



HANGGLIDER

EEE | 박동호 서예지 양창원 유세빈 이희진



Contents

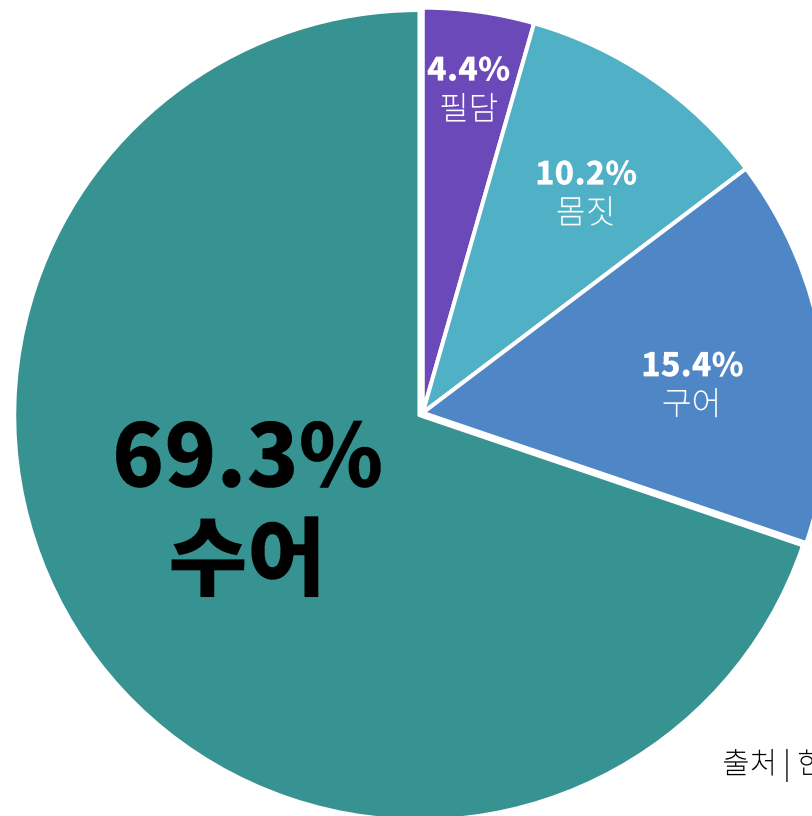
- 01** 프로젝트 개요
- 02** 프로젝트 목적
- 03** 개발 환경
- 04** 팀원 역할
- 05** 프로젝트 프로세스
- 06** 프로젝트 단계별 내용
- 07** 프로젝트 결과
- 08** 관련 논문 및 레퍼런스

프로젝트 개요

농인이 주로 사용하는 의사소통 방법은 **수어**

농인의 제1언어는 '수어'다

일상적인 의사소통에서 가장 많이 사용하는 언어가
'수어'라고 응답한 농인이 69.3%로 조사되었다



출처 | 한국수어 사용 실태조사

프로젝트 개요

코로나19 사태 이후 언택트 시대에 맞게 교육 패러다임이 변화하고 있다.

이러한 변화 속에서 **청각장애인들은 비대면 온라인 수업 수강에 어려움**을 겪고 있다.



농학교 학생 1

수어만 하는 친구들은 수업을 아예 이해하지 못해요.

자막이 나와도 한국어와 수어의 문법체계가 달라서 이해하는 데 한계가 있어요.

맞아요. 또 구화를 하더라도 강사가 등을 돌리거나 영상 화질이 안 좋으면 입모양을 제대로 볼 수가 없어요. 흐름을 놓치면 자막이 어느 부분을 말하는지도 모르겠어요.



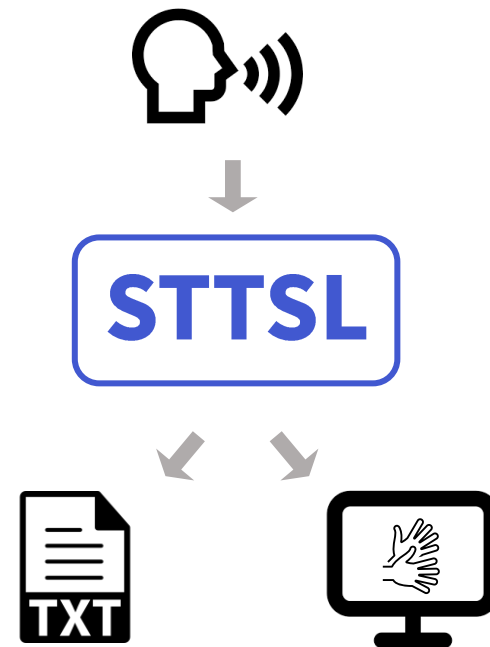
농학교 학생 2

프로젝트 개요

온라인 화상플랫폼 상에서 발언자의 **음성**을 **자막**과 **수어 영상**으로 **실시간** 변환

Speech to Text Sign Language

청각장애인 학생들의 교육 평등권을 보장할 수 있도록
음성을 텍스트와 수어로 변환해서 제공하는 프로젝트를 기획했다.



프로젝트 목적

HANGGLIDER

행동과 글을 사용자에게 자유롭게 변환해서 제공하는 서비스를 구축하는 것

사용자의 음성을 자막과 수어로 바꾸어 보여줌으로써
청각장애인도 청인과 동등한 교육의 기회를 받도록 장려한다

개발 환경

버전 관리



대면



Kakaotalk



Github



Swit

소통 & 개발 환경

Beautifulsoup4	4.6.0	Numpy	1.18.1
Blender	2.90.1	Openpose	1.3
Django	3.1.2	Pykomoran	0.1.5
Html5lib	1.0.1	Python	3.7.6
Json5	0.9.1	Requests	2.22.0
Moviepy	1.0.3	Urllib3	1.25.8



10월 24일

Demo 1.0v

임시 데이터 DB 구축, 자연어처리 알고리즘 구현



11월 3일

Demo 1.1v

Webcam 연동, 자막 기능 추가, STT 모델 연동, mapping 알고리즘 구현



11월 12일

Demo 1.2v

애니메이션화된 최종 DB 구현, 웹 어플리케이션 구축, 오류 수정 및 성능 개선

04

팀원 역할



팀장 | 박동호

자연어 전처리 모델 개발
웹 연동



팀원 | 서예지

STT API 연동
웹디자인



팀원 | 양창원

STT API 연동
수화 애니메이션 제작



팀원 | 유세빈

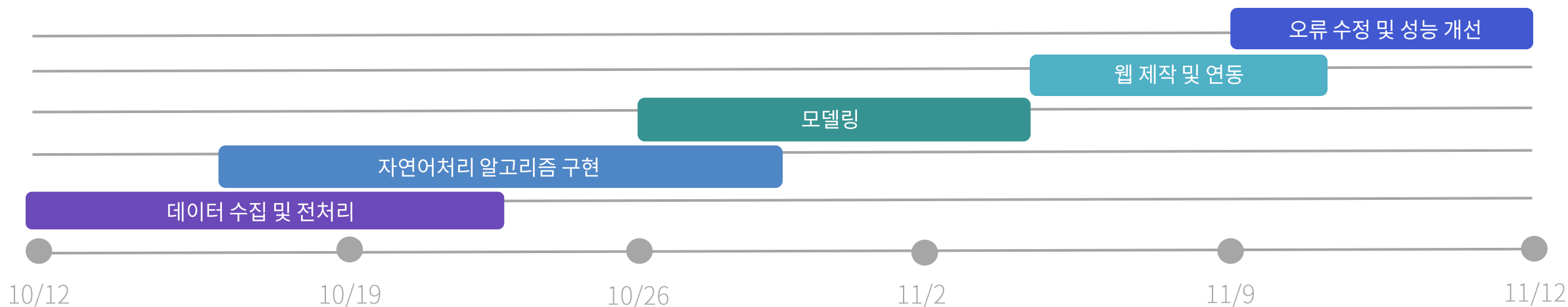
자연어 전처리 모델 개발
웹 연동



팀원 | 이희진

데이터 수집
DB 구축

프로젝트 프로세스



1주차 | 데이터 수집 및 전처리

약 2만 5천개 수어영상 크롤링 및 촬영, 프레임 편집 및 애니메이션화

2주차 | 자연어처리 알고리즘 구현

수어 문법에 맞게 형태소 분석 및 자연어처리 알고리즘 구현

3주차 | 모델링

STT 모델, 어휘간 유사도 분석 API 연동

4주차 | 웹 제작 및 연동

Django, Javascript, Sqlite 활용

5주차 | 오류 수정 및 성능 개선

mapping 체계 수정, STT 모델과 자연어처리 알고리즘 개선

프로젝트 단계별 내용

DATA

ALGORITHM

MODEL

데이터

1. 데이터 수집

원본 수어 영상 데이터가 균일하지 않아 재생 속도와 프레임 수를 편집하는 전처리를 거쳐 DB에 저장했다.

2. 수어 영상 애니메이션화

수어 영상 데이터 속 등장하는 인물이 계속 바뀌기 때문에 Blender와 Open pose를 활용해 영상 속 인물을 캐릭터로 변환하여 DB에 저장했다.

프로젝트 단계별 내용

DATA

```
def cut_video(url):
    vcap = cv2.VideoCapture(url)
    # vcap의 프레임 수 get
    frame_cnt = vcap.get(cv2.CAP_PROP_FRAME_COUNT)
    # 프레임 수 1/3로 나누기
    frame_cnt = int(frame_cnt/3)
    frame = str(frame_cnt)
    return frame

def save_signlanguage_video(href, frame, count, type):
    input_location = 'aws/media/signLanguage/' + type + '/' + str(count) + '.mp4'
    output_location = 'aws/media/signLanguages/' + type + '/' + str(count) + '.mp4'
    output_location2 = 'aws/media/sign/' + type + '/' + str(count) + '.mp4'

    #if txt is not None:
    #    txt.encode('utf8')

    end_time = int(frame)/30
    ffmpeg_extract_subclip(href, 0, end_time, targetname = str(input_location))
    ffmpeg_resize(input_location, output_location, (560, 360))
    clip = VideoFileClip(output_location)
    new_clip = crop(clip, x1=70, y1=0, x2=490, y2=270)
    new_clip.write_videofile(output_location2)

    return output_location2
```

1. 데이터 수집

크롤링 시 데이터 전처리

- 크롤링한 데이터 : 수어 영상, 의미, 품사 정보
- **cut_video** 함수 :
영상의 프레임 수 1/3
- **save_signlanguage_video** 함수 :
수화 영상 길이와 사이즈,
프레임 전처리 후 new_clip으로 저장
저장한 로컬 경로를 return하여 DB화

프로젝트 단계별 내용

DATA

```
def GetPoint(Array, index):
    baseIndex = index*3
    x = float(Array[baseIndex])
    y = float(Array[baseIndex + 1])
    reliability = float(Array[baseIndex + 2])
    return [x,y,reliability]

def rotationPoint(Array, index):
    if index > 0:
        alpha_degree = math.atan2((GetPoint(Array[index] ,index)[0] -
                                   GetPoint(Array[index-1] ,index-1)[0]),
                                   GetPoint(Array[index] ,index)[1] -
                                   GetPoint(Array[index-1] ,index-1)[1]) * 180 / math.pi
        beta_degree = 90 - alpha_degree
    else:
        alpha_degree = 0
        beta_degree = 0
    return [alpha_degree/57.3, beta_degree/57.3]
```

2. 수어 영상 애니메이션화

캐릭터 관절 위치, 변화 구하기

- **GetPoint** 함수 :
open pose로 얻은 keypoint
json 파일 중 특정 관절에 대한 위치값
- **rotationPoint** 함수 :
현재 상태의 관절의 변화 각도량

프로젝트 단계별 내용

DATA

```

if index > 0:

    x1_right_arm_1 = GetPoint(pose_list[index-1], 3)[0]
    x2_right_arm_1 = GetPoint(pose_list[index], 3)[0]
    z1_right_arm_1 = GetPoint(pose_list[index-1], 3)[1]
    z2_right_arm_1 = GetPoint(pose_list[index], 3)[1]

    right_arm_1_angle = [rotationPoint(x1_right_arm_1,x2_right_arm_1,
                                      z1_right_arm_1,z2_right_arm_1)[0],
                        rotationPoint(x1_right_arm_1,x2_right_arm_1,
                                      z1_right_arm_1,z2_right_arm_1)[1]]

    x1_right_arm_2 = GetPoint(pose_list[index-1], 4)[0]
    x2_right_arm_2 = GetPoint(pose_list[index], 4)[0]
    z1_right_arm_2 = GetPoint(pose_list[index-1], 4)[1]
    z2_right_arm_2 = GetPoint(pose_list[index], 4)[1]

    right_arm_2_angle =[rotationPoint(x1_right_arm_2,x2_right_arm_2,
                                    z1_right_arm_2,z2_right_arm_2)[0],
                      rotationPoint(x1_right_arm_2,x2_right_arm_2,
                                    z1_right_arm_2,z2_right_arm_2)[1]]

else:

    right_arm_1_angle = initial_angle_right
    right_arm_2_angle = (0,0,0)

right_arm_2.rotation_mode = 'XYZ'
right_arm_2.rotation_euler.x = right_arm_2_angle[0]
right_arm_2.rotation_euler.y = right_arm_2_angle[1]

index += 1

```

2. 수어 영상 애니메이션화

위치값 차이와 각도 변화량 구하기

- 캐릭터 신체 상반신 관절에 대한 위치값 및 각도 변화량 입력
- 특정 관절에 대한 i번째와 i-1번째의 위치값 차이 구함
- 위치값 차이에 따른 각도 변화량 구함

프로젝트 단계별 내용

DATA

ALGORITHM

MODEL

알고리즘

1. 수어 기반 형태소 분석

한국어와 수어의 문법 체계는 다르다.

STT로 넘어온 한글 문장을 수어 형식에 맞게 형태소를 나눠줘야 한다.

2. DB와 Mapping

형태소 분석이 완료된 후

수어 형식에 맞게 나눠진 단어와 일치하는 영상을 DB에서 가져온다.

프로젝트 단계별 내용

ALGORITHM

```
def relocateMorpheme(self, subtitle_path):
    result=[]
    word_list=[]
    morph_list=[]
    line = self.komoran.get_list(subtitle_path)
    line = self.splitLine(line)    # ex) [['식사', 'NNG'], ~~~]
    for w, m in line:
        r, word, morph = self.pr.process_morph(m, w)
        if r == 1:
            if (word == 'ㄴ니까') or (word == '하다') or (word == '끝'):
                if len(result) == 0:
                    word_list.append(word)
                    morph_list.append(morph)
                elif result[len(result) - 1][0] != word:
                    word_list.append(word)
                    morph_list.append(morph)
            else:
                word_list.append(word)
                morph_list.append(morph)
    result.append(word_list)
    result.append(morph_list)
    return result
```

1. 수어 기반 형태소 분석

텍스트를 수어 문법에 맞게 변환

입력 : 음성에서 변환된 텍스트 데이터

출력 : 수어 문법에 맞게 변환된 단어와 형태소 리스트

- **komoran.get_list** 함수 :
문장을 수어 형태소에 맞게 변환
- **process_morph** 함수 :
형태소 분류 및 매칭

프로젝트 단계별 내용

ALGORITHM

```
# 형태소가 숫자일 때,  
if word.isdigit():  
    find_word = Number.objects.get(word=word)  
    results.append(find_word.location)  
  
# 형태소가 DB 검색 시 1개 일 때,  
elif Basic.objects.filter(word=word).count() == 1:  
    find_word = Basic.objects.get(word=word)  
  
# 형태소가 DB 검색 시 2개 일 때,  
elif Basic.objects.filter(word=word).count() == 2:  
    find_word = Basic.objects.filter(word=word)  
    results.append(find_word[0].location)  
  
# 형태소가 DB 검색 시 여러 개 일 때,  
elif Basic.objects.filter(word=word).count() > 2:  
    find_word = Basic.objects.filter(word=word)
```

2. DB와 Mapping

수어 문법에 변환된 단어와 DB 매칭

입력: 수어 문법에 맞게 변환된 단어

출력: 매칭된 수어 영상의 저장 경로

- 단어가 숫자일 때 또는 일치하는 데이터가 1,2개일 때
해당 DB에 영상 매칭
- 데이터가 여러 개일 경우 유사도 함수 실행

프로젝트 단계별 내용

DATA

ALGORITHM

MODEL

모델

1. STT (Speech To Text)

웹 내장 마이크를 사용해서 음성을 입력 받는다.

입력 받은 음성을 화면에 띄운다.

2. 단어간 어휘 유사도 분석

텍스트로 변환된 문장을 수어 형식에 맞게 나눈다.

나뉘진 단어 중 DB에 동음이수어*가 있는 경우 단어간 유사도를 분석한다.

* 동음이수어 : 발음은 동일하나 의미가 다른 수어

프로젝트 단계별 내용

MODEL

```
recognition.onresult = function(event) {
  var interim_transcript = '';
  for (var i = event.resultIndex; i < event.results.length; ++i) {
    if (event.results[i].isFinal) {
      final_transcript += event.results[i][0].transcript;
      final_transcript = capitalize(final_transcript);
      final_span.innerHTML = linebreak(final_transcript);
      interim_span.innerHTML = linebreak(interim_transcript);
      $.ajax({
        type: 'POST',
        url: url4,
        data: {
          text1: final_span.innerHTML,
          csrfmiddlewaretoken: $('input[name=csrfmiddlewaretoken]').val(),
          action: 'POST'
        },
      },
      success: function(json){
        console.log("data pass", json.q);
        new_arr = json.q;
        if (firstview){
          video_list = $.merge([], new_arr);
          onload();
        } else{
          video_list = $.merge( $.merge([], video_list), new_arr);
        }
        final_span.innerHTML = '';
        interim_span.innerHTML = '';
        final_transcript = '';
        firstview = false;
      }
    )
  }
}
```

1. STT (Speech To Text)

음성을 텍스트로 변환하여 웹에 연동

입력 : 사용자의 음성

출력 : 입력 받은 음성이 변환된 텍스트

- STT API를 통해 얻은 텍스트를 html span태그에 담아 자막을 구현
- ajax를 이용하여 비동기 통신을 구현하여 span 태그에 담긴 텍스트를 모델을 통해 처리

프로젝트 단계별 내용

MODEL

```
def calc_similarity(self, true_word, ref_word):
    result_sim = []
    for i in range(len(ref_word)):
        response = self.similarity_voca(true_word, ref_word[i])
        response = json.loads(str(response.data, "utf-8"))
        sim = response["return_object"]["WWN WordRelInfo"]["WordRelInfo"]["Similarity"]
        # print(str(ref_word[i]) + ": " + str(sim))
        for i in range(len(sim)):
            s = + sim[i]["SimScore"]

        a = s / len(sim)
        result_sim.append(a)
        # print('평균: '+str(a))
    print(result_sim)
    if max(result_sim) <= 0.0:
        return -1
    return result_sim.index(max(result_sim))
```

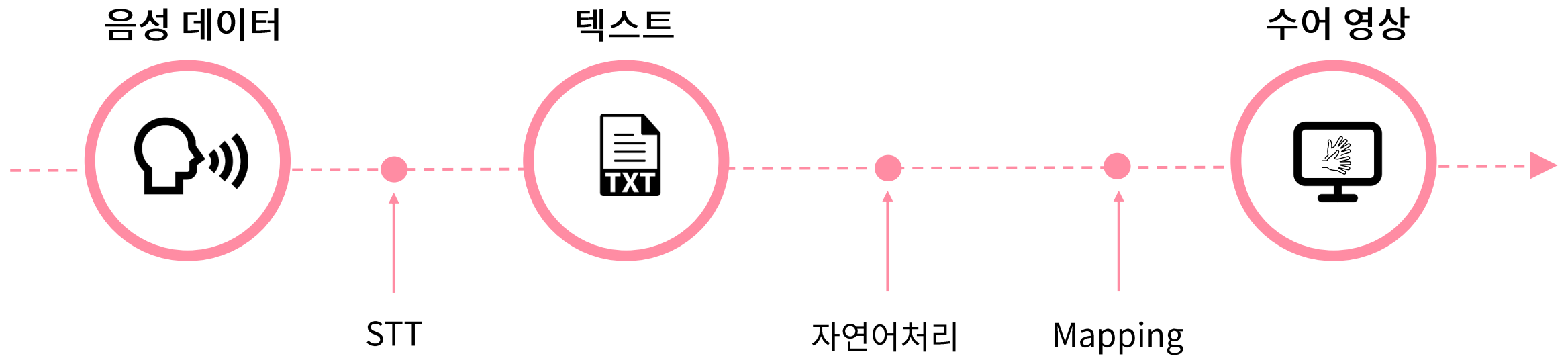
2. 단어간 어휘 유사도 분석

true_word와 ref_word를 입력하여 유사도 비교

- **true_word** - STT를 통해 얻은 문장에서 동음이의어와 거리가 제일 가까운 명사
- **ref_word** - 동음이의어 단어들 중 의미를 대표하는 명사를 가진 참조단어 들의 리스트
- **true_word**와 **ref_word** 속 각 단어들과 유사도를 비교하여 수치로 나타내어 제일 높은 수치를 가진 단어를 반환

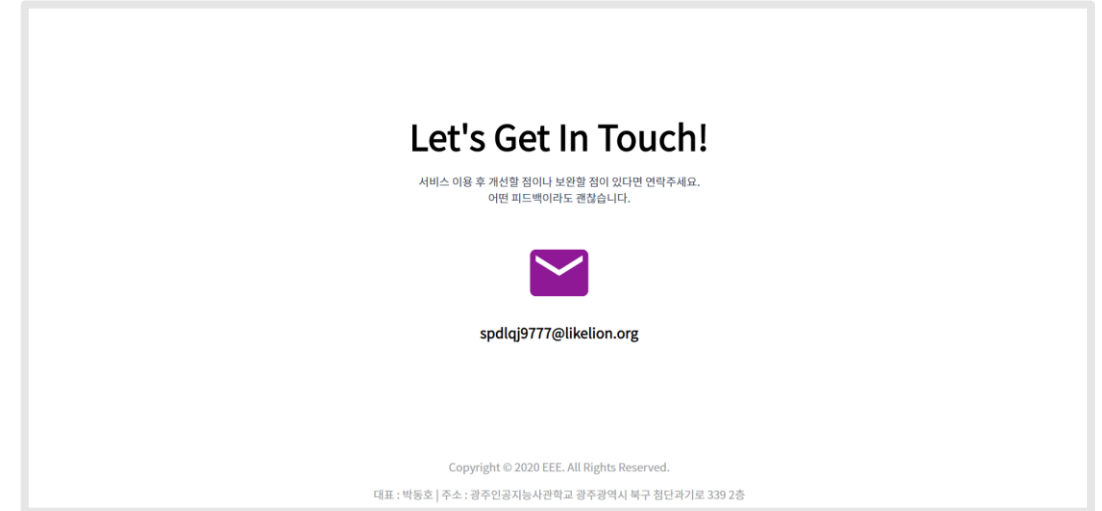
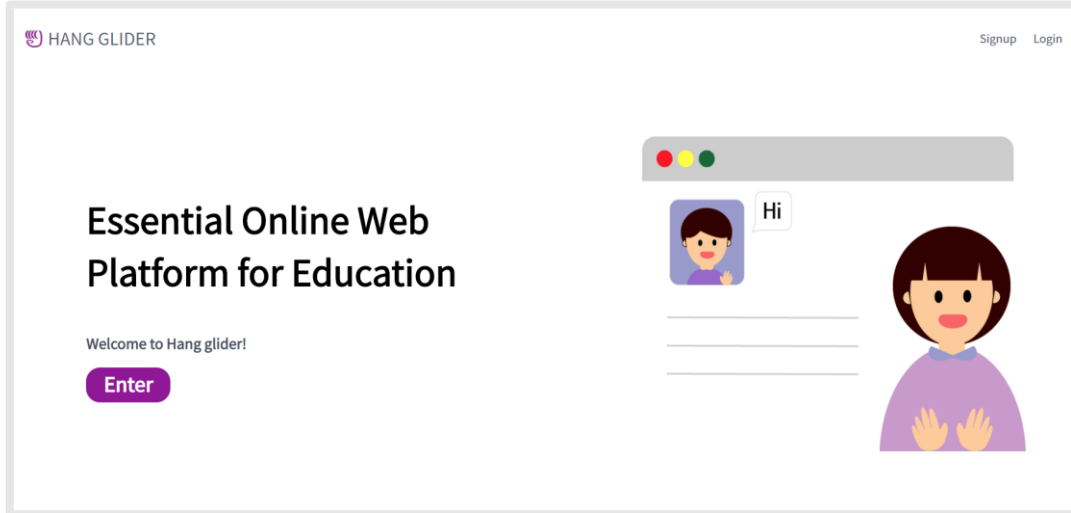
프로젝트 결과

1. 최종 결과 : 수업 진행자의 **음성**이 **실시간**으로 입력되어 **텍스트**와 **수어 영상**으로 출력된다



프로젝트 결과

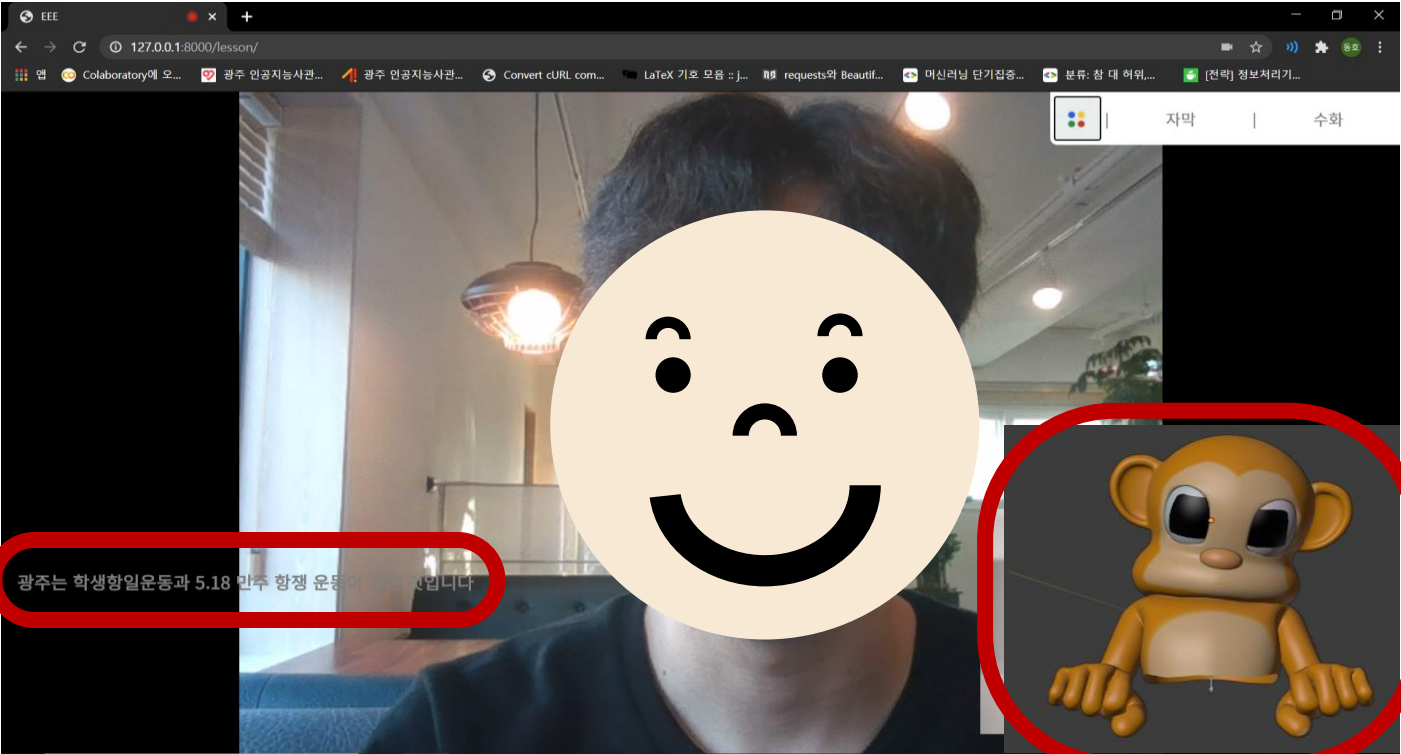
1. 최종 결과 : 온라인 화상플랫폼을 웹으로 구현 / 언제든지 개발자에게 불편사항 접수 가능



Demo 1.2v home.html

프로젝트 결과

1. 최종 결과 : 온라인 화상플랫폼 상에서 발언자의 음성을 자막과 수어 영상으로 실시간 변환



자막

수어 영상

Demo 1.2v lesson.html

프로젝트 결과

2. 프로젝트 자체 평가

총 문항 개수 : 20문항

측정 방법 : ○ 5점 ▲ 3점 ✕ 0점

목표 점수 : 80점

평가 항목	평가 내용	달성여부
데이터 수집	1) 신뢰도 있는 데이터를 확보했는가	○
	2) 필요한 데이터를 충분히 확보했는가	▲
	3) 영상 데이터의 재생 속도와 프레임 수를 적합하게 편집했는가	○
	4) 모션 인식을 활용하여 자체 캐릭터로 애니메이션을 만들었는가	○
자연어처리 알고리즘	5) 한글 문장을 수어 문법에 맞게 형태소를 나누었는가	○
	6) 나뉜 형태소에 따라 영상 데이터를 알맞게 매칭했는가	○

프로젝트 결과

2. 프로젝트 자체 평가

평가 항목	평가 내용	달성여부
자연어처리 알고리즘	7) 동음이수어가 존재할 때 유사도 측정을 기반으로 정확하게 구분했는가	▲
모델링	8) STT 모델의 속도가 실시간에 적합한가	○
	9) 변환된 텍스트의 의미와 맞춤법이 정확한가	○
	10) 음성이 말하는 속도에 관계없이 잘 인식되어 변환되는가	▲
	11) TEXT가 자동으로 적절히 나뉘져서 자연어처리로 넘어가는가	▲
	12) 영상 데이터가 mapping 되어 출력되는 속도가 말하는 속도와 비슷한가	○
웹 제작 및 연동	13) 쌍방향 화상 플랫폼을 구축했는가	×

프로젝트 결과

2. 프로젝트 자체 평가

평가 항목	평가 내용	달성여부
웹 제작 및 연동	14) 웹캠 스트리밍이 버퍼링 없이 적합한 속도로 출력되는가	○
	15) 출력되는 웹캠 화면의 화질이 적합한가	○
	16) 웹페이지가 구조적으로 잘 짜여 있는가	○
	17) 웹 디자인이 한 눈에 깔끔하게 잘 보이는가	○
기타	18) 프로젝트 진행 과정에서 역할 분배가 적절히 이루어졌는가	○
	19) 소통을 통한 협업이 원활하게 이루어졌는가	○
	20) 일일보고서 작성, 회의 참여도가 90% 이상 달성했는가	○

프로젝트 결과

2. 프로젝트 자체 평가

목표 점수 : 80점

실제 점수 : 87점

세부 항목 : ○ 15개 ▲ 4개 ✕ 1개

평가 결과 : 목표 달성

한줄 평가 :

모든 팀원들이 협업하여 양질의 데이터를 확보한 뒤 자연어처리 알고리즘을 통해 실시간으로 STTSL을 웹에 구현하였다.

프로젝트 결과

3. 피드백 & 보완점

데이터	수화로 표현하지 못하는 단어들이 많아 데이터 수집에 어려움을 겪었음 꼭 필요한 단어는 수어 통역사를 초빙해 촬영했지만, 그 외 데이터는 확보하지 못했음
STT 모델링	길이 제한으로 문장을 넘길 시 음성인식이 되지 않는 텀이 생김 음성 속도가 빠를 경우 인식을 하지 않음
Mapping	숫자의 경우 DB와 매치가 잘 되지 않음
유사도 API	단어와 단어의 유사도를 구할 때 정확도가 떨어짐
웹	RTC 구현을 하지 못해 쌍방향 소통을 할 수 없음

관련 논문 및 레퍼런스

Moviepy 공식 문서 <https://zulko.github.io/moviepy/>

행동 인식을 위한 open pose <https://github.com/CMU-Perceptual-Computing-Lab/openpose>

행동 인식을 위한 open pose <https://github.com/CMU-Perceptual-Computing-Lab/openpose>

open pose를 활용한 rig 추출 및 애니메이션 구현 <https://github.com/nkeeline/OpenPose-to-Blender-Facial-Capture-Transfer>

open pose, SVG 이미지 파일을 활용한 2D 애니메이션 구현 <https://github.com/yemount/pose-animator>

KOMORAN 공식 문서 <https://komorandocs.readthedocs.io/ko/latest/index.html>

자연어처리 형태소 분석기 KOMORAN <https://github.com/shineware/KOMORAN>

어휘간 유사도 분석 http://aiopen.etri.re.kr/guide_word.php

Urllib <https://urllib3.readthedocs.io/en/latest/>

STT 모델(Web Speech API) https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API



THANK YOU

감사합니다