|  |
| --- |
| 2021 전기 졸업과제 중간보고서 |
| 사투리 번역 웹/앱 개발 |



|  |  |
| --- | --- |
| 분과 | A조 |
| 팀명 | 내마리그마리다 |
| 팀원 | 201746108 김윤종 |
| 201710707 김도영 |
| 201801205 황승찬 |
| 지도 교수 | 이도훈 |

|  |
| --- |
| 목차 |

[I. 요구 조건 및 제약 사항 3](#_Toc110148259)

[1.1 요구 조건 3](#_Toc110148260)

[1.1.1 음성인식 API 3](#_Toc110148261)

[1.1.2 문장 번역 서비스 3](#_Toc110148262)

[1.1.3 웹-앱 서비스 4](#_Toc110148263)

[1.2 제약사항 4](#_Toc110148264)

[1.2.1 음성인식 API 4](#_Toc110148265)

[1.2.2 문장 번역 서비스 5](#_Toc110148266)

[1.2.3 웹-앱 서비스 7](#_Toc110148267)

[II. 설계 상세화 및 변경내역 7](#_Toc110148268)

[1.1 음성인식 API 7](#_Toc110148269)

[1.2 문장 번역 서비스 7](#_Toc110148270)

[1.3 웹-앱 서비스 9](#_Toc110148271)

[III. 과제 추진 계획 9](#_Toc110148272)

[IV. 구성원별 진척도 9](#_Toc110148273)

[V. 과제 수행 내용 및 중간 결과 9](#_Toc110148274)

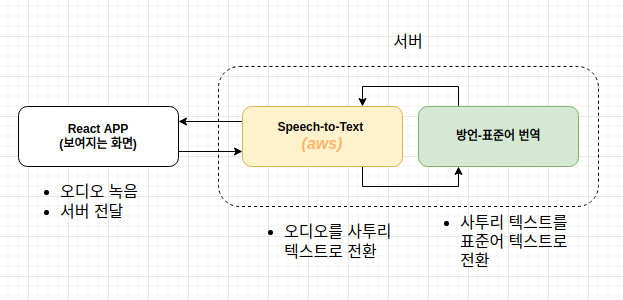
[1.1 음성인식 API 9](#_Toc110148275)

[1.2 문장 번역 서비스 13](#_Toc110148276)

[1.3 웹-앱 서비스 14](#_Toc110148277)

1. 요구 조건 및 제약 사항

## 요구 조건



본 과제는 사투리를 음성으로 받아 표준어로 번역하는 사투리 번역기 프로그램 개발을 목표로 한다. 팀원 3명이 각자 3파트로 나뉘어 진행하며, 웹-앱 서비스, 음성인식 API, 방언-표준어 번역 서비스의 3가지이다. 각 파트의 요구 조건은 다음과 같다.

### 음성인식 API

* 사투리 음성을 사투리 문장으로 반환

|  |
| --- |
| 스마트폰, PC 등 플랫폼에서 웹에 접속하여 업로드 한 사용자의 음성을 텍스트로 만드는 기능을 수행해야 한다. 음성을 녹음하는 것은 웹-앱 파트에서 하지만 오디오 형식을 맞춰 데이터를 주고받아야 하고, STT API에 넘겨 방언 텍스트를 만든 다음 방언-표준어 번역 프로그램에 넘겨야 한다. 즉 다른 파트들의 중간 매개체 파트이기도 하다. |

* 웹-앱, 방언-사투리 번역 프로그램과 연결

|  |
| --- |
| 웹-앱과의 연결은 서버 통신으로 이루어지고, 방언 번역 서비스 간의 연결은 같은 서버 PC 내에 있을 것이므로 코드를 합치거나 파일 위치를 공유하여 연결한다. |

* 서버에 구축 필요

|  |
| --- |
| 과제의 목표는 시스템을 서버에 구축하는 것이다. 외부에서 상시 접속이 가능해야 하며, 앱을 연결 시에도 다량의 데이터를 다룰 수 있도록 음성인식 API와 번역 서비스는 서버에 올려들 필요가 있다. |

### 문장 번역 서비스

* 사투리 문장-표준어 문장으로 반환

사투리 음성에서 텍스트로 반환 문장을 다시 표준어로 반환하는 과정을 거친다. 해당 과정에서 최근 번역에서 좋은 성능을 보여주는 딥러닝 기반의 번역을 진행하려고 한다.

### 웹-앱 서비스

* 다양한 형태의 웹

|  |
| --- |
| 스마트폰의 등장으로 언제 어디서나 정보로의 접근을 가능하게 만들었다. PC뿐만 아니라 스마트폰, 태블릿에서도 웹에 접속할 수 있는 IT환경이 만들어졌다. 웹에 접근할 수 있는 디바이스의 폭이 넓어진 만큼, 각각의 디바이스에 적합한 해상도와 레이아웃을 일일이 대응해야 한다. |

* 웹 배포의 지속적인 관리

|  |
| --- |
| 기능을 수행하는 웹을 배포하는 것 뿐만이 아니라, 유지보수의 지속관리가 필요하다, |

* iOS/Android 앱 개발

|  |
| --- |
| 연령대, 성별, 직장별로 다양한 스마트폰을 사용한다. 운영체제는 크게 두 가지로 나뉜다. 애플의 iOS를 사용하는 아이폰과, 구글Android를 사용하는 갤럭시 등의 나머진 스마트폰이다. 겉으로 보기에는 큰 차이가 없으나, 구조상 차이와 지원하는 언어와 기능들이 다르기 때문에, 각자 개발하려면 수많은 인력과 시간이 필요하다. |

## 제약사항

### 음성인식 API

* Open API 사용이 지배적

|  |
| --- |
| 음성을 국제 음성 기호(음소)로 변환하는 API는 현재 없고, 음성인식 API의 원리 상 음소를 정확히 찾아내는 것이 아닌 비슷한 파형의 음성 배열에서 등장할 확률이 높은 단어를 통해 단어 배열을 결정하는 것이기 때문에 정확한 음소를 찾는 것이 크게 의미가 없다. 현재는 가장 활용성이 open API를 사용해 구현하는 방법을 선택했다. |

* 텍스트이(가) 표시된 사진

  자동 생성된 설명구글 개인용 API의 시간 제약

|  |
| --- |
| 구글의 STT API는 한 달 60분 이상의 오디오 파일 사용 시 비용이 청구된다. |

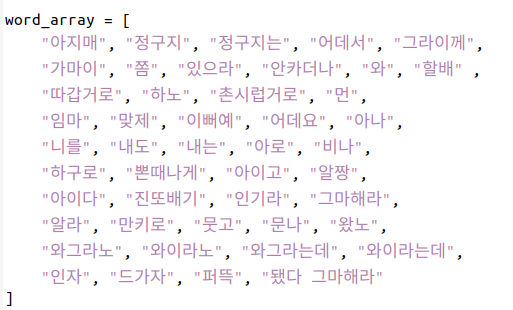
텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| 서버에 올려 계속 사용할 시 어떤 이유에서 인지 개인 계정에서는 서비스 이용 위반이 된다. API 키 파일을 서버 PC에 암호화 없이 그대로 올려둔 것과 사용자 PC 이외의 PC로 접근 등 생각해볼 수 있는 이유는 많지만 해결 방법은 아직 찾지 못했다.  위의 두가지 내용은 음성인식 api를 서버에 올려 지속적으로 활용하는 데 큰 제약 사항이 된다. 비용 자체는 비싸지 않고, 60분을 넘기는 번역을 사용할 일도 크게 없다. |

* Speech Context(사용자 지정 단어) 5000 단어(문장) 제약

|  |
| --- |
| 이 제약 사항은 방언 음성인식 API의 구현 성능에 관여하는 가장 큰 문제점이다. 표준어 기반의 API를 방언을 인식하도록 바꾸려면 상당한 양의 방언 데이터를 추가하게 되는데, 구글 API는 5000단어라는 다소 부족한 수준의 추가 단어만 지원한다. 우리말 샘 방언 사전에 실린 경상도 명사만 20000개가 넘는 것을 생각하면 큰 제약 사항이다.  지역별 방언 세트를 나누어 지원하고, 자주 쓰이지 않는 단어를 제외, 수작업으로 데이터 셋을 다시 짜는 것으로 계획하고 있다. |



|  |
| --- |
| 사투리 단어는 배열 데이터 형식으로, 구글 API 내부 함수에 인자로 들어간다. |

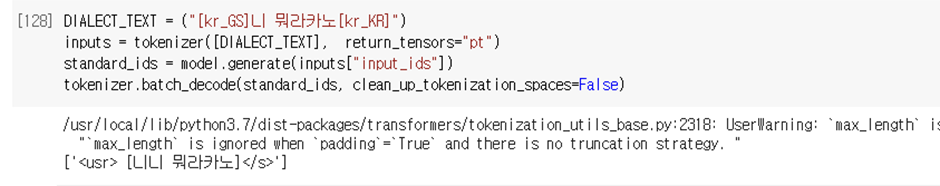
### 문장 번역 서비스

* 학습 데이터 셋의 부족



* 좌(방언), 우(표준어)

|  |
| --- |
| 방언 문장과 표준어 문장은 ai hub에서 쉽게 구할 수 있다. 하지만 해당 데이터 셋은 발화 데이터 셋으로 일상생활에서의 대화를 수작업으로 옮긴 데이터셋으로 두 가지의 문제점이 존재한다.   1. 위의 표에 빨간색으로 나타낸 부분은 문장에서 전혀 필요 없거나 표준어로 라벨링된 데이터셋에 방언이 그대로 들어가 있다. 2. 위의 표는 총 21개의 문장이다. 하지만 해당 데이터셋에서 방언으로 번역된 것은 주황색으로 표시된 단어 단 1개다.   ai hub는 총 32만개가 넘는 데이터셋을 가지고 있지만 이런 데이터 셋으로는 제대로 된 모델의 학습이 어렵다. 우리는 방언을 표준어로 변환하는 것을 목표로 잡고 있기 때문에 방언에서 표준어로 바뀌는 데이터셋이 어느 정도 필요하다고 판단하였다.  -사전학습 모델에 있어서 방언 적용의 제약  토크나이저는 딥러닝 기반의 모델이 텍스트를 이해할 수 있게 숫자로 나눠주는 도구를 말한다.  ex) 나는 밥을 먹었다 🡪 나는/밥을/먹었다 🡪 1234/1323/3454  토크나이저는 모델의 학습을 위해 모델의 학습 이전에 미리 우리가 사용할 데이터셋을 통해 학습을 시켜 해당 데이터셋에 대한 단어들을 숫자로 미리 바꿀 수 있게 된다. 이런 토크나이저가 미리 학습을 한 상태에서 우리는 해당 토크나이저가 인식할 수 있는 단어를 모아둔 것을 '단어사전(Vocabulary)라고 하고 만약 이런 단어사전에 없는 단어 즉 토크나이저가 인식할 수 없는 단어들을 OOV(Out Of Vocabulary)라고 한다.  딥러닝 기반 번역에서 가장 좋은 성능을 내는 모델은 현재 사전학습 모델이며 대표적인 예로 T5, BART등을 들 수 있다. 하지만 사전학습 모델은 대량의 데이터셋을 미리 학습시킨 모델로 이미 토크나이저가 지정되어 있으며 우리들이 원하는 데이터셋에 해당 토크나이저를 학습시킬 수 없다. 사전학습 모델은 표준어기반의 kowiki를 통해 사전학습을 진행하여 많은 방언 단어에 대해 OOV 문제를 일으키고 이는 큰 학습 저하로 이어졌다. |



|  |
| --- |
| 위의 그림은 SKT에서 제공하는 KoBART를 통해 기계번역을 시도한 것이다. 하지만 뭐라카노 같은 단어는 단어사전에 존재하지 않아 제대로 학습이 되지 않은 모습을 볼 수 있다. |

### 웹-앱 서비스

* 다양한 형태의 웹

|  |
| --- |
| 기존 데스크탑 뿐만이 아니라, 스마트폰의 인터넷의 사용이 오히려 더 많은 오늘날, 웹 페이지는 여러 크기와 형태의 모바일 웹까지 지원해야 한다. 다양한 디바이스와 다양한 크기의 화면에서 일관되게 잘 작동하는, ‘반응형 웹 페이지’ 를 제작하려고 한다. |

* 웹 배포의 지속적인 관리

|  |
| --- |
| 도커 등과 같은 툴을 사용하여 유지보수와 기능 추가 등에 용이하게 하려고 한다. 추후 필요성이 느껴지면, 또 다른 배포 관리 툴을 추가할 예정이다. |

* React-Native로 iOS/Android 앱 개발

|  |
| --- |
| 리액트 네이티브는 iOS와 안드로이드에서 모두 동작하는 네이티브 모바일 앱을 만드는 자바스크립트 기반 프레임워크 이다. 네이티브 브릿지를 통해 네이티브 스레드(Native Thread)와 통신하면서 웹앱과 다르게 성능을 최적화한다. 이렇게 웹을 사용하지 않고 네이티브와 통신하는 방법을 사용하는 것을 하이브리드 앱(Hybrid App)이라고 한다. |

1. 설계 상세화 및 변경내역

## 음성인식 API

* 구글 STT API, 네이버 CLOVA CRS, 카카오 Speech API 비교 분석

음성인식을 위한 적합한 Open API를 선정

* 구글 STT API의 speech context 활용

음성을 방언 텍스트로 반환하기 위한 자체 단어 지원 기능 활용

* 음성-텍스트 변환 API 서버 구축

상시 사용 가능하고 사용자의 기기와 분리된 시스템 서버 구축

* 웹/앱 오디오 녹음 및 서버 연결

서버와 분리된 웹/앱과 서버를 연결하는 중간 매개 역할 개발

## 문장 번역 서비스

* 기존 데이터셋 정제 및 영화 대사 추가

우리가 필요한 데이터셋은 제대로 번역된 방언에서 표준어로의 변환된 데이터 셋이었다.

사투리가 심한 영화들(바람, 범죄와의 전쟁)의 한글 자막을 가져온 후 이를 직접 표준어 텍스트로 바꾸는 작업을 진행하였다.

with open("/content/drive/MyDrive/Data/movie\_raw\_text/bum.txt","r",encoding='UTF8') as f:

    strings = f.readlines()

    print(strings)

raw\_text\_list = []

strings = "".join(strings)

strings = strings.split('<SYNC Start')

import re

result = []

sejongking = re.compile('[^ ㄱ-ㅣ가-힣]+')

for string in strings:

  if len(sejongking.sub('',string).strip()) > 3:

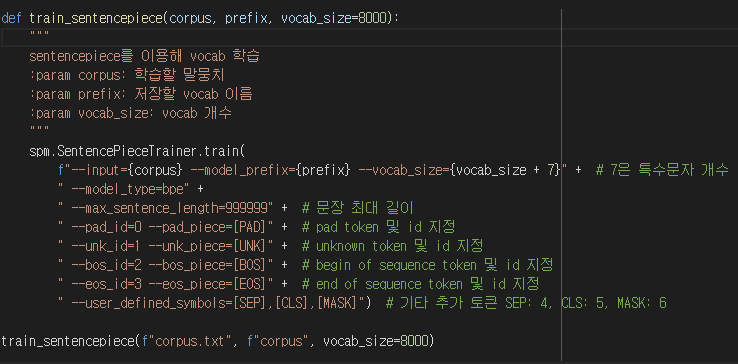
    result.append(sejongking.sub('',string).strip())



위와 같은 문장 약 5천개를 수작업으로 표준어로 바꾸는 작업을 진행하였다. 또한 aihub 데이터셋 총 5만개를 정제하는 작업을 진행 중이다.

* 사전학습 모델이 아닌 transformer만으로 성능을 최대화

transformer는 사전학습 모델 이전의 번역에 있어서 가장 성능이 좋은 모델이다. 또한 우리가 지정한 토크나이저가 사용이 가능하기 때문에 transformer를 학습 모델로 지정하여 사용하였다. 토크나이저는 방언 특성상 단어안의 조사만으로도 될 수 있기 때문에 word보다 작은 단위인 subword로 토크나이징하는 wordpiece모델인 sentencepiece를 사용하였다.



## 웹-앱 서비스

* 웹서버 배포를 위한 react와 nginx 사용

nginx는 웹 서버 소프트웨어로, 가벼움과 높은 성능을 목표로 한다. Nginx는 요청에 응답하기 위해 비동기 이벤트 기반 구조를 가진다. 이러한 구조는 서버에 많은 부하가 생길 경우의 성능을 예측하기 쉽게 해준다. nginx의 핵심은 **가벼움과 높은 성능**이다. Apache에 비하여 스레드를 많이 사용하지 않기 때문에 CPU 소모도가 상대적으로 낮다. 적은 수의 스레드로 효율적인 일처리가 가능하기에 많은 접속자들이 와도 대응을 할 수 있다.

* docker를 통한 웹 배포관리

컨테이너는 애플리케이션을 환경에 상관없이 실행하는 기술이다. 예를 들어, 우분투 OS를 설치하려 해도 도커는 명령어로 실행하면 어느 환경이든 상관 없이 다음 명령어를 사용하여 실행할 수 있다. 서버 제작과정에 견고함과 유연성, 편리함이 좋고 다른 사람이 만든 서버를 소프트웨어 사용하듯이 바로 사용할 수 있는 장점이 있다. 또한 여러 대에 배포할 수 있는 확장성을 가진다.

* react-native로 iOS/Android 앱 동시 개발

리액트 네이티브는 페이스북에서 만든 오픈소스 모바일 애플리케이션 프레임워크이다. JavaScript로 개발이 가능하며 하나의 프로그래밍 언어로 IOS와 안드로이드 모바일 앱을 동시에 개발할 수 있는 크로스 플랫폼이다.

페이스북의 리액트에서 파생되었기 때문에 그 방식을 그대로 사용했다. React는 Component 기반으로 기능들을 분리해 재사용할 수 있는 구조로 되어 있다. 또한 기존에 리액트로 개발된 사이트가 있다면 그 사이트의 Component 코드를 그대로 사용할 수 있기 때문에 재사용 가능하면 생산성은 더욱 올라간다.  
  
또한 하이브리드 앱의 특징으로 두 개의 OS인 Android와 IOS 모두 한 번에 개발할 수 있기 때문에 개발을 효율적으로 할 수 있다.

* expo 사용으로 RN프로젝트의 테스트 및 배포를 용이

Expo는 RN 개발 시에 개발, 구축, 배포 과정을 빠르게 진행할 수록 도와주는 툴이고 서비스 셋이다. Expo는 RN을 개발 할 때 React-native-cli로 프로젝트를 생성하는 방법 말고, expo-cli 프로젝트로 생성하여 사용할 수 있다.

1. 과제 추진 계획

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 7월 | | | | 8월 | | | | 9월 | | | |
| 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 |
| 음성 인식 API | 서버 구현 | | | | | |  |  |  |  |  |  |
|  | 사투리 데이터 추가 | | | | | | |  |  |  |  |
|  |  |  | 웹/앱, 번역 서비스 연결 | | | | | | |  |  |
|  |  |  |  |  | 서버 구동 테스트 | | | | | | |
| 웹/앱  서비스 | 녹음 기능 앱 개발 | | | | | | |  |  |  |  |  |
|  |  |  | 웹/앱 디자인 및 기능 추가 | | | | | | |  |  |
|  |  | 녹음 기능 웹 개발 | | | | | | | |  |  |
|  |  |  | 도커 사용 및 배포 관리 | | | | | | | | |
| 방언 표준어 번역 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

1. 구성원별 진척도

|  |  |  |
| --- | --- | --- |
| 이름 | 내용 | 진척도 |
| 김윤종 | 방언 텍스트, 표준어 텍스트 데이터 수집 | 완료됨 |
| 방언 텍스트, 표준어 텍스트 Machine Translation 진행 | 완료됨 |
| 방언 텍스트, 표준어 텍스트 데이터 정제 및 추가 | 진행중 |
| Machine Translation 성능 향상 | 진행중 |
| 김도영 | 녹음 기능 앱 개발 | 진행중 |
| 웹/앱 디자인 및 기능 추가 | 진행중 |
| 녹음 기능 웹 개발 | 진행중 |
| 도커 사용 및 배포 관리 | 진행중 |
| 황승찬 | 서버 구현 | 진행중 |
| 사투리 데이터 추가 | 진행중 |
| 웹/앱, 번역 서비스 연결 | 진행중 |
| 서버 구동 테스트 | 보류 |

1. 과제 수행 내용 및 중간 결과

## 음성인식 API

구글 STT API, 네이버 CLOVA CRS, 카카오 Speech API 비교 분석

|  |  |  |  |
| --- | --- | --- | --- |
|  | 구글  STT(Speech-to-Text) | 네이버  CRS | 카카오톡  Speech API |
| 가격 | 60분까지 무료  60분 이상 진행시  15초당 0.006USD | 유료(15초당 4원) | 무료(이전엔 유료였음) |
| 제공형태 | Android SDK  ios SDK  REST API | Android SDK  ios SDK | Android SDK  ios SDK  REST API |
| 사용자 지정 단어 추가 가능여부 | O | X | X |

[표 1] 음성인식 API 비교표

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 1] 카카오 Speech API 사이트 설명

사용자의 음성을 사투리 텍스트로 바꾸는 것이 이번 프로젝트에서 음성 인식 파트의 주 목적이다. 본인은 2022년 기준 이용 가능한 open API를 사용하거나, 음소를 추출 가능한 프로그램을 만드는 두 가지 측면에서 접근하였다. 결론적으로 음소 추출 방식은 현재 본인 실력으로 구현이 힘들고, 사투리 음성 데이터의 부족으로 open API 방식을 선택하게 되었다.

카카오와 네이버는 한국 표준어에 특화되어 있지만, 카카오 Speech API 설명 예시처럼 사투리는 지원하지 않는다. 특히 네이버는 Rest API를 지원하지 않아 웹 개발이 어렵다. 반면, 구글의 Speech API는 사용자 지정 단어 (Speech Context)라는 것을 지원하여 표준어에 있는 단어(문장) 이외에도 커스텀이 가능하며, 사투리 문장을 만들어내야 하는 목표 특성 상 구글 API를 선택하게 되었다.

구글 STT(Speech-to-Text) API speech Context

텍스트이(가) 표시된 사진

자동 생성된 설명[[1]](#footnote-1)

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림 2] 구글 지정 단어 기능 설명(Speech Context)

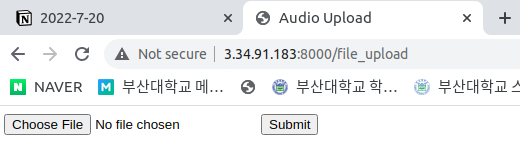
음성 인식 프로그램에서 임의의 음성을 문장으로 만드는 원리는 위의 논문 내용의 이론을 따른다.한 음원의 파형 배열은 여러 음소로 나타낼 수 있고, 한 문장 배열 또한 여러 음원으로 나타날 수 있다. 따라서 한 음성 데이터를 한 문장으로 특정하기 위해서는 음소 뿐만 아니라 다른 요소를 더 필요로 한다. 특정 단어가 함께 나타날 확률 p(w)이다. 구글 STT API의 사용자 지정 단어의 원리는 특정 단어로 해석될 확률을 높이는 것으로 단어를 추가한 것과 같은 효과를 내는 것이다.구글 API 또한 표준어 기반이기 때문에 사투리 문장은 비슷한 발음의 표준어 단어로 대체해 버리지만, Speech Context를 통해 단어를 추가하게 되면 다르다.

음성-텍스트 변환 API 서버 구축

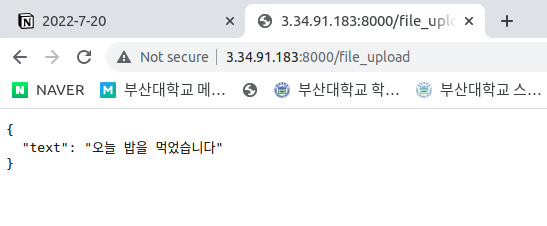
[그림 3] 음성 텍스트 변환 서버 파이썬 코드

|  |  |
| --- | --- |
| 텍스트, 클립아트이(가) 표시된 사진  자동 생성된 설명 | 텍스트이(가) 표시된 사진  자동 생성된 설명 |

웹-앱 과의 연결을 위해 구축한 테스트 서버다. 웹-앱이 오디오 파일을 44100hz의 .wav 형식으로 서버에 올리면 api를 통해 사투리 문장으로 만든 후 텍스트를 json에 넣어 반환하는 단순한 사이트이다. 웹브라우저로 직접 접속할 경우 다음과 같이 단순한 시험용 업로드만 구현되어 있다.



[그림 4] 서버 웹 브라우저 접속 시 모습

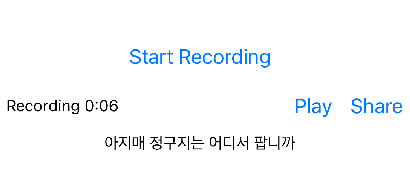


[그림 5] 오디오 파일 업로드 시 결과 json

리액트 웹앱 오디오 녹음 및 서버 연결

|  |  |
| --- | --- |
|  | 텍스트이(가) 표시된 사진  자동 생성된 설명 |

웹-앱 파트에 추가될 서버 연결 코드도 구현했다. 오디오 형식과 서버 주소, 업로드 파일 형식 등을 맞춰야 하는 이유로 음성인식 API 파트에서 구현했다.



[그림 6] 앱 내 구현 화면

오디오 녹음과 텍스트를 서버에서 받아 앱 내 화면에 출력하는 것까지 구현했다. 서버와 연결은 완료한 상태다.

## 문장 번역 서비스

모델의 구축은 마친 상태이며 현재까지의 모델의 성능을 다른 모델, 그리고 처음 데이터셋을 바꾸기 전 모델과 비교 분석을 시도했다. 방언의 특성 상 해당 문장이 잘 번역되었는지에 대한 평가는 정성평가로 진행하였다.

- AI HUB 데이터 셋(50,000) vs 영화 대사 추가(55,000)

둘 다 동일한 평가를 위해 데이터셋에 사용하지 않은 친구1 대사를 사용하였고, 에폭 수는 50으로 설정하였다.



**순서대로 원문, 영화 대사 추가 모델, ai hub만 사용한 모델**

영화 데이터셋을 추가한 모델에서 훨씬 더 좋은 모델의 성능을 볼 수 있었다.

- Tunib vs My Model

기준으로 잡은 모델은 NLP AI스타트업 기업 Tunib에서 제공하는 방언 번역기 모델이다.

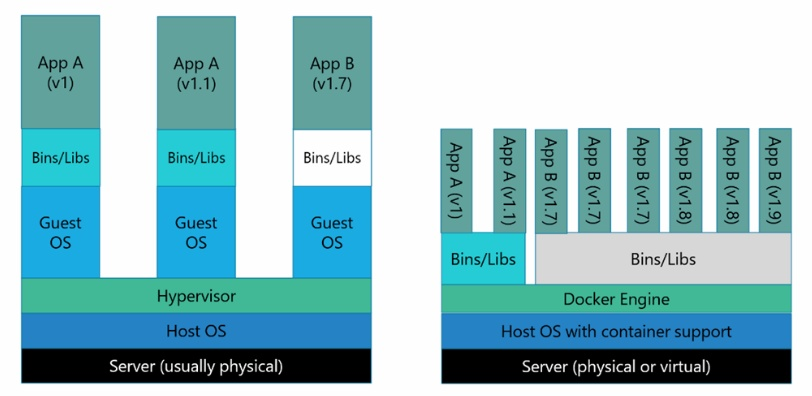


**순서대로 원문, tunib모델, 우리 팀 모델**

전혀 사용하지 않은 친구1의 대사를 테스트 데이터로 사용하여 진행한 결과는 우리 팀의 모델이 조금 더 맞춘 문장도, 조금 더 틀린 문장이 있지만 거의 동등한 성능을 내는 것을 알 수 있었다.

## 웹-앱 서비스

Docker로 nginx + react 프로젝트 관리하기



VM 가상화 플랫폼 vs Docker 가상화 플랫폼

VM같은 경우엔 Host OS 위에 가상화를 시키기 위한 Hypervisor 엔진 그리고 그 위에

Guest OS를 올려 사용합니다. 이는 가상화된 하드웨어 위에 OS가 올라가는 형태로 거의

완벽하게 Host와 분리된다고 봐도 무방합니다. 반면에 컨테이너 기반 가상화는 Docker 엔

진 위에 Application 실행에 필요한 바이너리만 올라가게 됩니다. OS 가상화를 보면 Host

OS와 완전히 분리되는 장점은 있지만 OS위에 OS를 올리기 때문에 무겁고 느릴수 밖에 없

습니다. 하지만 컨테이너 기반 가상화는 Host OS 그리고 Docker 엔진위에서 바로 동작하

며 Host의 커널을 공유합니다. 커널을 공유하게 되면 io처리가 쉽게 되어 성능의 효율을 높

일 수 있습니다.

Dockers는 애플리케이션 구축, 테스트 및 배포를 할 수 있는 소프트웨어 플랫폼 입니다.

컨테이너(Container)라는 표준화된 유닛으로 패키징하며, 이 컨테이너에는 라이브러리, 시

스템 도구, 코드, 런타임 등 소프트웨어를 실행하는 데 필요한 것이 포함되어 있습니다.

Docker는 코드를 실행하는 표준 방식을 제공합니다. 또한 컨테이너를 위한 운영 체제입니

다. 가상 머신이 서버 하드웨어를 가상화하는 방식과 비슷하게 컨테이너는 서버 운영 체제를 가상화합니다. Docker는 각 서버에 설치되며 컨테이너를 구축, 시작, 또는 중단할 수 있는

간단한 명령어를 제공합니다.

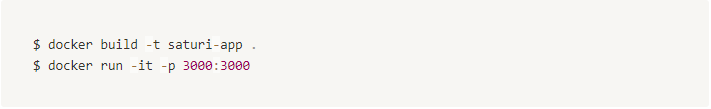
Docker를 사용하면 코드를 더 빨리 전달하고, 애플리케이션 운영을 표준화하고, 코드를 원

활하게 이동할 수 있습니다. 또한 리소스 사용률도 높여 효율적으로 관리할 수 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Dockerfile 생성



Docker Image 생성 후, Container 만들고 실행

텍스트이(가) 표시된 사진

자동 생성된 설명

Dockerfile-prod 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

default.conf 생성

텍스트이(가) 표시된 사진

자동 생성된 설명

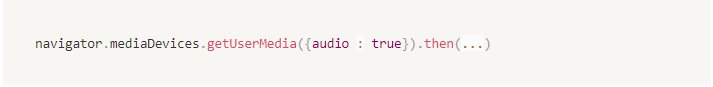
Dockerfile-prod 빌드 후, 80포트로 실행

텍스트, 스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명

Docker Desktop 스크린샷

React 에서 녹음 파일 생성하기

텍스트이(가) 표시된 사진

자동 생성된 설명

녹음 권한 받기

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

녹음 하기

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

녹음 중지하기

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

파일로 변환하고 링크 생성

React-Natve로 iOS/Android 개발



리액트 네이티브는 Javascript 기반 프로그래밍 언어로 IOS와 안드로이드 앱을 한 번에 개발할 수 있는 크로스 플랫폼입니다. 리액트 네이티브는 네이티브 브릿지(Native Bridge)를 통하여 스레드(Native Thread)와 통신하기 때문에, 웹앱과 다르게 성능에 최적화되어 있습니다.

리액트는 컴포넌트 기반으로 기능을 분리하여 재사용할 수 있고, IOS와 안드로이드 앱을 동시에 개발할 수 있기 때문에 개발을 효율적으로 할 수 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Expo는 리액트 네이티브를 베이스로 한 것으로 앱을 간편하게 빌드, 배포를 할 수 있도록 도와주는 프레임워크입니다. Expo는 바닐라 리액트 네이티브로 앱을 개발할 때 필요한 초기 환경 설정과 빌드를 쉽게 해주는 장점이 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

앱 스크린샷

1. 김우성, 구명완, 통계적 언어 모델의 clustering 알고리즘과 음성 인식에의 적용, 한국정보과학회 언어공학연구회:학술대회논문집, 1996, 146쪽 [↑](#footnote-ref-1)