - 기존 Aistage에서 진행했던 대회의 Project 1, 2 내부와 유사하게 만들어주면 좋을 것 같습니다.
 - KLUE-RE Project처럼 쌓아갈 수 있도록

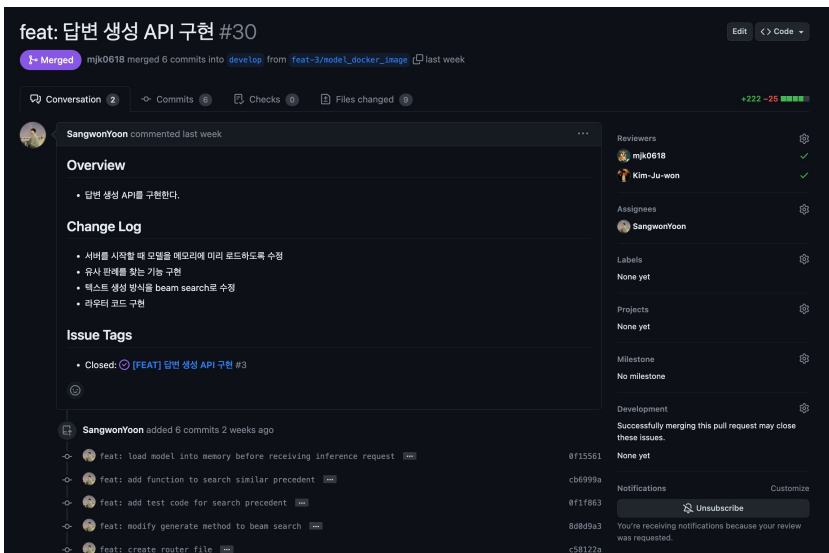
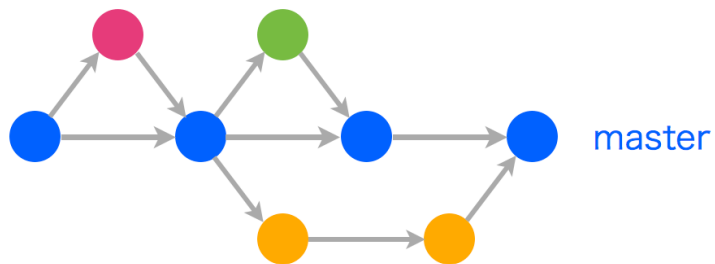
```

level3_final_Project
├── backEnd
└── frontEnd
    
```

- Kanban Board를 이용하여 체계적으로 To do List 관리
- 노션 협업기구를 활용해 회의 및 기록 체계화



GitHub flow



☀️ Ground Rule ☀️

🔥 1. main branch Pull Request 관련 Rule

- Level 단위로 develop branch에서 main에 merge한다
- 배포용 브랜치로 develop에서 refactoring, bugfix가 완료 된 상태에서 push한다.
- main 브랜치로 merge는 팀원 4명 중 2명이상 accept한 경우 merge를 진행한다.

🔥 2. Develop branch Pull request 관련 Rule

- feature 에서 기능이 모듈화가 완료된 경우(feat → refactor) Pull Request를 진행한다.
- feature branch에서 develop branch로 merge를 진행할 때 conflict가 생길 경우 관련한 컴퍼들이 해당 이슈를 해결한다. (새로운 feature branch를 사용 혹은 이전 feature branch에서 내용 수정)
- main 브랜치로 merge는 팀원 4명 중 2명이상 accept한 경우 merge를 진행한다.

🔥 3. feature Branch Commit 관련 Rule

- feature 브랜치는 develop branch에서 **feat-이슈번호/브랜치이름** 형식으로 생성한다.
 - ex) feat-12/crawler, feat-1/skeleton
- commit 메시지의 Header(변경내용)와 Footer(이슈번호)는 필수, Body는 선택사항
 - 참고 [commit convention](#)
- header의 prefix는 **feat, fix, docs, style, refactor, perf** 사용
- 하나의 commit은 코드의 유의미한 변화가 있는 최소 크기로 나누어서 올린다.

```

# 예시 :
git commit -am "refactor: modify api handler logic

- change logic 1
- change logic 2

#1"
    
```

05/ Project Procedure - 기술 스택 & 개발 환경



구분	기술 스택	사용언어	환경 및 개발 방법(툴)
백엔드	Fast API	Python	<ul style="list-style-type: none"> - OS: Ubuntu Linux (Docker) - Tool : Visual Studio Code
프론트 엔드	react.js, tailwindcss	Javascript, HTML, CSS	<ul style="list-style-type: none"> - OS: Ubuntu Linux(EC2), MacOS - Tool : Visual Studio Code - Figma : 와이어프레임, UI/UX 프로토타입
데이터 파이프라인	selenium, BeautifulSoup, re	Python	<ul style="list-style-type: none"> - OS: Window - Tool : Visual Studio Code
AI 모델	peft, bitsandbytes, huggingface, transformer	Python Javascript	<ul style="list-style-type: none"> - OS: Window, Linux - GPU: V100 5대, RTX 4090 - Tool: PyCharm
Others	-	-	<ul style="list-style-type: none"> - GitHub(CI 파이프라인 구축) - AWS(EC2 Proxy Server 구축, 웹 배포) - Nginx(웹 어플리케이션 배포)



팀원	구현 완료한 업무
강민재	데이터 탐색, 대한법률구조공단 사이버 상담 웹사이트 QA 데이터 크롤링 모듈 구현 전처리 모듈 구현, 데이터 파이프라인 구축, 학습 데이터셋 구축
김주원	Project Manager, 노션 협업 기구 관리(Kanban Board), 깃헙 관리(PR/ISSUE Template 등록, GitHub 관리 전략 수립), UI/UX 개발, 프로토타이핑, 프론트엔드 구축, LLM 모델 평가지표 개발, 반응형 웹 페이지 제작, LLM 모델 학습 및 실험
김태민	LLM 모델 평가지표 개발(PPL), BM25/Bert 유사도 기반 리트리버 개발, LLM 모델(koalpaca,llama, kullm) 학습 및 인퍼런스 코드 구현, LLM 모델(koalpaca,llama, kullm) 파인튜닝
윤상원	답변 생성 API 구현, pytest 기반 단위 테스트 구현, self-hosted runner를 이용한 CI 파이프라인 구축, 로드 밸런싱 적용, LLM의 프롬프트 엔지니어링, FastAPI 기반 백엔드 서버 구축, 모델 서버와 router간 ssh 터널링 구축, Airflow를 이용한 재학습 파이프라인 구축 , Auto Scaling을 통한 Failover
신혁준	데이터 탐색, 대한법률구조공단 법률상담사례 데이터 크롤링 모듈 구현, 맞춤법 검사기 크롤러 모듈 구현, 파이프라인 구축, 학습 데이터 셋 구축

05/ Project Procedure - Project Timeline

내용	6월 1주차	6월 2주차	6월3주차	7월 1주차	7월 2주차	7월3주차	7월4주차
아이디어 계획							
아이디어 계획							
개발 계획							
웹 프론트엔드 개발							
와이어프레임 작성							
UI/UX + 프로토타이핑							
웹 프론트엔드 페이지 구축							
데이터 파이프라인							
데이터 탐색 & 크롤러 모듈 개발 및 데이터 수집							
데이터 전처리							
생성모델 기반 데이터 증강							
모델							
LLM 법률 조언 모델 구축							
BERT/BM25 Retrieval 기반 유사판례 추출모델							
BERT기반 질문 Classifier							
프로덕트 서빙							
백엔드 구축(API 개발)							
CI/CD 파이프라인 구축							
로드 밸런싱							
Airflow를 이용한 재학습 파이프라인							



LawBot

- 법률 상담
- Sign In
- Sign Up
- Upgrade to Pro
- Documentation
- Help

법률 문제를 입력해주세요.

Send

유사 판례 조항 1

사건 케이스 :
사건 번호 :
사건 분류 :
관련 법 조항 :

Read more →

유사 판례 조항 2

사건 케이스 :
사건 번호 :
사건 분류 :
관련 법 조항 :

Read more →

유사 판례 조항 3

사건 케이스 :
사건 번호 :
사건 분류 :
관련 법 조항 :

Read more →



[결과 & 자체 평가 의견]

- 서비스에 필요한 우선순위를 두어 3주간의 시간안에 목표했던 서비스 구현에 성공했습니다.
- 하지만, 법률 서비스를 제공하기에는 LLM 모델의 정확도가 낮은 편이라 성능 개선이 필요해보입니다.
- Airflow를 이용한 재학습 파이프라인을 구현하여 높은 수준의 MLOps 시스템을 구축할 수 있었습니다.
- 다양한 협업 툴을 효과적으로 사용하여 협업을 진행하였습니다.
- 트래픽 등 다양한 에러 상황을 가정하여 시스템을 구축하였습니다.
- 데이터를 눈으로 확인하며 규칙을 찾고 전처리를 만들며 데이터에 대한 인사이트를 기를 수 있었습니다.

[개선 방안 측면 개선 방안]

질문 히스토리 조회 기능

- 사용자가 이전에 질문했던 내용을 조회할 수 있는 기능을 추가할 예정입니다. 이 기능을 위해서 DB 서버를 모델 서버와 분리하여 모든 모델 서버가 DB 서버에 접근할 수 있도록 아키텍처를 구성할 예정입니다.

[데이터 측면 개선 방안]

멀티 쓰레드 데이터 파이프라인

- 멀티 쓰레드를 이용한 데이터 크롤링 파이프라인을 구축한다면 보다 효율적인 데이터 파이프라인을 구축할 수 있을 것 입니다.



[모델 축문 개선 방안]

RLHF

- 실제 변호사가 직접 데이터를 만들어 점수를 평가하는 보상모델을 만들고 RLHF를 이용하여 학습시킨다면 성능을 개선할 수 있다고 생각합니다.

Full Finetuning Pretrained model

- 대부분의 한국 Pretrained model에는 한국법률이 추가되지 않았습니다. 따라서 많은 Hallucination과 정확하지 않은 답변을 출력하게 됩니다. 이를 해결하기 위해 고성능의 GPU를 이용하여 처음부터 한국 법률을 포함한 Pretrained model을 만들어 사용할 경우 더 높은 정확성을 달성할 것이라고 생각합니다.

많은 파라미터를 가지고 있는 모델 사용

- 실제 서비스되는 대형 LLM의 경우 30B, 65B, 70B의 대규모 언어 모델로 서비스가 진행됩니다. LLM의 특성상 모델의 파라미터가 많을수록 보다 높은 정확성을 달성하게 되는데 현재 가진 GPU의 한계로 최대 12.8B이며 인퍼런스의 경우 5.8B를 사용할 수밖에 없었습니다. 고성능 GPU를 사용하여 많은 파라미터를 가진 모델을 사용하게 되면 성능이 향상될 것입니다.

DPR

- 유사판례 및 유사 QA를 가져올 때 허깅페이스의 cosine similarity , BM25 Retrieval을 사용하여 추출하고 있습니다. 따라서 Dense Embeddin을 활용하여 DPR을 사용할 경우 이전보다 높은 유사도를 가진 유사판례 및 QA를 추출할 수 있을 것 입니다.

지금까지 발표를
들어주셔서
감사합니다.