

Data Scientist & ML Engineer

끈기로 극복하는 개발자

윤도연

010-3219-9511

ydy89899@gmail.com

CONTENTS

1. 자기소개
2. 경력
3. 보유 기술
4. 참여 프로젝트 목록
5. 주요 프로젝트
6. 기타 참고사항

자기소개



안녕하세요.
항상 끈기로 승부하는 개발자, 윤도연입니다.

다양한 도메인 경험

4년차 데이터 과학자로서, 대기업 반도체 프로젝트부터 의료데이터, 추천 도메인과 자연어에 이르기까지 다양한 task와 도메인을 경험해왔습니다. 다양한 문제를 해결하는 과정에서 문제의 본질을 빠르게 파악하고 인사이트를 찾아내는 것을 즐깁니다.

비전공자의 노력

수학과 출신으로서, 처음 데이터 과학을 접했을 때 생각보다 낫게 느껴지는 진입장벽으로 재밌었습니다. 하지만 배울수록 엔지니어링 역량의 부족함을 누구보다도 절실히 느꼈습니다. 이를 채우기 위해 결국 데이터 과학자도 ‘개발자’임을 인지하고 항상 끈기로 업무 공백을 채우려 노력합니다.

동료와 지식 나누기

사내 DS팀에서 동료들과의 성장을 중요시하며, 정기적으로 세미나와 스터디를 진행했습니다. 자연어 처리를 중심으로 한 기술 전파를 통해 팀의 역량을 강화하고, 개인적으로도 부족한 엔지니어링 지식 공백을 채우기 위해 MLOps 스터디도 진행하며 함께 발전하고자 노력합니다.

경력



CAREER

- 2021.11 ~ Present 라이앤캐쳐스 근무
- 2019.06 ~ 2020.03 에이치비 근무
- 2018.11 ~ 2018.12 포항공대 인공지능연구원 인턴

EDUCATION

- 2017 광운대학교 수학과 석사 졸
- 2015 광운대학교 수학과 학사 졸
- 2008 서울과학기술대학교
기계시스템디자인학과 입학

EXPERIENCE

- 2021 Naver AI Boostcamp 1기
 - KLUE: Relation Extraction 프로젝트 참가
 - KLUE: Dialogue State Tracking 프로젝트 참가(1위)
 - KLUE: Deep Knowledge Tracing 프로젝트 참가(4위)
- 2019 아산병원 의료 Segmentation 대회 참여 (5위)
- 2018 포항공대 인공지능연구원 교육 4기
- 2017 내일배움카드 국비지원 머신러닝 교육 수료

AWARDS

- 2022 대-스타 해결사 플랫폼 준우승

보유 기술

머신러닝 & 딥러닝

- Tensorflow, Pytorch를 활용한 AI architecture 구현 및 커스텀 가능
- HuggingFace를 활용한 pre-trained 모델 활용 및 fine-tuning 경험 다수
- 기타 ML 관련 라이브러리 사용 가능

API 구현 및 배포

- Python, FastAPI를 활용한 API 구현
- Langchain, Langserve를 활용한 LLM 프로젝트 설계 및 배포 경험
- Docker 활용 가능

협업 및 프로젝트 관리

- JIRA, HEIGHT를 활용한 업무 처리 가능
- Git(Github, Gitlab, Bitbucket)을 활용한 소스코드 버전관리 가능
- Confluence, Notion, Slack을 활용한 협업 경험 다수

데이터베이스 & 벡터DB

- MongoDB, Redis
- FAISS, Milvus, ChromaDB 사용 경험

참여 프로젝트 - 업무

프로젝트명	기간	설명	역할 및 기여
LLM 기반 교육 플랫폼 구축 사업	2024.07 ~ 현재	Langchain과 Hyperclova X를 사용한 RAG 시스템 구축	LLM 모델 서빙, RAG 시스템 배포, 인프라 관리 및 고도화
AI 채용서류 평가 자동화	2023.06 ~ 2024.04	기업 서류 자동 평가를 위한 14개 과업 개발 및 고도화	언어모델 fine-tuning, 구현체 메모리 성능 개선
문서 클러스터링을 위한 유사 문서 랭킹	2023.01 ~ 2023.04	문서 산 유사성을 계산하는 클러스터링 모델 구현 및 서빙	Reranking을 위한 Simcse 모델 구현 및 서빙
대-스타 해결사 플랫폼 데이터 증강 대회	2022.09 ~ 2022.12	중소벤처기업부 주관 'NLP 모델 개발을 위한 텍스트 증강 모델 개발' 대회	다양한 증강 기법 양상화, 빠른 베이스라인 구축
와인나라 추천 시스템 구축	2021.12 ~ 2022.04	와인 구매 데이터를 활용한 추천 시스템 개발	추천 모델 개발, cold start 문제 개선
도서 추천 서비스 '비블리' 고도화	2021.11 ~ 2021.12	사용자 데이터를 기반으로 한 추천 시스템 고도화	언어모델을 활용한 자연어 메타데이터 임베딩으로 성능 개선
반도체 이상탐지 모델 개발	2019.09 ~ 2020.03	반도체 공정 이상 탐지를 위한 AI모델 구현	이미지 데이터의 Auto-encoder를 활용한 이상탐지 모델 구현

참여 프로젝트 - 개인

프로젝트명	기간	설명	역할 및 기여
ChatPDF 클론 코딩	2023.06 ~ 2023.08	FAISS를 사용하여 Streamlit에 간단한 RAG 시스템 배포	Ollama를 활용하여 EEVE 모델 사용
비-글 문제변환	2022.06 ~ 2022.09	사내 문제변환 사이드 프로젝트(성경체, 급식체, 아재체 etc)	BentoML로 모델 배포, 모델 학습
KLUE: Dialogue State Tracking	2021.04 ~ 2021.06	매 턴마다 사용자 대화 턴으로부터 “Slot-Value” 쌍을 추론하는 task	SOM-DST 구현 및 fine-tuning, 실험관리
KLUE: Relation Extraction		문장 내 두 entity의 관계를 분류하는 task로 진행한 대회	모델 구현, 전처리 및 데이터 증강
Cardiac Segmentation 대회	2018.09 ~ 2019.01	아산병원 주관 의료데이터(심장 CT image)를 segmentation 하는 대회	.mha format 3d 이미지 전처리, 3D-UNet 모델 구현
Senticle		뉴스 데이터를 활용한 주가 상/하락 예측	1DCNN 모델 구현, LIME 적용

몇 가지 주요 프로젝트 소개

1. LLM 기반 교육 플랫폼 구축 사업
2. KLUE: Dialogue State Tracking(DST) Competitions
3. 유사 문서 검색 및 랭킹 API 개발
4. 대-스타 해결사 플랫폼
5. AI 채용 서류 평가 자동화 프로젝트

(1) LLM 기반 교육 플랫폼 구축 사업 - 1

프로젝트 기본 정보

프로젝트명	Langchain 기반의 AI 조교 서비스 구축
기간	2개월 +
참여인원	8명(기획2, 디자인1, 프론트2, 백엔드2, LangChain 서버1)
설명	글로컬대학에 구축할 LLM 기반 AI 조교 서비스 개발
역할	클러스터링 모듈 파이프라인 설계 모델 구현 및 학습(SimCSE) API 서빙 구현

기술 스택

- 프레임워크 : Langchain, Langserve
- 데이터베이스 : Redis, Milvus
- 모델 : HyperclovaX, ChatGPT
- 협업 : Gitlab, Slack, Height
- 라이브러리 버전관리 : Poetry
- 인프라 : Jenkins, Docker

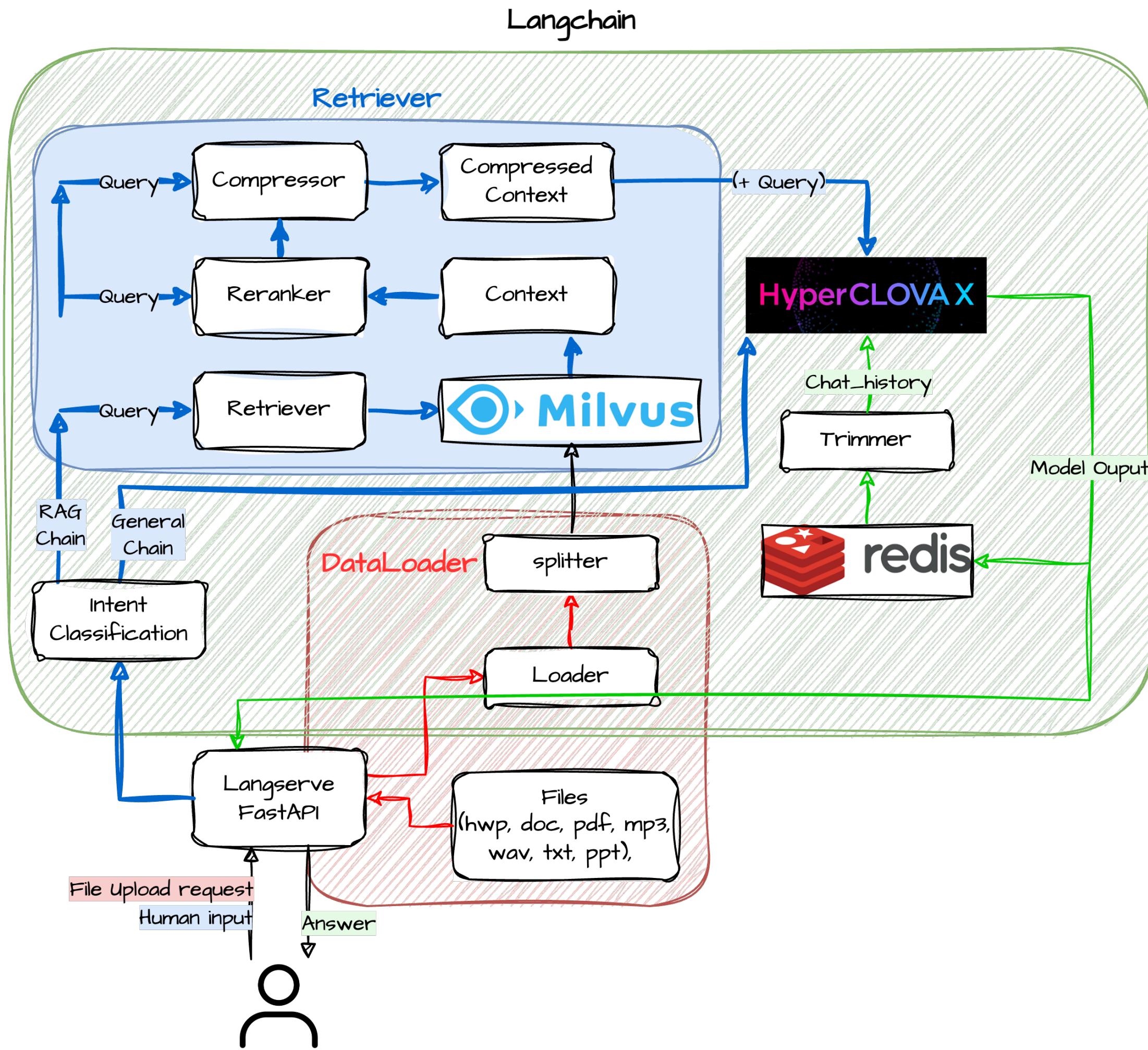
Task 설명

AI 조교 역할을 수행할 수 있도록 하는 RAG 기반의 교수 도우미 서비스입니다. 대학에서 요구하는 강의계획서, 논문초안, 강의자료, 시험문제, 과제 템플릿에 맞게끔 출력물이 작성되어야 합니다. 물론 일반적인 대화도 가능해야 합니다.

- Langchain과 HyperClova X를 활용한 AI 조교 서비스 구축 PoC입니다.
- 그 중 LangChain 서버 구축을 담당하였습니다.
- LangChain 서버에서 필요한 기획 사항
 - Langchain에서 지원하지 않는 모델인 HyperClova X의 streaming 기능
 - 다양한 포맷 지원(HWP, PPT, DOC, PDF, TXT, 유튜브 링크, MP3 등)
 - 멀티턴 대화
 - RAG 지원(벡터스토어 구축)
 - 벡터스토어로부터 유저별, 대화 세션별로 별도 검색하는 기능
 - 파일 **사전** 업로드 : 유저의 모든 세션에서 검색 가능한 파일
 - 파일 **세션** 업로드 : 해당 세션 내에서만 검색 가능한 파일

(1) LLM 기반 교육 플랫폼 구축 사업 - 2

Architecture



주요 개발 사항

- 랭체인에서 지원하지 않는 HyperClovaX를 래핑하여 **스트리밍 기능** 구현
- 벡터DB에 **다양한 포맷** 적재되도록 로더 커스텀 구현
- 토큰 제한에 신경쓰며 Retriever 구현
 - input+output 토큰의 합이 4093 토큰인 HyperClovaX의 제한으로 인해 RAG context 추출에 주의하며 구현 진행
 - 약 2천 토큰 정도의 할당만 가능하여, **hybrid search** 후 **reranker**와 **compressor**로 토큰 절약과 정확성 동시에 확보
- **멀티턴** 기능 구현
 - TTL 1시간 설정된 Redis에서 대화 이력 최신순으로 로드하여 모델 추론
- 사용자 utterance를 실행적으로 **의도 분류**하여 inference 속도 절약 및 정해진 시나리오대로 유도하도록 구현

(1) LLM 기반 교육 플랫폼 구축 사업 - 3

회고

Bad

- 성능 개선보다 “구현”에 초점을 맞추며 개발했던 점이 아쉽습니다. 촉박한 PoC 일정으로 인해 기능 구현에 초점을 뒀지만, 그 와중에 다양한 실험을 통한 성능 개선까지 가져갔으면 어땠을까 하는 아쉬움이 남습니다.
- 초반에 확장성을 고려하여 개발하지 못했던 점이 시간 낭비로 이어졌던 점이 살짝 아쉽습니다. Advanced한 RAG 서버 구축은 개인적으로도 처음 이었기에 심리적 여유가 없어 확장성을 고려하지 못했습니다. 추후 기획안이 변경되면서 기능이 붙는 과정에서 아차싶어 다시 돌아가는 과정에서 낭비된 시간이 아쉬운 점입니다.

Good

- 처음 Langchain으로 모든 기능 구현이 가능하다고 생각했던 것과는 달리 커스텀 요소가 많았음에도 생각대로 구현 했다는 점에서 잘했다는 생각이 듭니다. 생각보다 불친절한 공식 도큐먼트로는 부족하여, 라이브러리를 뜯어보면서 구현했던 점이 많은 도움이 되었습니다.

Keep

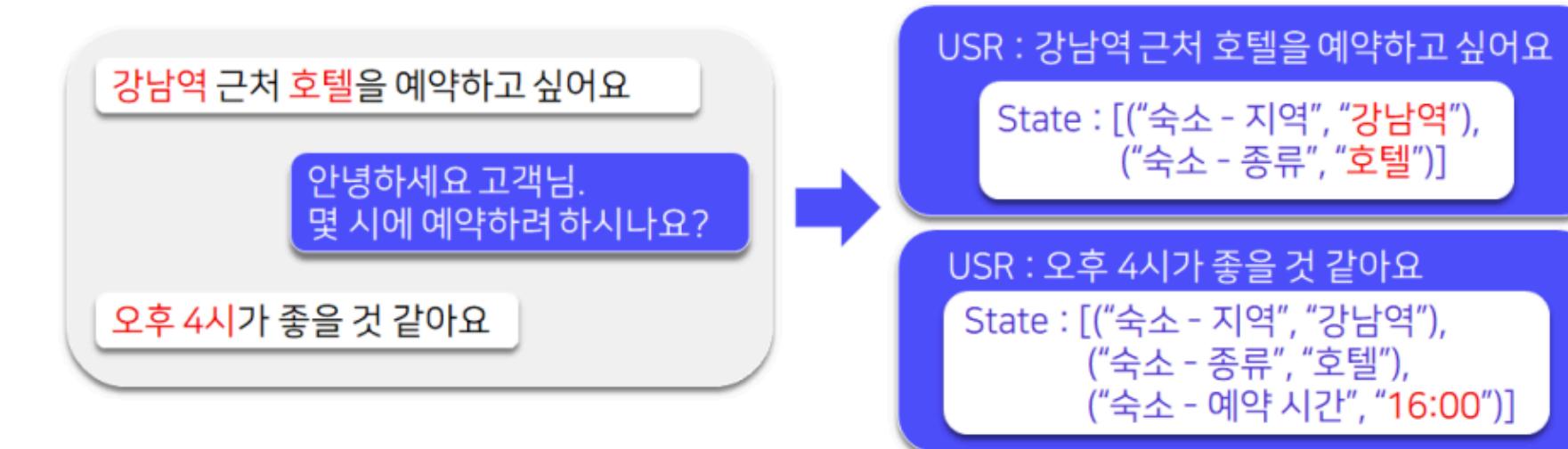
- 평소 관심있게 Langchain 커뮤니티 등을 살펴봤던 점이 짧은 프로젝트 일정을 맞출 수 있게 도와준 것 같다고 생각했습니다. 기능을 구현하는 것은 또 다른 문제이지만, 어떤 것이 있는지 알았기에 해당 기능을 커스텀할 수 있다고 생각하게 되었습니다. 운영 및 고도화 관점에서 언제 어떤 문제를 맞닥드릴지 모른다는 생각으로 평소처럼 여러 기술들을 follow up 하는 습관을 들여야겠다고 생각합니다.

(2) KLUE: Dialogue State Tracking(DST) - 1

프로젝트 기본 정보

프로젝트명	KLUE:Dialogue State Tracking Competition
기간	1개월
참여인원	6명
설명	사용자 멀티 턴 대화로부터 “Slot-Value” 쌍을 추론하는 task
역할	SOM-DST 구현 및 fine-tuning SOM-DST + CoCo 적용 및 fine-tuning SOM-DST + TAPT 적용 및 fine-tuning

Task 설명



DST는 대화의 의도를 파악하고 대화 상태를 추적하는 기술입니다. 대화형 AI 시스템이나 챗봇에서 주로 사용되며, 사용자가 여러 턴(turn)의 대화를 이어갈 때, 대화의 문맥과 상태를 유지하는 역할을 합니다.

- 시스템과 유저의 대화가 쌓이면서, 대화로부터 얻은 **정보**를 누적하게 됩니다.
- 정보 리스트는 (**Domain, Slot, Value**)로 구성되는 **Pair**를 원소로 갖습니다
- 대화가 진행되는 동안 **Pair**는 **추가** 되기도, **수정** 되기도, **삭제** 되기도 합니다.
- 대화가 끝난 시점의 정보 리스트(**STATE**)를 예측하는 것이 목표입니다.

(2) KLUE: Dialogue State Tracking(DST) - 2

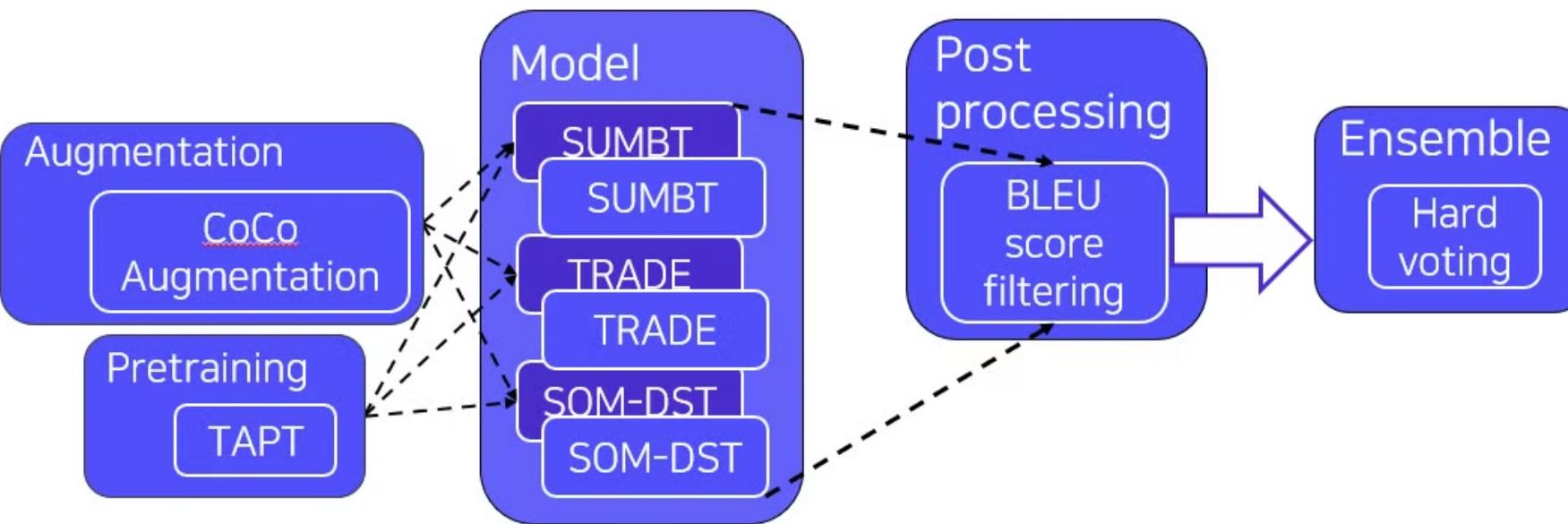
Datasets

- 여행사와 여행객의 대화에서 수집된 데이터입니다.
- 데이터는 아래와 같은 형식으로 구성되어 있습니다.
- JSON 형식의 multi-turn 대화 원문과, 대화 말미에 예측해야하는 정보 (State)를 라벨데이터로 가집니다.
- 예측해야 하는 State(GT)는 **(Domain, Slot, Value)**의 순서로 구성되어 있습니다.
 - Domain : 5개 Class
 - Slot : 45개 Class

```
{
  "dialogue": [
    {
      "role": "user",
      "state": [
        "숙소-가격대-저렴",
        "숙소-종류-모텔",
        "숙소-지역-서울 북쪽"
      ],
      "text": "서울 북쪽에 있는 저렴한 가격대의 모텔에 예약하고 싶어요"
    },
    {
      "role": "sys",
      "text": "안녕하세요. 더 원하시는 조건이 있으신가요?"
    },
    {
      "role": "user",
      "state": [
        "숙소-가격대-저렴",
        "숙소-종류-모텔",
        "숙소-지역-서울 북쪽"
      ],
      "text": "네. 주차장이 있는 곳이면 좋겠어요."
    },
    {
      "role": "sys",
      "text": "네. 원하시는 조건의 숙소로 힐 모델과 모델 킹이 있습니다. 모델 킹의 경우 가격대비 서비스가 좋아 3.8점의 좋은 평점을 받고 있는데 예약 해 드릴까요?"
    },
    {
      "role": "user",
      "state": [
        "숙소-가격대-저렴",
        "숙소-종류-모텔",
        "숙소-지역-서울 북쪽",
        "숙소-예약 요일-금요일",
        "숙소-예약 명수-5",
        "숙소-예약 기간-3",
        "숙소-이름-모델 킹"
      ],
      "text": "네. 모델 킹으로 예약 해 주세요. 금요일 5명으로 3일 예약 해주세요."
    }
  ]
}
```

- Json Format - Each examples consists of information below:
 - dialogue_idx - str** : unique id for a dialogue
 - domains - List[str]** : single or multi domains for the dialogue
 - dialogue - List[Dict]** : turn-level information that contains attributes below:
 - role - str** : a speaker of the turn, user or sys
 - text - str** : an utterance (transcription)
 - state - List[str]** : dialogue-state formed in DOMAIN-SLOT-VALUE

Team Strategy

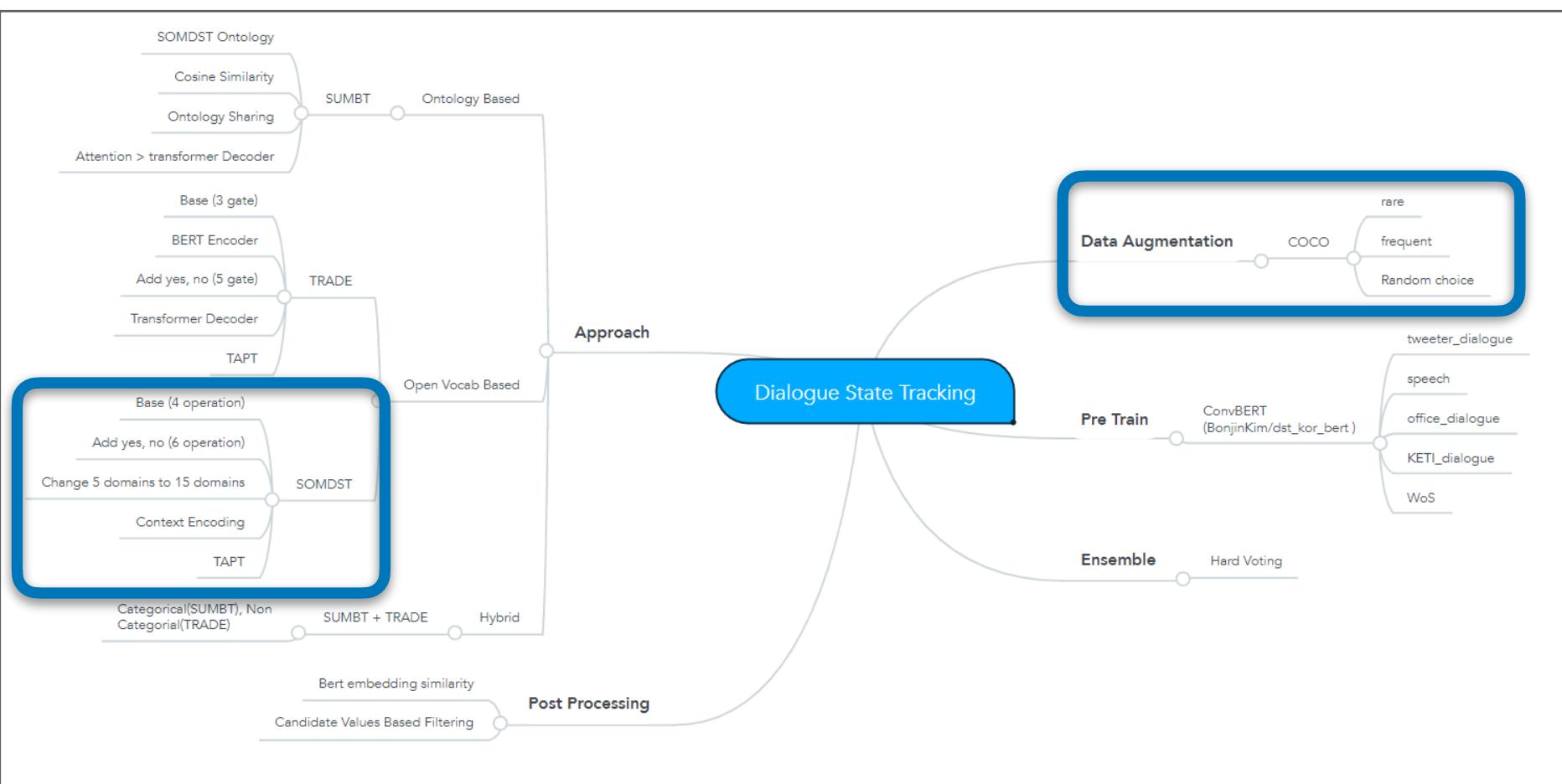


- 전통적으로 DST Task는 두 가지 타입의 모델 아키텍쳐가 존재합니다.
 - 온톨로지 기반** : Domain-Slot 쌍을 미리 만들어 놓고, 그 안에서 분류
 - Open-vocab 기반** : 생성 모델로써, Domain-Slot 쌍을 학습하고, 모델이 직접 생성
- Ontology 방식의 모델(SUMBT, TRADE)과 Open vocab(생성모델) 방식의 SOM-DST 모델을 각각 학습 후 양상을
- 3가지 모델에 대하여 전부 CoCo Augmentation, TAPT 적용**
- 생성형 모델인 SOM-DST에 대해서는 예측 결과에 대해 후처리 진행**
- 전체 모델 6종류를 양상을하여 예측 결과 출력

(2) KLUE: Dialogue State Tracking(DST) - 3

나의 역할

- 위 'Team Strategy' 중 SOM-DST 관련 모든 구현체 담당
- SOTA 모델인 Open vacab 기반의 SOM-DST 모델 구현 및 성능 재현
- SOM-DST 싱글 모델 이외에 CoCo augmentation 적용
- SOM-DST 모델에 TAPT를 적용한 학습 진행



- 위 바운더리 영역에 해당하는 역할 수행

회고

아쉬운 점

- 구현 속도 이슈로 인해 더 테스트 해보고 싶은 것들을 해보지 못한 점.
- SOM-DST의 architecture는 현재 턴과 이전 턴을 비교한다.
- 여기서 끝나지 않고 전전 턴까지 총 3개 턴을 비교해보지 못한 점이 아쉽다

모델 아키텍쳐 커스텀 역량 습득

- 논문을 바탕으로 하나의 아키텍쳐를 깊게 판 결과 다양한 아이디어가 떠 올랐고, 해당 아이디어들을 그대로 구현하던 시행착오를 겪으면서 모델 커스텀에 대한 역량을 끌어올렸습니다.

(3) 유사 문서 검색 및 랭킹 API 개발 - 1

프로젝트 기본 정보

프로젝트명	유사 문서 검색 및 랭킹 API 구현
기간	1개월
참여인원	1명
설명	사내 서비스 내 작가의 글들 간의 유사도를 바탕으로 클러스터링 해주는 API 구현
역할	클러스터링 모듈 파이프라인 설계 모델 구현 및 학습(SimCSE) API 서빙 구현

Task 설명

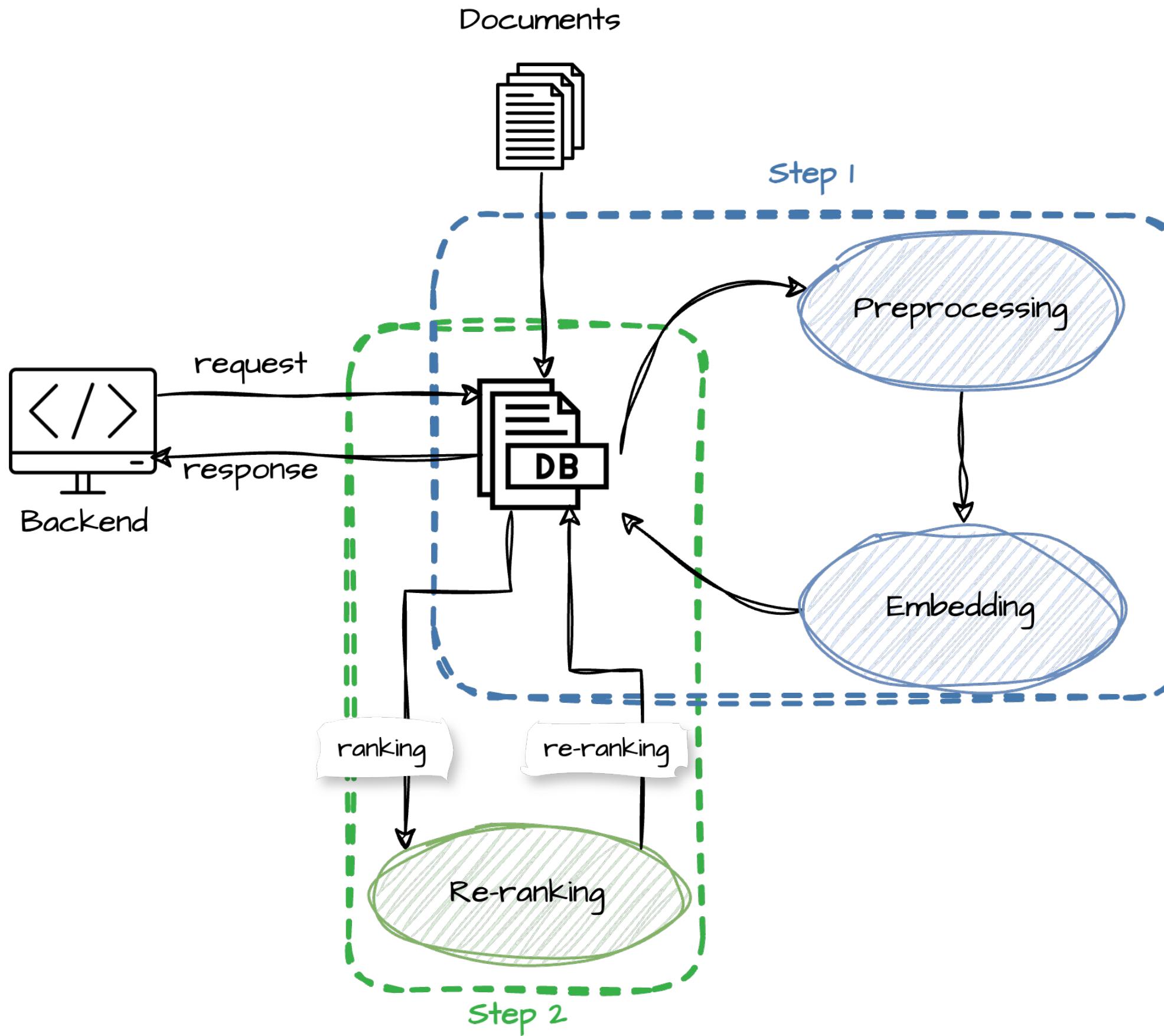
블록체인을 결합한 사내 지식 거래 플랫폼에 필요한 “**유사 문서 검색 및 랭킹**” 모듈을 API 형태로 서빙하는 부분을 구현하였습니다.

플랫폼 내에서는 특정 주제에 대한 의견이나 견해를 먼저 작성하는 작가들의 글의 가치가 높게 부여되도록 유도하고자, 유사한 문서들끼리 클러스터로 묶일 수 있도록 하고, 클러스터 중 가장 먼저 작성된 글이 클러스터의 최상단으로 노출되도록 기획하였습니다.

이를 위해서 문서들의 클러스터링을 단순히 태그나 주제 카테고리로 묶는 것이 아니라 의미적 유사성을 계산하고 묶어주기 위한 모델 개발 task를 진행했습니다.

(3) 유사 문서 검색 및 랭킹 API 개발 - 2

Pipeline



DB 적재 샘플

_id	text	preprocessed	emb_bert	emb_simcse	candi_ind	rerank_ind	rank_score
0	개인의 판단에 큰 영향을 끼치는 두 가지 심리적 현상을 우리 모두가 잘, 간접적으로...	[[-0.48644745349884033, -0.43016865849494934, ...]	[[0.08876019716262817, 0.1867300122976303, -0...]	[1, 3, 7, 5, 6, 2, 9, 0, 8, 4]	[1, 3, 5, 6, 2, 9, 0, 8, 4]	[1, 0, 0.54918, 0.54635, 0.53936, 0.42311, 0.33...	
1	어렸을때부터 전 환자를 무척이나 싫어했습니다.理财产品를 좋아하는가? 외...	[-0.10283223539590836, -0.2286939173936844, 0...	[[0.23231926560401917, 0.628568708896637, -0.4...	[2, 9, 5, 7, 6, 2, 9, 1, 0, 8]	[2, 9, 7, 1, 4, 6, 5, 3, 0, 8]	[10, 10, 0.42543, 0.39478, 0.37736, 0.3657, ...]	
2	월 천 모금이 월천리가 되고 삶이 하는 시대! 낸신인기술(T+7)업	[[0.30570387840270996, 0.20778916776180267, 0...	[[0.28200551867485046, -0.08288934081792831, -...	[3, 7, 5, 1, 2, 9, 9, 6, 0, 8, 4]	[3, 7, 5, 1, 2, 9, 6, 0, 8, 4]	[1, 0, 0.54679, 0.53852, 0.51937, 0.40045, 0.40...	
3	말로만 들으면서 감탄하고 신기해 했었는데 오늘은 제대로 대화를 한 번 해 보기로 하...	[[0.595695972442627, -0.0624887767710686, -0...	[[0.2748512327671051, -0.12438580393791199, -0...	[4, 5, 7, 2, 9, 3, 6, 0, 8, 1]	[4, 7, 2, 9, 6, 5, 3, 1, 0, 8]	[10, 0.55686, 0.5112, 0.5112, 0.47415, 0.4013...	
4	테크 광용들은 인공지능(AI)에 전력을 쓰는 와중에도 각자의 길을 걷고 있다.\\...	[[0.05398641899228096, 0.7409682273864746, -0...	[[0.16070157289505005, 0.12148624658584595, -0...	[5, 7, 3, 2, 9, 1, 4, 0, 8, 6]	[5, 7, 3, 1, 2, 9, 6, 0, 8, 4]	[10, 0.57775, 0.50494, 0.47359, 0.37531, 0.37...	
...
98	많은 사이트들의 경우 방문자의 최소 절반이 검색연진을 통해 유입되는데, 찾았어 완전...	[[0.07103759795427322, 0.39540907740592957, 0...	[[0.025500331073999405, 0.404475092878784, -0...	NaN	NaN	NaN	NaN
99	ChatGPT [나는 모르는 질문에 답변하기 위해 거짓말을 한다. 이 점에 대해 ...]	[[0.4194827973842621, -0.7322870492935181, -0...	[[0.295594722032547, 0.21586757898330688, -0.2...	NaN	NaN	NaN	NaN
100	작년 조은 HRD라는 학상 새운트로트란에 민감하게 반응해야 하는 입장에 속해 있으...	NaN	NaN	NaN	NaN	NaN	NaN
101	\nChatGPT-3.5가 2022년 11월 30일에 출시된 이후, AI 기능이 불교...	NaN	NaN	NaN	NaN	NaN	NaN
102	많은 사이트들의 경우 방문자의 최소 절반이 검색연진을 통해 유입되는데, 찾았어 완전...	NaN	NaN	NaN	NaN	NaN	NaN

개발 고려사항

- 랭킹을 계산할 때 “쌍방 유사함”이 아니라, **나중에 들어온 문서가 먼저 들어온 문서에 유사하다는 기획을 유지하도록 순차적 유사도 계산 고려**
- 긴 문서와 짧은 문서의 비교가, 짧은 문서간의 비교와 다르게 계산되는 현상 고려하였습니다.
- 의미적 유사도 성능을 끌어올리기 위한 모델 선정 고려

(3) 유사 문서 검색 및 랭킹 API 개발 - 3

역할

- 한국어 사전 학습 모델(koelectra)을 사용하여 candidate retriever 구현.
- SimCSE 구현 및 학습 진행 후 re-ranker 구현.
 - 데이터셋: KorNLI, KorSTS
- 문서 인덱스를 요청하면 그와 유사한 문서 인덱스를 내뱉는 API 구축
- 전체 파이프라인 구현 및 설계
 - Infra : Docker, FastAPI
 - DB : MongoDB
 - ML : Pytorch, HuggingFace

회고

Bad

- 서베이가 부족했던 점이 아쉽습니다. 당시 RAG가 핫 해지기 이전의 프로젝트였기에 Vector DB의 존재(FAISS 정도만)를 알고는 있었지만, 서베이 기간에 떠오르지 않았습니다. 레이턴시 이슈나 성능을 비교해볼 수 있던 기회를 놓친 것이 아쉽습니다. 최신 기술이나 현황을 follow up 해야한다는 것의 중요성을 다시 느꼈습니다.

Good

- 프로젝트 직전 MLOps 스터디를 통해 Docker를 활용한 유기적인 환경 구축에 대해 공부한 뒤, 실무에 접목 시켰다는 점이 개인적으로 만족스러웠습니다.

Keep

- 유독 이 프로젝트에서 문서 정리를 꼼꼼하게 해놨던 점이 후에 타 프로젝트에서 많은 도움이 되었습니다. 이후 문서 정리에 대한 중요성을 더욱 체감하며 습관이 바뀌는 계기가 되었습니다.

(4) 대-스타 해결사 플랫폼 - 1

프로젝트 기본 정보

프로젝트명	대-스타 해결사 플랫폼
기간	2개월(2022.09.05 ~ 2022.10.25)
참여인원	2명
설명	중소벤처기업부 주관 의도 분류 task를 위한 데이터 증강 대회
역할	Text Classification 베이스라인 구축 사내 문체변환 모델을 활용한 데이터 증강 그 외 증강 기법 구현 및 테스트(Back Translation, TEM 등) Wandb를 활용한 실험관리

Task 설명

중소벤처기업부에서 주관한 ‘인공지능’ 분야 4개 과제 중, 대기업 - 스타트업이 협력 파트너를 찾도록 도와주는 지원 프로그램으로써 **롯데정보통신의 “자연어 인공지능 모델 개발을 위한 텍스트 증강 모델 개발” 과제**에 참여하였습니다.
대회 형식으로 진행되며, 자연어 의도 분류 모델의 성능을 높이기 위해 기본 제시된 데이터를 증강하는 것이 task의 개요입니다.

기술 스택

- 라이브러리 : HuggingFace
- 실험환경 : Docker, jupyter notebook,
- 모델 : klue/roberta-large in hf.
- 협업 : Notion
- 라이브러리 버전관리 : Poetry

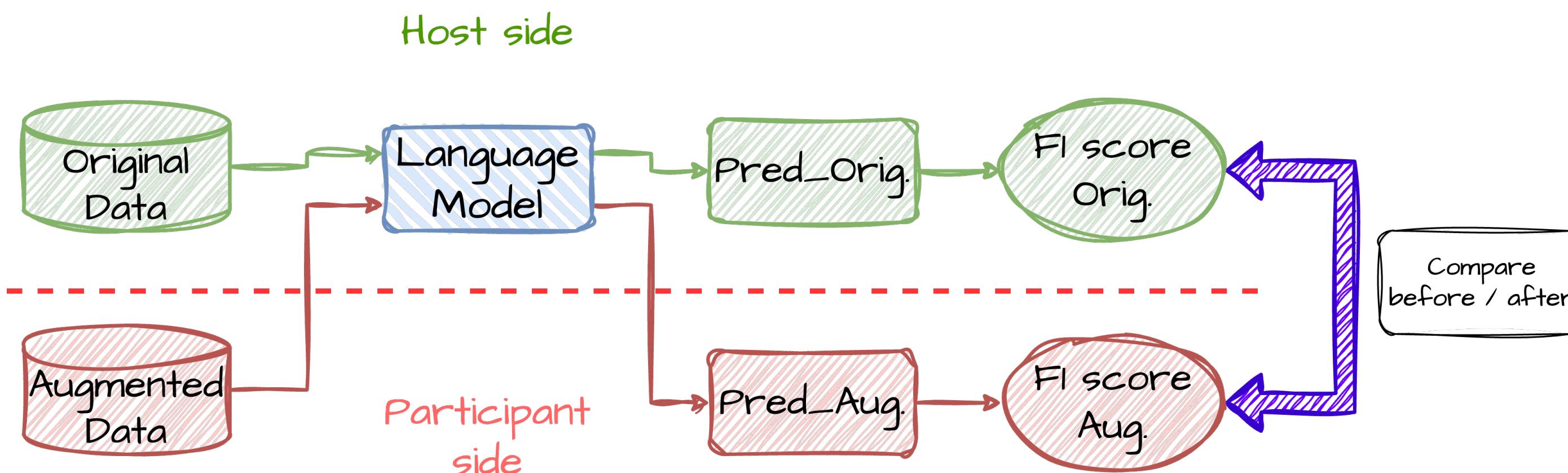
(4) 대-스타 해결사 플랫폼 - 2

대회 진행 방식

1. 주어진 공공데이터(AI HUB)를 증강합니다.
2. 증강된 데이터를 제출하면 주관사의 언어모델로 의도분류 task를 진행합니다.
다. (참가자는 LM이 어떤 아키텍쳐인지, 어떤 사전학습 모델인지 모름)
3. 동일 모델로 평가한 데이터 증강 전/후 F1-score를 각각 계산 후 비교합니다.
다. (증강 전 대비 몇 퍼센트 성능 개선인지)
4. 모든 참가자들의 제출은 단 1회 제출로 판별납니다.

개발 고려사항

- 출제자의 언어모델을 모르기 때문에 최대한 general한 모델을 구현하는 것을 고려
- 최대한 다양한 증강 기법을 시도할 것을 고려
- 실험관리를 철저하게 할 것을 계획하였음
- 어차피 증강 된 데이터로 학습하는 것은 우리가 아니기 때문에, 데이터의 양보다 질에 집중함.
- 증강 데이터를 fine-tuning할 때 하이퍼파라미터나 옵션 등에 대해 알 수 없는 상황이었기 때문에 무차별적인 데이터 양의 증가보다 퀄리티에 집중하였음.



(4) 대-스타 해결사 플랫폼 - 3

나의 역할

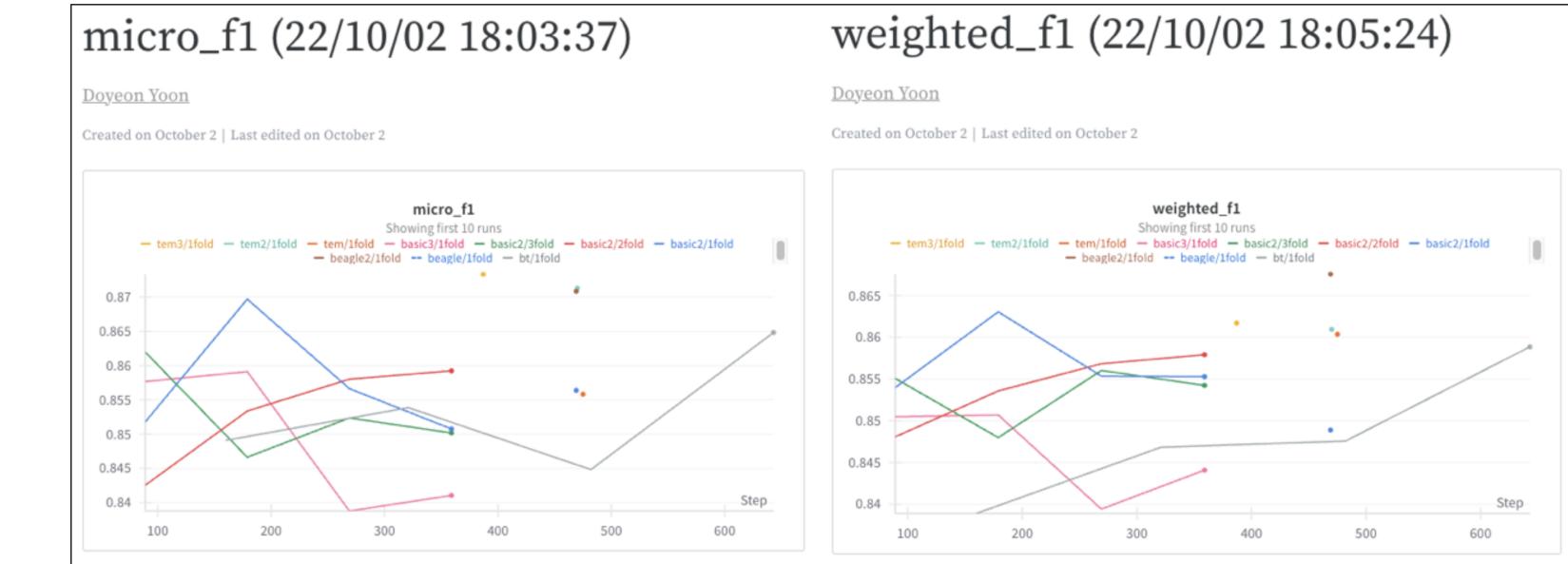
베이스라인 모델 구현

- Pytorch로 기본적인 Intent classification task에 맞는 베이스라인 구축
 - 사용한 pretrain 모델 : klue/roberta-large
 - 파라미터 튜닝 : 기본 데이터셋에서 가장 좋은 성능을 내는 하이퍼파라미터로 설정
- Wandb로 실험 로깅이 되도록 구현

증강체 구현 및 실험

- Back Translation 구현
- TEM 구현
- 사내 사이드 프로젝트로 사용했던 문체 변환 모델로 inference
 - 문체 변환은 smilegate-ai/korean_smile_style_dataset을 kobart로 학습한 문체 변환 모델을 사용하였습니다.
 - 초딩/아재/할배/사극체 등의 말투를 내뱉습니다.
- 다양한 조합으로 변인 통제하면서 실험 진행

실험 결과 및 대회 성적



(4) 대-스타 해결사 플랫폼 -4

회고

Bad

- 증강 데이터셋이 많이 발생하는 대회였기에 실험관리를 철저히 신경 썼음에도 불구하고 헷갈리거나 놓친 실험이 있었습니다. 그 원인으로는 잡은 실험으로 인한 데이터 버전관리를 놓친 것이었는데, 시간이 여유로워서 실험을 처음부터 다시 했기에 대회 성적에 영향을 미치진 않았지만 그럼에도 아찔했던 기억이 아쉽습니다.

Good

- 다소 혼선은 있었으나 처음 의도하고 계획한대로 모든 실험 및 구현 일정을 맞춘 것이 좋았습니다. 그럴 수 있었던 이유로는 평소 익숙한 task이기도 했고, 무작정 증강을 하는 것이 아닌 똑같을 것으로 예상되는 언어모델 베이스라인을 구현했던 전략이 잘 맞아 떨어진 듯 합니다. 그간의 연습이나 보유하고 있던 지식들이 실험 아이디어 및 인사이트 도출에 영향을 준 것으로 생각되며, 그래서라도 더욱 최신 기술이나 아이디어들을 체화시켜야한다고 다짐하게 되었습니다.

관련 기사

'자연어 인공지능 모델 개발을 위한 텍스트 증강 모델 개발'에 도전



'대스타 해결사 플랫폼'에 참여한 라이엔캐쳐스 윤도연·이종혁 머신러닝 엔지니어

데이터 테크기업 라이엔캐쳐스(대표 허윤)가 중소벤처기업부가 주관하는 '대스타 해결사 플랫폼' 경진대회에서 파이널리스트에 올랐다. '자연어 인공지능 모델 개발을 위한 텍스트 증강 모델 개발'에 혁신적인 방안을 제시해 준우승이라는 쾌거를 이뤄냈다.

(5) AI 채용 서류 평가 자동화 - 1

프로젝트 기본 정보

프로젝트명	AI 채용 서류 평가 자동화
기간	10개월
참여인원	1명
설명	기업 채용에 사용되는 자기소개서의 위반사항을 자동으로 탐지하는 프로그램 개발
역할	총 14개 과업 사항에 대한 위반행위 검출 모델 구현

기술 스택

- 라이브러리 : HuggingFace, Pytorch
- 실험환경 : Docker, jupyter notebook,
- 모델 : klue/roberta-base in hf.
- 협업 : Notion
- 라이브러리 버전관리 : Poetry

Task 설명

지원자의 채용 서류 위반사항을 자동으로 탐지하고 계산하도록 하는 시스템 구현 프로젝트입니다. 총 14개 과업(표절, 블라인드 위반, 형식 오류 등)으로 이루어져 있습니다. 그 중 핵심적이며 AI 모델이 필요한 Task에 대해서만 정리합니다.

- 표절** 검출 : 임의의 두 문장 속 연속된 6어절 이상이 동일한 경우 해당 문장은 서로를 표절
- 타기업 지원자 글** 검출 : A기업 지원자의 자기소개서에 B기업 지원을 목표로 한 문장 검출.
- 비속어 사용 문장** 검출
- 블라인드 위반 문장** 검출 : 가족직업, 성별, 지역, 학벌 등 노출한 문장 검출
- 지원기업명 오기재** 검출 : 지원하는 기업명을 올바르지 않게 쓴 오타검출

(5) AI 채용 서류 평가 자동화 - 2

개발 고려사항

- 표절 검출 : 메모리 및 속도 이슈에 대한 개발 고려
- 블라인드 위반, 비속어 작성 문장 검출 : 데이터셋의 확보와 레이블링에 대한 명확한 정의를 필수적으로 고려함
 - 부족한 데이터로 인해 증강이 필수적이었음.
- 타기업 지원 문항 검출 : 명확하게 타기업을 지원한 잘못된 문장인지 vs 단순히 경험 작성 중 언급된 타기업명인지를 분류하는 것이 필수 고려사항이었음.
- 자사명 오기재 문장 검출 : 어떤 기준을 갖고 오타인지 아닌지를 판별할 것인지가 관건이었음.

구현 방식 - 1

표절 검출

- 표절의 정의 : “6어절 이상이 같은 두 문장은 서로 표절이다”
- 모든 지원자의 모든 문항에 대해 문장 단위로 자르고, **선택 된 2개 문장**의 표절 여부를 파악하여 계산.
- 전체 문장 중 두개의 문장을 선택하는 방법:
 - greedy한 방법은 매우 오랜 시간이 걸림
 - 조금이라도 시간을 절약하기 위해 문장을 토크나이징 진행. 각 토큰이 포함된 문장의 인덱스를 저장해놓은 뒤, 표절 가능성 있는 문장끼리만 비교

블라인드 위반 검출, 비속어 포함 문장 검출, 타기업 지원 문항 검출

- Intent classification task로 접근하여 pytorch와 huggingface의 사전학습 모델을 사용하여 구현.
 - 사전학습 모델은 모두 klue/roberta-base를 사용
- 단 각 위반사항의 예외 케이스가 정의되어 있어, 각 task별 ban word 스크리닝 후 모델 추론하여 해당 문장의 위반 사항 여부를 threshold로 분류 진행.

(5) AI 채용 서류 평가 자동화 - 3

구현 방식 - 2

자사명 오기재 검출

- AAAA라는 기업명을 AAA**B**A라고 작성했을 때 검출되도록 하는 과업
- Levenshtein distance를 사용하여 구현 진행
 - 두 문자열이 각 자리에서 얼마나 다른지를 카운팅할 수 있도록 해주는 알고리즘
- 오픈소스인 Soynlp를 활용하여 한글을 단순 letter 단위가 아니라, 자모 단위로 전부 분해 하여 초, 중, 종성을 기준으로 Levenshtein 거리를 계산

회고

bad

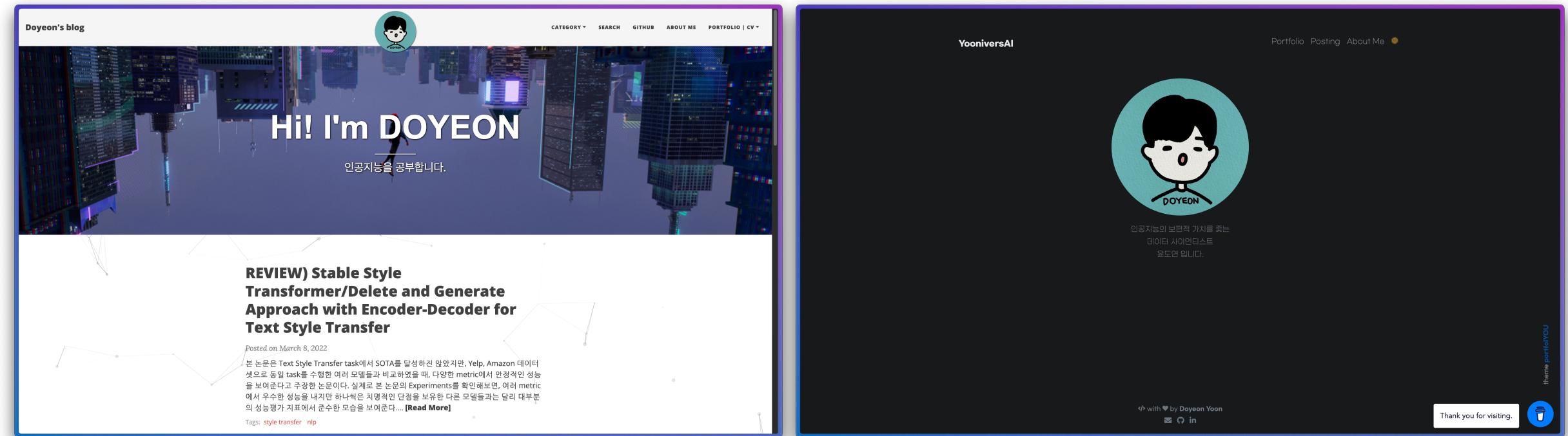
- 함께 투입되려던 두명의 크루원들의 퇴사로 혼자 개발하게 되면서 일정 조율과 기획 그리고 개발을 동시에 하는 것이 힘들었습니다. 클라이언트 측의 요구사항에 감각이 없어 다소 무리한 개발 일정을 소화했던 것이 아쉽습니다.

good

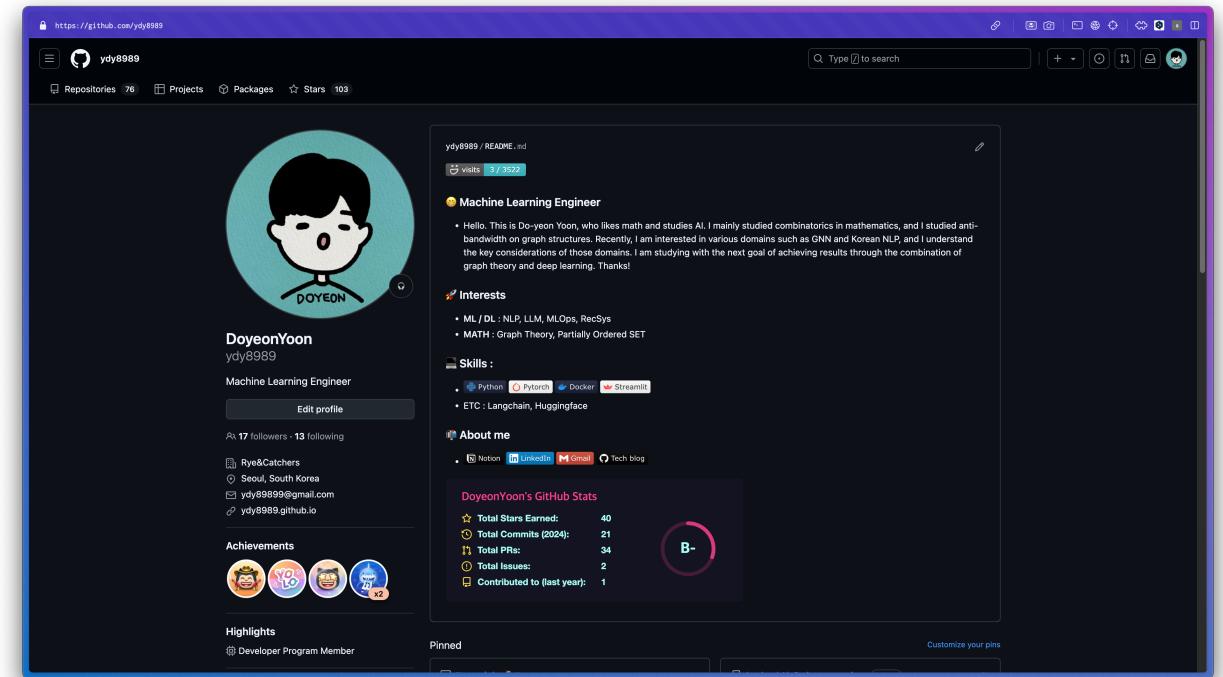
- 프로젝트를 진행하면서 실무에서 개발 기간 중 메모리나 알고리즘에 대해 고민해볼 수 있던 점은 매우 배울 점이 많았습니다. 단순히 ML 모델을 사용하고 말고의 문제를 넘어서, 현재 과업을 해결하는데에만 집중하고 오로지 문제 해결에만 집중하기 위해 이런 저런 아이디어들을 시도했던 경험들이 좋은 경험이었다고 생각합니다.

기타 참고사항

- 블로그: (구) ydy8989.github.io
(신) yooniversai.github.io
* 블로그 이전 진행 중입니다...



- Github: github.com/ydy8989



감사합니다.