개발 문서

팀명 : 조정박씨

조용범, 정석현, 박재연

프로젝트 제목 : 너두야

프로젝트 설명 : 너두야 프로젝트는 영단어 학습 Application 개발 프로젝트입니다. 사용자 단어장의 단어들이 실제로 어떻게 사용되는지 알려주고, 단어의 뜻과 문맥적 의미를 학습시키고자 방대한 양의 컨텐츠(미드, 영드 및 영화)에서 해당 단어들이 쓰이는 예문들을 검색한 뒤, 해당 예문들을 묶어 Word playlist로 제공합니다.

-총 3명이 데이터 전처리, REST API 구현, CLIENT 개발 파트를 나눠서 구현하였습니다. 3명이 각자 프로젝트를 진행해서 루트 디렉토리 내부의 각 3개가 하나의 프로젝트를 담당합니다.

-데이터 전처리 파트

영상을 전처리하는 과정은 우리가 관심이 있는 단어가 있는 영상 클립과 그것에 관한 정보를 추출하는 것을 목표로 합니다. 해당 과정을 위해서는 한글+영어 통합 자막이 필요하며 해당 자막이 없을 경우에는 네이버 AI API를 통해 생성하게 됩니다.

전처리 파트는 디렉토리 내의 Processing 폴더 내부에 있습니다

-사용 데이터 목록

빅뱅이론, 프렌즈, 실리콘밸리 미국드라마 파일을 이용했습니다. 단어는 파고다토익 토익 필수단어 중 20여개를 선정하였습니다.

업로드시 영상 데이터의 크기를 고려해 원본 영상 파일은 업로드하지 않았고, 나눈 클립 파일만 업로드하였습니다.

-한글+영어 자막이 있을 경우

1. 자막 인덱싱

SMI 자막파일을 분석해서 자막에 등장하는 모든 단어를 인덱싱합니다. 이 인덱싱 파일은 어떤 단어가 나오는지, 해당 단어가 나오는 문장과 문장의 등장 시간 등의 정보를 가지고 있습니다.

전처리 디렉토리에서 bigbang.csv 등의 파일이 해당 인덱싱 파일입니다. 이 파일은 bigbang 디렉토리에 있는 모든 영상에 대한 정보를 가지고 있습니다.

Main 파일에서 주석되어있는 make\_indexing 메서드를 통해 인덱싱을 수행할 수 있습니다. 이는 prepare.py에 구현되어 있습니다.

2. word list에 해당하는 단어 추출

해당 인덱싱 파일들을 분석해 우리가 관심있는 단어의 등장을 인덱싱 파일을 통해 검색합니다. 이 결과를 csv파일로 출력합니다. 자막 인덱싱 파일 정보와 추가적으로 어떤 영상에서 나왔는지에 대해 정보를 담고 있습니다.

또한 이 출력 파일은 자동으로 추출한 해당 문장 전후 문맥에 대한 정보를 포함합니다. 사용자는 학습에 도움이 되게 전후 문맥을 어디까지 포함할 지, 이 예문을 포함할 지 등에 대한 정보를 수정합니다. 이 수정 파일을 토대로 비디오 클립을 생성합니다.

수정하기 전 정보는 wordlist\_raw.csv, 수정한 후 정보는 wordlist\_mod.csv에 저장되어 있습니다.

Main.py의 make\_word\_list 메서드를 통해 실행 가능합니다.

3. video clip 생성

해당 정보를 바탕으로 비디오 클립을 생성합니다. 해당 클립 생성 후 클립들에 관한 정보를 clips/clips.db 에 저장하게 됩니다. 이 db를 바탕으로 클라이언트에서 REST API에 영상이나 정보를 요청합니다.

-영어 자막만 있을 경우

영어 자막을 Papago 번역 API를 통해 번역합니다. 이는 translate.py를 실행하면 수행됩니다. 또한 관련 함수는 naverapi.py에 정의되어 있습니다.

-영어 자막이 없을 경우

영어 자막을 clova stt api를 통해 생성합니다. 일단 해당 비디오를 60초 이내로 쪼갠 후, 음성을 추출합니다. 이 추출한 음성을 가지고 stt를 시행합니다. 추출한 텍스트를 합쳐 영어 자막을 생성합니다. 이 자막을 Papago 번역 api를 통해 번역합니다.

자막 추출은 nvaerapi.py를 실행하면 가능합니다. 이 stt데이터를 통해 자막을 만드는 것은 make\_subtitle.py를 실행하면 가능합니다.

-기타 구현

Smi parser와 smi 인코딩 등에 관한 수정에 필요한 함수도 구현하였습니다. Smi 관련 함수는 smipy.py에 구현되어 있습니다.

-REST API 제작 관련

REST API는 클라이언트 단에서 요청을 받아서 해당 클립의 정보와 해당 영상 클립을 반환하는 API입니다. Flask를 이용해서 구현하고 Google cloud platform을 이용해서 API 배포를 하였습니다. smi자막파일을 포함한 모든 데이터는 json으로 통신합니다.

1. 데이터(csv)를 클라이언트에서 요청한 형태로 서버에서 변경 (flask 사용, static local 위치 업로드)

- 영상 자막 파일(smi) 파일을 json 형태로 처리

- 서버 관리 - google cloud platform

1) 데이터(csv)를 클라이언트에서 요청한 형태로 서버에서 변경 (flask 사용, static local 위치 업로드)

처리상의 이점 때문에 해당 데이터를 json으로 변환합니다. 이 작업에는 pandas를 이용합니다

아래 그림 형태를 보면 CSV -> JSON 변환 방식과 데이터 처리 과정을 시각적으로 알 수 있다.

실행 방식

Go to hackerton repository

import Flask, request, jsonify, pandas, json, os

Run the app.py

You can get your local server, “ip address”:5000/getdata

You can get Ip address by ipconfig on Terminal

You can get clips\_db.json from csv file

2) 영상 자막 파일(smi) 파일을 json 형태로 처리

3) 서버 관리 - google cloud platform

static 서버에 들어가면 smi와 영상 파일에 접근 가능

-프론트엔드

1. 개발한 프레임 워크  
   프론트 엔드는 React Native와 Expo SDK를 이용하여 개발하였다. 가장 최상위 단에 WordCardList 라는 View를 만들고, 내부 View들은 function으로 구현하여 state의 단계를 최소화하고자 했다.
2. Client 실행방법  
   - Expo Client  
   1. smartphone에 Expo Client를 설치한다  
     
   - Working Directory  
   1. $ cd youtoo //youtoo directory로 이동  
   2. $ expo start //  
   3. 브라우저에 표시된 QR코드로 접속하여 Expo Client 를 실행한다.
3. 사용된 모듈
   1. Video : expo-av
   2. Autoscrolling : ‘react-native-autoscrolling’