졸업작품개발보고서

1. 작품명

국문:울산대학교 챗봇

영문:Ulsan University Chatbot

작품분류 : 인공지능, 자연어처리, 챗봇

Keyword : 인공지능, 자연어처리, 챗봇, koElectra, 웹개발

2. 작품개요

울산대학교 대학생활 정보를 제공해줄 챗봇을 제작한다.

현재 대학생활 정보는 uwins에 흩어져 분포되어 있어서 접근성이 다소 떨어진다. 떨어지는 접근성은 검색의 시간이 늘어나는 결과를 초래한다.

이 결과는 직접 문의를 유도하는데, 직접 문의가 늘어날 수록 학과사무실 근무자의 업무효율 하락을 초래하며 심지어 문의량이 폭주하는 때에는 전화 연결 되는 것 조차 힘든 경우가 많다.

이 현상은 학생 최근 심리상 가급적 전화 문의를 하지 않으려는 추세를 가속 시키며 따라서 학생들은 학생 온라인 커뮤니티(에브리타임)에 질문을 하는 빈도가 늘고 있다.

이 경우 양호한 접근성을 가지지만 답변의 정확도가 의심된다.

따라서 Q&A의 접근성을 향상시키며 동시에 답변의 정확도를 보장하기 위한 울산대학교 챗봇을 제작한다.

챗봇은 빅데이터 기반 신경망 인공지능이며 울산대학교 정보통신원의 자문을 통해 제작하였다.

3. 개발 배경 및 목적

- 졸업작품과 관련된 기존의 국내외 연구 내용

기존의 hugging face의 transformers 기반 NLP 모델은 BERT를 중심으로 연구되었다.

BERT는 MLM (Masked Language Model) 방식으로 훈련된 모델이다.

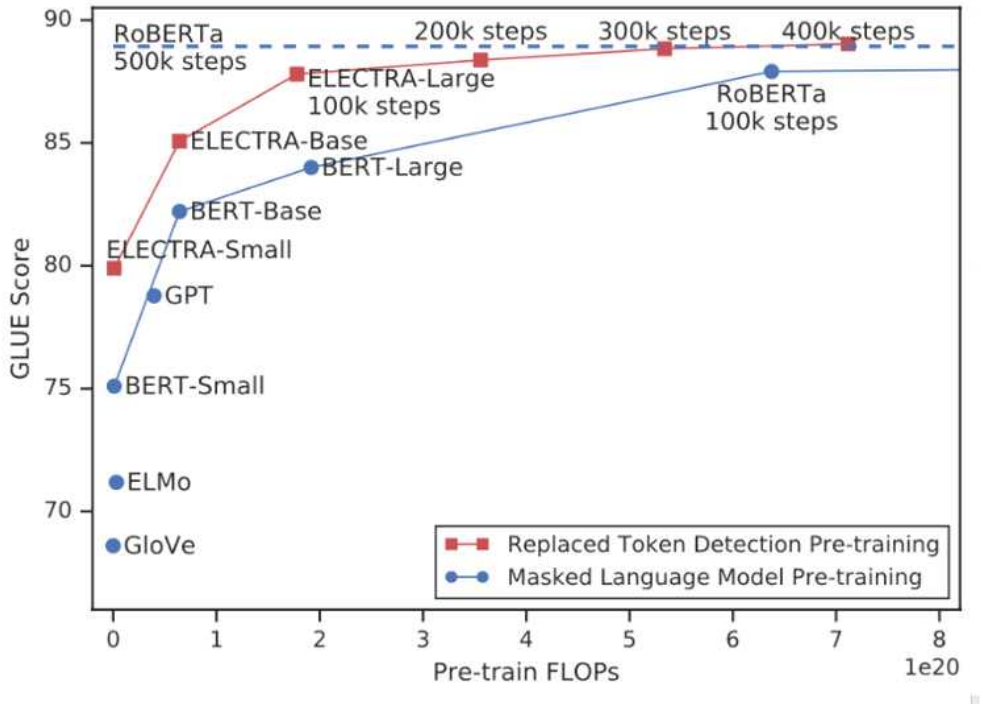
MLM은 입력 토큰(단어)을 특정확률로 손상 시킨 후 오리지널 토큰으로 수정하며 학습을 진행한다.

MLM의 학습 메커니즘은 전체토큰의 15%를 [MASK]로 치환을 하고 마스킹된 문장에 대해서 원래 토큰이 무엇인지 예측 하는 과정이다.

장점은 준수한 성능을 보유하고 있으나 단점은 전체 토큰 중 15%만 학습 시키기에 일반적으로 많은 양의 컴퓨팅 자원이 필요하다. 이 단점은 Low Computing에 취약하다.

본 프로젝트에서 제시하는 NLP 모델은 RTD (Replaced Token Detection) 방식으로 훈련된 모델이다. RTD는 크게 두 개의 신경망 Generator와 Discriminator로 구성되어 있으며 Generator는 앞서 설명한 MLM 방식으로 만든 모델이다. Discriminator 모델은 Generator가 예측한 토큰이 원래 입력한 토큰과 동일한건지 아닌지 이진 분류하는 모델이다. 이렇게 제작한 모델은 MLM 부분은 버리고 Discriminator 부분만 사용하여 fine tuning을 한다.

이 방식은 모든 토큰에 대해서 학습이 가능하기 때문에 MLM에 비해서 컴퓨팅 자원이 적게 필요하다. 따라서 컴퓨팅 자원이 적은 환경에서 더욱 두드러지는 성능을 낼 수 있다.



위 그림은 RTD방식의 모델들이 MLM 방식의 모델보다 모든 컴퓨팅 자원 환경에서 좋은 성능을 가짐을 보여준다. 특히 컴퓨팅 자원이 적을때 더욱 두드러진다.

- 졸업작품의 중요성과 필요성

현재 챗봇은 수많은 서비스가 존재하고 있다. 하지만 현재 울산대학교 자체로 서비스하는 챗봇은 존재하지 않는다. 현재 사용되고 있는 울산대학교 관련 정보를 얻기 위해서 대표적으로 학교사이트에서 직접 찾아 구하거나 관련 부서에 연락하는 행동을 취한다.

학교사이트 검색의 경우 많은 시간이 필요하다. 현재 사이트내의 정보는 충분하다. 그러나 정보들은 여러 곳에 흩어져 있으며 흩어진 정보 중에서 원하는 정보를 찾기까지의 시간은 비합리적이게 소모된다.

관련 부서 문의는 가장 먼저 학생들은 부서마다 어떤 업무를 맡는지 알기힘들기에 질문에 적절한 부서를 찾는 것이 어렵다. 관련 부서를 찾더라도 수강신청기간과 같이 문의량이 폭주하는 기간에는 대기시간 또한 급수적으로 늘어난다.

결론적으로 각각의 방법에서 시간적인 소모가 크다는 점이 문제가 되고있는 것이다.

따라서 현재 학생들에게 쉽게 정보에 접근이 가능한 챗봇이 필요하다.

- 졸업작품의 개발 목적

	의뢰 비용(천원)	월 지속 비용(천원)	비고
채널톡	문의 필요	70	한달 활성 사용자 20000명 이하
깃플챗	문의 필요	50	월 기본 요청 2000회 초과시 2000회당 50000추가
단비 AI	문의 필요	30	월 기본 제공량 대략 10000회 초과시 대략 1000회당 3000원 추가
Closer AI	0	75	
Cloud Turing	2,000	100	챗봇 대용량 서버 제공

한달 이용자 5천 이하, 트래픽 10000개 내외의 챗봇을 의뢰할 시 들어가는 비용은 위의 표와 같다.

챗봇 제작을 의뢰하면 초기비용은 물론이고 지속적으로 예산이 요구된다. 이것은 금전적으로 큰 부담이 된다. 또한 의뢰한 챗봇은 추가적인 업데이트에서 많은 제한을 가진다. 이 제한은 현재 Q&A 데이터가 적은 이유로 수집과 업데이트를 반복해야하는 울산대학교 챗봇 상황상 적절하지 않다. 따라서 챗봇을 의뢰하는 것 보다 자체적으로 개발을 하는 것이 적절하다.

학교에서 챗봇 서비스 제공을 계획중이라 설명하였지만 아직 정확히 확정된 사항이 아니며 제공 일자도 알수가 없는 상황이라 현재 불편함을 느끼고 있는 학생들을 위해 선발적으로 챗봇을 개발할 필요가 있다.

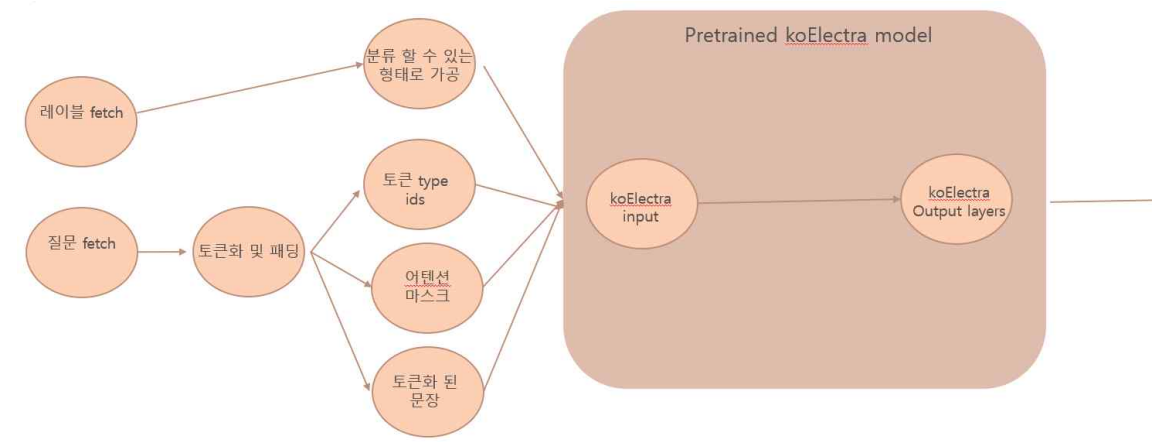
위에서 설명된 사항들을 해결 또는 완화 하기 위하여 개발 기획을 하였다.

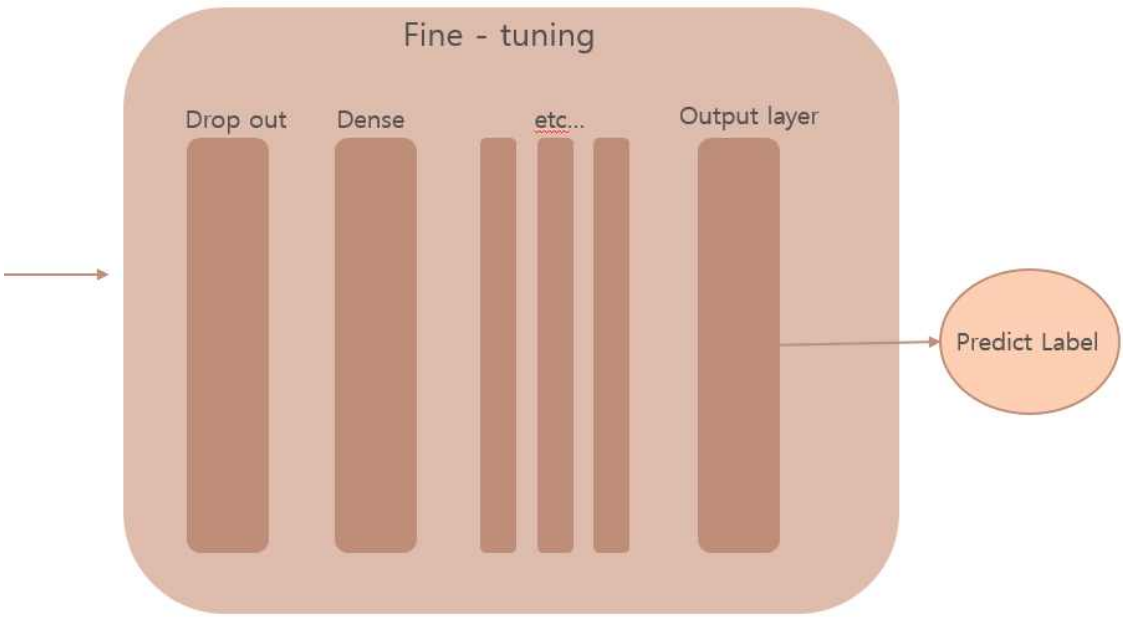
4. 졸업작품 내용

4-1. 시스템 구성

1) 소프트웨어 구성(구성도 포함)

A) AI 모델 구성도





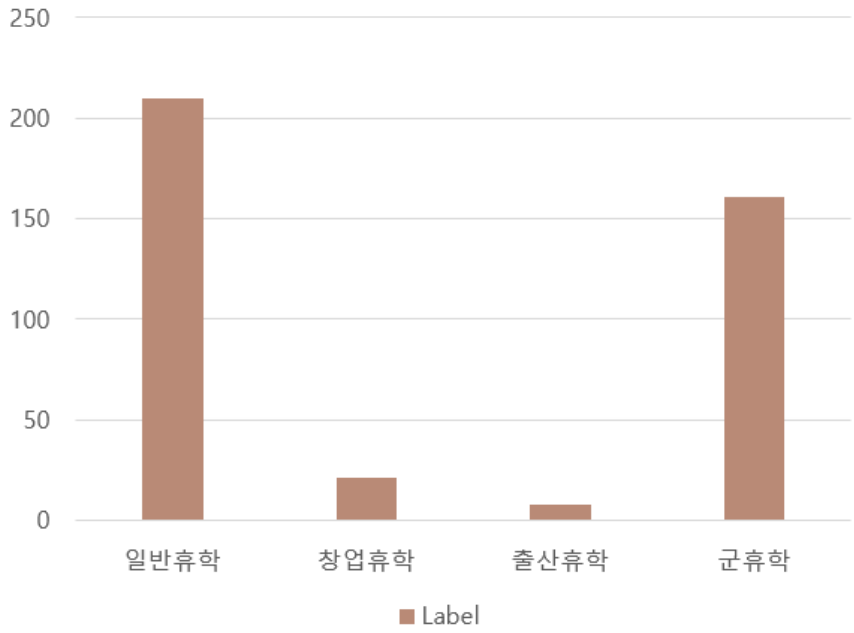
위의 그림은 챗봇 모델 구성도이다.

Pretrained koElectra는 이미 훈련된 koElectra의 discriminator 부분이다. 여기서 4개의 Dense layer을 추가하여 fine tuning을 하였고 3개의 Drop out을 추가하여 over fitting을 방지하였다.

다음으로 AI 모델을 제작할 때 마주한 가장 큰 문제 두가지를 소개하고 어떻게 극복 했는지 설명

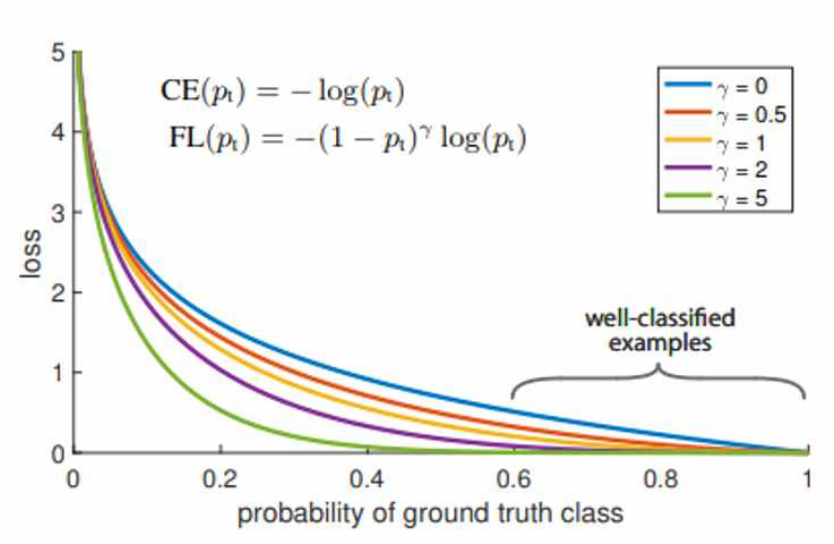
한다.

첫 번째로 Imbalanced Data(데이터 불균형)문제이다



위의 그림처럼 label당 학생 질문 빈도수가 다르기에 학습 데이터끼리의 label 불균형 문제가 심각하다.

이 문제는 기존 모델처럼 CrossEntropy Loss를 사용하여 훈련 시켰을때 개수가 지나치게 적은 label은 훈련이 덜 되어 최종 AI 모델의 정확도가 낮아지는 결과가 초래 된다.



Imbalanced 문제는 Cross Entropy Loss를 사용하는 대신 Focal Loss를 구현하여 사용하는 것으로 극복하였다.

Focal Loss는 기존 Cross Entropy Loss를 베이스로 하여 데이터의 수가 적은 label에 더 많은 가중치를 부여하여 loss값을 더욱 빨리 낮춘다.

	Cross Entropy Loss	Focal Loss
전체 테스트 정확도	86%	92%
소수 데이터 정확도	73%	92%

그리하여 Cross Entropy Loss를 사용한 model과 Focal Loss를 사용한 model사이에서 데이터 수가 적은 label의 테스트 정확도 차이가 유의미함을 발견하였다.

두 번째로 절대적인 학습 데이터의 부족이다.

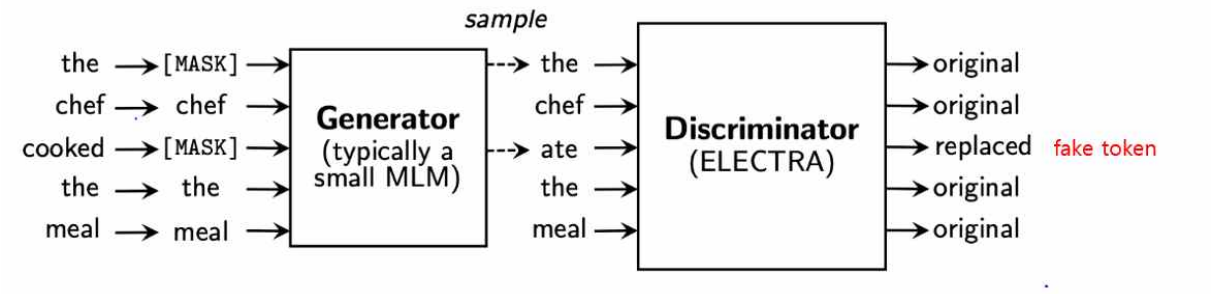
Uwins의 Q&A데이터들을 모두 끌어모아야 충분한 상황에서 다양한 제한사항으로 폐기되어지는 다수의 데이터로 인하여 학습 데이터 수의 부족 현상을 겪었다.

이 문제는 papago와 sbert의 문장간 코사인 유사도 측정 AI 모델을 이용하여 Data augmentation을 시도하여 해결하였다.

상세 과정은 먼저 노이즈를 주고 싶은 문장을 영어(또는 중국어, 일본어 등등)을 번역하고 또다시 한국어로 번역한다. 이 번역한 문장은 sbert의 문장 유사도 측정을 하여 특정 수치를 넘으면 전체 label 비율대로 추가해주었다.

이 방법으로 부족한 원본 데이터에 노이즈를 준 데이터를 augmentation하였다.

B) koElectra 구성도



크게 두 개의 신경망 Generator와 Discriminator로 구성된다.

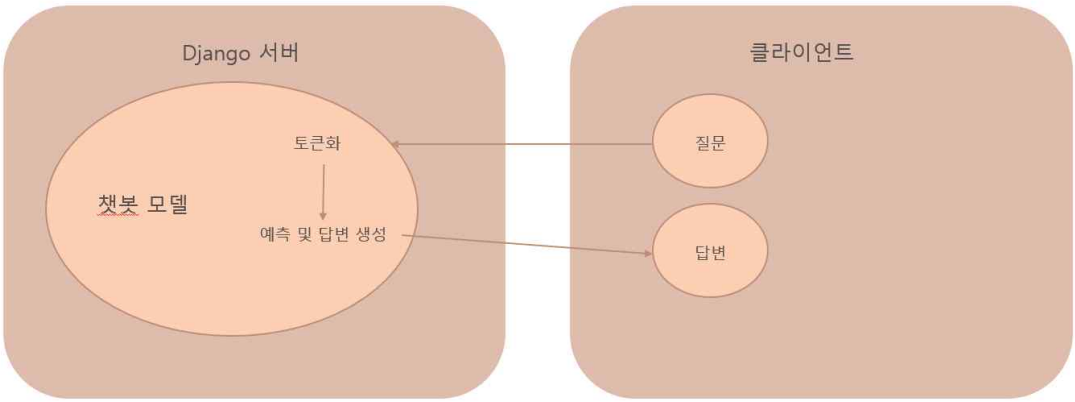
Generator는 MLM 방식으로 학습한다.

Generator는 입력 토큰 중 랜덤으로 [MASK]를 씌워서 전달된다.

Generator는 [MASK]를 내부에서 추론한다.

Discriminator는 Generator가 생성한 단어를 포함하여 각 단어가 원본 문장과 맞는지 아닌지 이진 분류하는 형태로 학습한다.

C) 웹서버 구성도



위의 그림은 Django 웹 서비스의 구성도이다.
Django 서버 안에는 챗봇 모델이 심어져있다.
클라이언트는 서버에게 쿼리를 던져주면 서버는 계산을 하여 클라이언트에게 답변을 전해준다.

4-2. 기능

현재 구현된 딥러닝 모델은 휴복학 Q&A와 등록 Q&A가 있다.
각 모델은 동일한 구조를 가지고 있으며 다른 train data로 학습되었다.
먼저 휴복학 Q&A는 1. 군복학 2. 군휴학 3. 창업휴학 4. 출산휴학 5. 선복학 6. 군휴학 연장 7. 질병휴학 8. 일반휴학 9. 일반복학 의 answer들을 지원한다.
등록 Q&A는 1. 등록금 납부 이월 2. 일반복학 3. 등록금 분할 납부 4. 장학금 5. 수업 연한 초과자 6. 졸업 유예자 7. 등록금 미납 8. 등록금 환불 및 반환 9. 일반 등록 의 answer들을 지원한다.
기타 기능으로는 원활한 유지보수를 위한 데이터 수집 기능이 있다. 이 기능은 시스템이 원하는 답변을 주었을 경우와 주지 못했을 경우 두가지로 나뉜다.
첫 번째로 시스템이 원하는 답변을 사용자에게 주었을 경우 시스템은 사용자가 했던 질의와 시스템이 분류했던 레이블을 데이터 베이스에 저장한다. 데이터 베이스에 추가 데이터가 일정 수 이상 채워졌을 때 시스템은 추가 데이터를 포함한 모델을 제작한다.
두 번째로 시스템이 원하는 답변을 사용자에게 주지 못했을 경우 시스템은 사용자가 했던 질의만 데이터 베이스에 저장한다. 이 데이터를 확보하는 이유는 후에 실무자가 따로 확인하여 적절한 레이블의 훈련 데이터에 추가 시키기 위함이다. 만약 기존의 레이블에서 찾아 볼 수 없는 새로운 유형의 질문이라면 실무자는 새로운 레이블을 추가 시킨다.

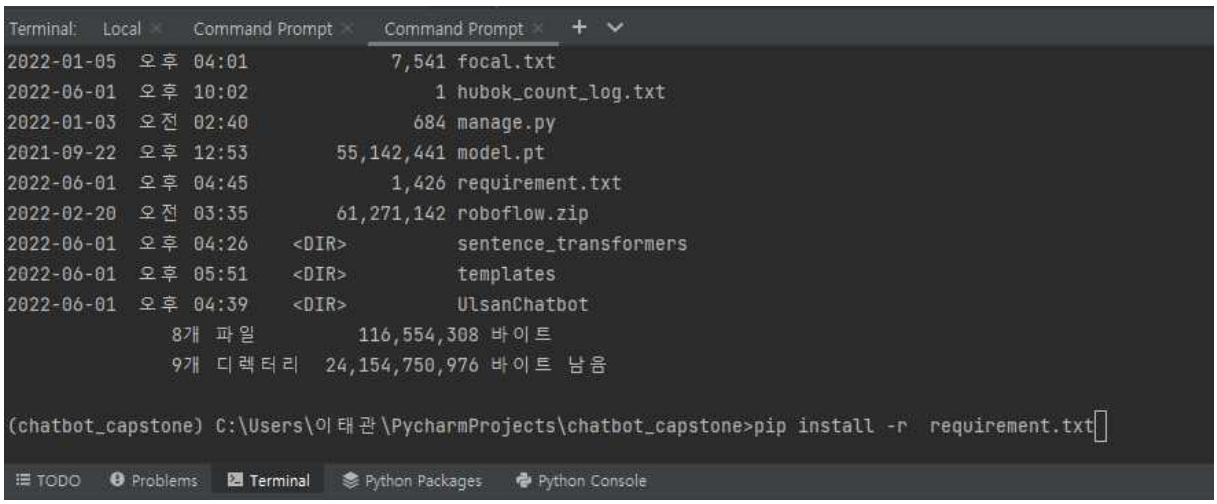
4-3 사용법

/* 설치 방법 포함 */

1. python (pycharm)과 annaconda 설치

2. 터미널에

>> pip install -r requirement.txt



-> 오류 가 날 경우 수동으로 하나하나 설치한다.

>> pip install django

>> pip install transformers

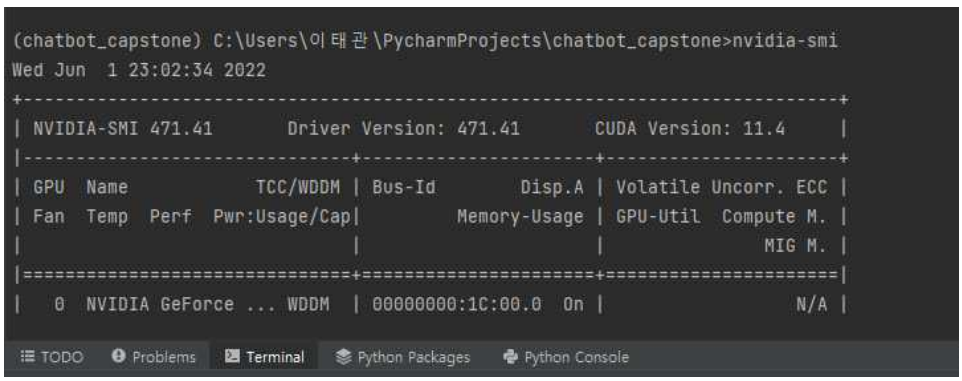
>> pip install numpy

>> pip install pandas

>>pip install IPython

>> pip install tqdm

>> nvidia-smi



pytorch 공식 홈페이지에서 CUDA version과 최대한 맞는 버전의 pytorch를 설치

>> conda install pytorch torchvision torchaudio
cudatoolkit=11.3 -c pytorch

만약 GPU를 쓰기 싫다면 CPU 버전을 설치해도 실행가능 (속도가 느려짐)

3. 프로젝트 파일 홈에서 장고를 실행

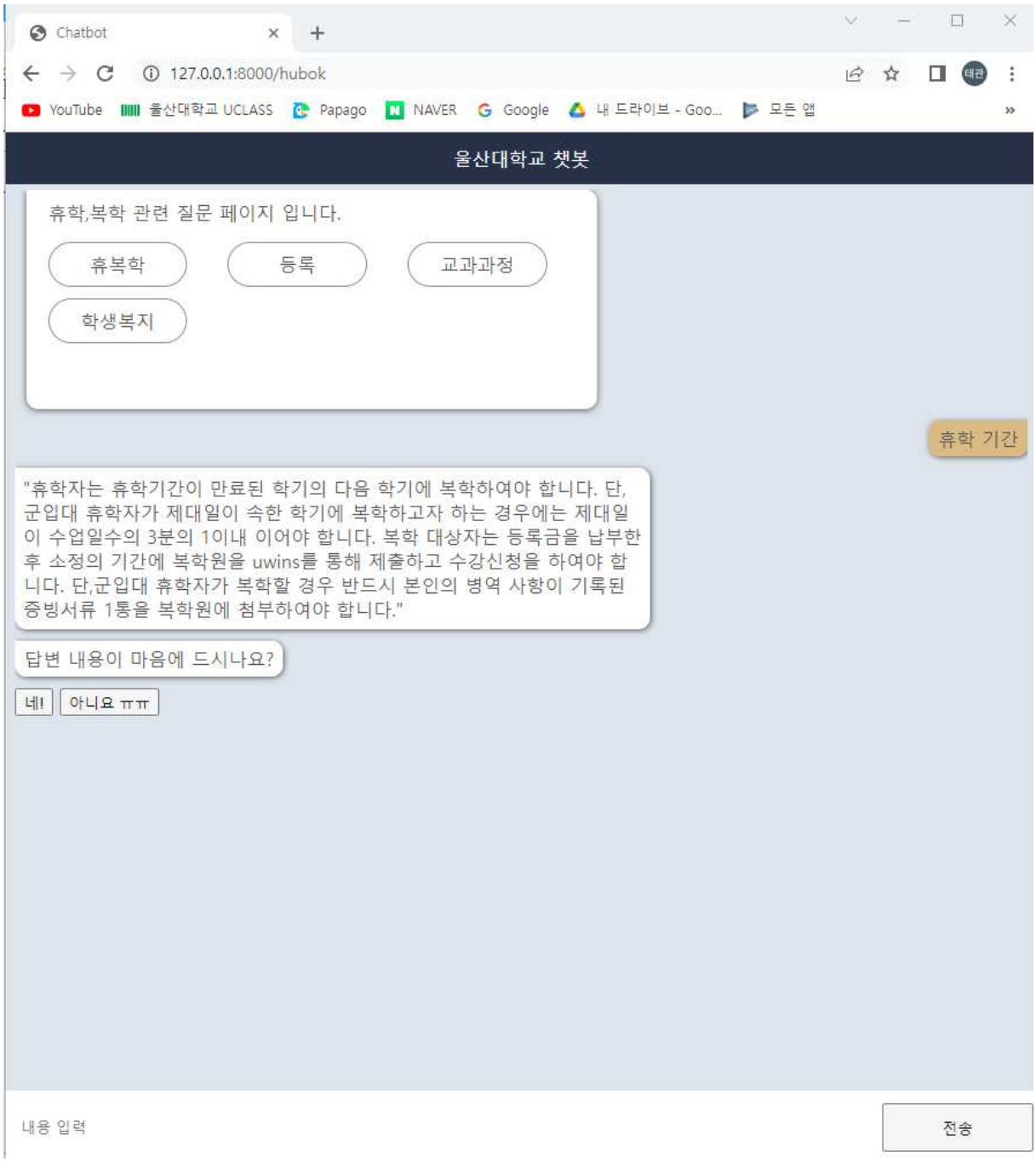
>> python manage.py runserver

```
(chatbot_capstone) C:\Users\이태관\PycharmProjects\chatbot_capstone>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

4. http://127.0.0.1:8000/ 에 접속

```
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

5. 하단의 내용입력에서 질문을 입력



4-4 개발환경

/* 언어, 설계/구현 도구, 사용시스템 등 */

운영체제 : Windows 10

주 언어 : Python 3.7.0

IDE : Pycharm - professional

구현 도구 : koElectra

사용 라이브러리 : numpy
Django 3.2.10
Pandas
Huggingface-hub 0.4.0
pyTorch 1.10.1
Transformers 4.10.2
Selenium
Beautifulsoup
IPython

개발 하드웨어 환경 : CPU - AMD Ryzen 5 2600 Six-Core Processor
GPU - NVIDIA GeForce GTX 1660
메모리 - 16GB
하드디스크 - Samsung SSD 970 PRO 512GB

4-5 졸업작품 설명

1) 파일/모듈 구성

UlsanChatbot.dataloader.chatbot.ChatbotTextClassificationDataset

UlsanChatbot.model.koelectra.ElectraClassificationHead

UlsanChatbot.model.koelectra.koElectraForSequenceClassification

UlsanChatbot.model.koelectra.Focal_loss.focal_loss

UlsanChatbot.model.koelectra.Focal_loss.label_to_one_hot_label

UlsanChatbot.model.koelectra.koelectra_input

UlsanChatbot.train.run_koelectra.train

UlsanChatbot.train.run_koelectra

Chatbot.views

Chatbot.views.addFineTune

Delok_count_log.txt

Hubok_count_log.txt

2) 함수/모듈 별 기능

UlsanChatbot.dataloader.chatbot.ChatbotTextClassificationDataset

- dataset을 훈련하기전 적절한 형태로 가공

UlsanChatbot.model.koelectra.ElectraClassificationHead

- fine tuning할 신경망의 layer 정의

UlsanChatbot.model.koelectra.koElectraForSequenceClassification

- pretrain 된 electra 모델을 정의한 후 훈련할 때 순전파의 동작을 정의

UlsanChatbot.model.koelectra.Focal_loss.focal_loss

- focal loss 구현

UlsanChatbot.model.koelectra.Focal_loss.label_to_one_hot_label

- focal_loss 함수에 넣기전 label들을 원핫인코딩 하는 기능

UlsanChatbot.model.koelectra.koelectra_input

- 훈련을 마친 모델이 실제 predict 작업을 수행할 때 input 문장을 적절한 형태로 바꿔주는 기능

UlsanChatbot.train.run_koelectra.train

- 훈련 할때 1epoch 당 수행할 행동 정의

UlsanChatbot.train.run_koelectra.__main__

- 훈련

Chatbot.views

- Django 웹서버에서 실행 될 메소드들을 정의

Chatbot.views.addFineTune

- Django 웹서버에서 축적된 질의 데이터들이 일정 수가 넘었을때 실행 되는 추가 훈련 메소드

Delok_count_log.txt

- Django 웹서버에서 축적된 질의 데이터가 몇 개인지 알 수 있는 로그 데이터

Hubok_count_log.txt

- Django 웹서버에서 축적된 질의 데이터가 몇 개인지 알 수 있는 로그 데이터

3) 데이터베이스 구조 또는 자료구조

1. 훈련 및 테스트 데이터 구조

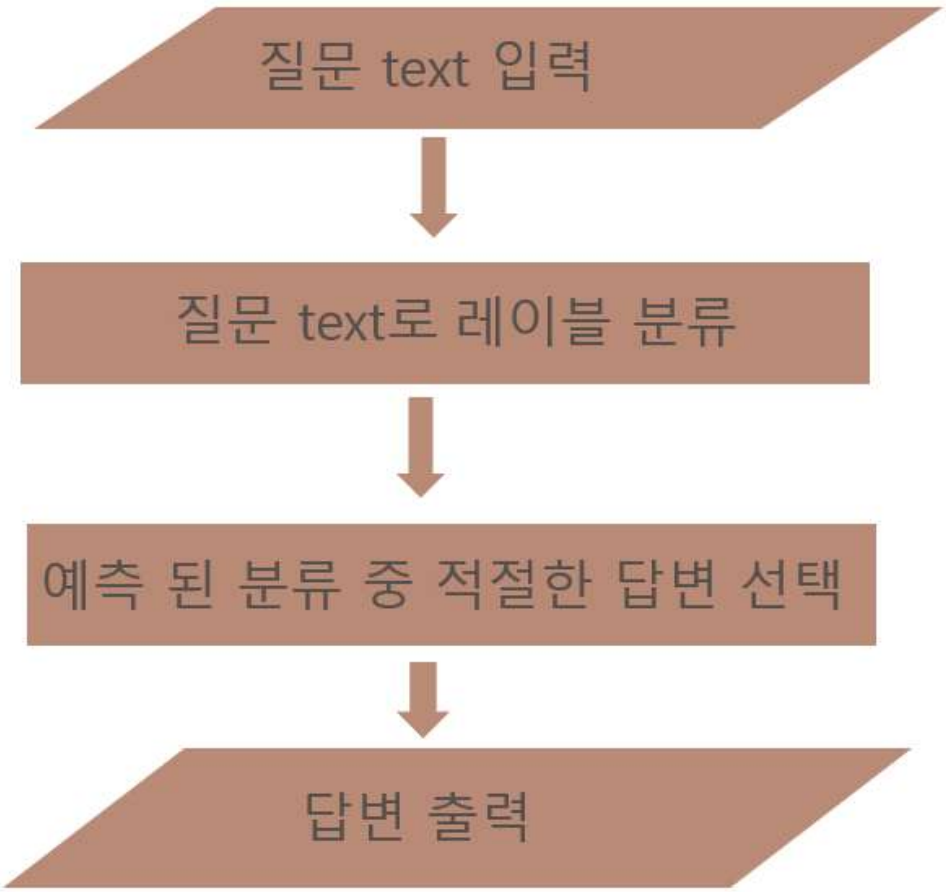
- 3개의 column으로 구성

- 첫 번째 column은 레이블
- 두 번째 column은 질문 데이터
- 세 번째 column은 답변 데이터

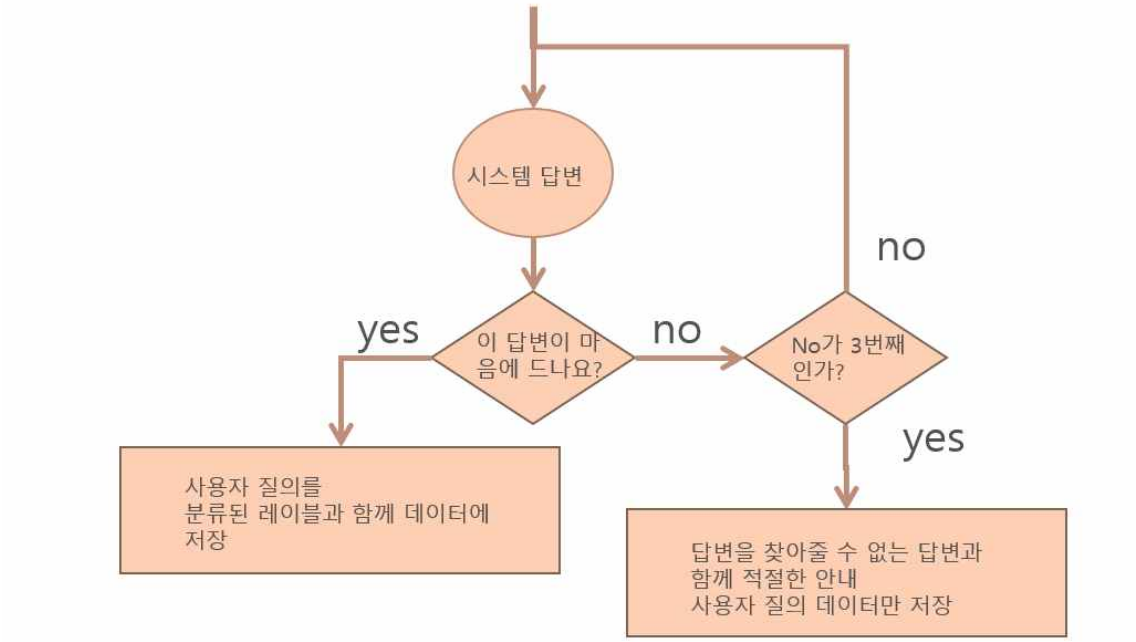
2. 웹서버가 softmax tensor를 보관하는 자료구조
- priority queue 사용
- 사용자에게 원하는 답변을 제공해주지 못했을 경우에 O(logn)의 속도로 다음으로 정확한 답변을 제공해주기 위함

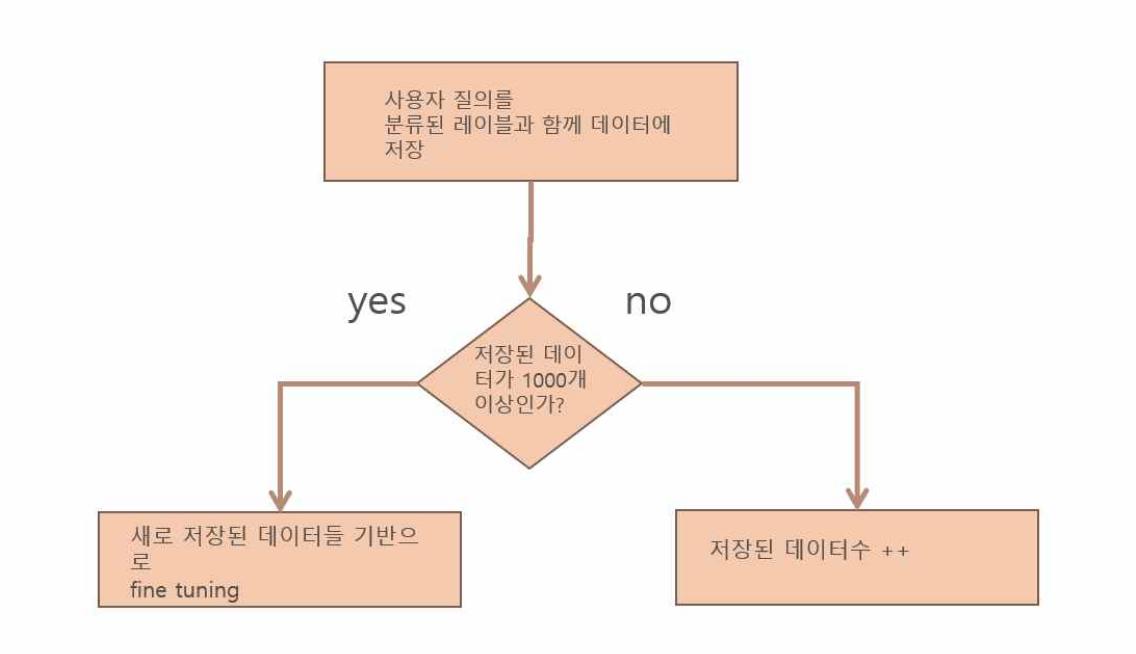
4) 주요 기능에 대한 흐름도

1. 딥러닝 모델 작동 흐름도



2. 데이터 유지보수


시스템 답변
이 답변이 마음에 드나요?
yes
no
No가 3번째 인가?
no
yes
사용자 질의를 분류된 레이블과 함께 데이터에 저장
답변을 찾아줄 수 없는 답변과 함께 적절한 안내
사용자 질의 데이터만 저장


사용자 질의를 분류된 레이블과 함께 데이터에 저장
yes
no
저장된 데이터가 1000개 이상인가?
새로 저장된 데이터들 기반으로 fine tuning
저장된 데이터수 ++

5. 제작 일정

MileStone	배정인원	21. 06	21. 07	21. 08	21. 09	21. 10	21. 11	21. 12	22. 01	22. 02	22. 03	22. 04	22. 05
분석													
아이디어 회의	이태관, 백동수												
수요 조사	이태관, 백동수												
현재기술 조사	이태관												
필요지식 스터디	이태관, 백동수												
설계													
개발 모델 결정	이태관, 백동수												
데이터 확보	백동수												
프론트엔드 제작	이태관												
백엔드 제작	이태관												
AI 모델 설계	이태관, 백동수												
추가 기능 논의	이태관, 백동수												
테스트, 디버그	이태관												
추가 기능 논의	이태관, 백동수												

6. 향후 연구와 기대효과

개발된 챗봇은 울산대학교 사이트에 흩어져있는 수많은 정보들을 한 곳에 규합하고 분류하여 학생들의 질문 해결 소요시간을 크게 단축시킨다. 그리고 문의를 받는 교직원들의 근무환경을 크게 좋게 변화시켜 업무 효율 증가를 기대한다.

본 프로젝트로 제작된 koElectra 기반 울산대학교 챗봇은 다음의 이슈가 있다.
첫 번째로 '바나나 먹고싶다'로 훈련된 모델은 '바나나 먹-' 형태의 문장은 올바르게 분류되나 '먹- 바나나' 형태의 문장은 '바나나 먹-' 형태의 문장과 동일한 의미를 가지고 있음에도 불구하고 다르게 분류되는 현상이 있다. 실제 문장의 의미와 훈련 데이터의 단어순서에 따라 달라지는 시스템이 받아 들이는 의미 사이의 괴리를 줄이는 것 대한 연구가 진행될 예정이다. 이 연구는 챗봇 모델의 분류 정확도를 향상 시키는 효과를 기대한다.

첨부: 졸업작품 CD

/* Label에는 다음 사항을 표시

2014-1

졸업작품명: O O O

제출자: OOO, OOO, ...

CD에는 다음 파일들을 포함

- 설치/실행 파일
- 설치매뉴얼
- 졸업작품보고서 등 */