

# Sprint 3 Review and Outcomes

Capstone Design

---

AI정보공학과\_ TEAM. 두근두근

권주연, 김지윤, 이상승, 한은영

1. Team Introduction ( 팀 소개 )
2. Sprint Outputs and Outcomes ( 목표 및 목적 )
3. Progress Report ( 진행상황 보고 )
4. Demo of Deliverables ( 성과물 데모 )
5. Challenges and Solutions ( 문제점 및 해결책 )
6. Sprint Retrospective ( 스프린트 회고 )
7. Next Sprint Planning ( 다음 스프린트 계획 )
8. Personal Retrospective ( 개인별 회고 )

# Last Sprint Goals

## 목표

< 지윤 >

- **경상남도 사투리 매핑**: 16,000단어의 경상남도 사투리 매핑
- **STT**: '현재 서비스와 코드의 연결에서 모듈의 충돌이 발생했습니다.' 충돌 해결

< 주연, 상승 >

- **표준어 TTS**: 수도권 사용자 10명이 TTS를 들었을 때, 9명 이상이 표준어로 인식하는 것
- **내 목소리 TTS**: 내 목소리를 아는 사람과의 통화 시, TTS를 사용해도 어색함 없이 1분 동안 통화를 유지하도록 제작

< 은영 >

- **Waveform**: 표준어와 내 목소리의 주파수 비교를 사용자 친화적이게 시각적 출력
- **LLM**: 현재 개발 중인 UI/UX와의 연결, 간단 구성

# Outputs and Outcomes

---

## Sprint Outputs

- **APPLICATION & LLM**
  - 방언 교정 애플리케이션 기본 인터페이스 설계 및 프로토타입 개발
- **TRANSLATE**
  - 방언과 표준어 간의 매핑 데이터베이스 구축
  - 방언 텍스트를 표준어 텍스트로 변환하는 알고리즘 개발

# Outputs and Outcomes

---

## Sprint Outputs

- **TTS (Text-to-Speech)**
  - 표준어 텍스트를 표준어 음성으로 변환하는 TTS 모델 개발, 학습
- **딥보이스**
  - 표준어 음성을 내 음성으로 변환하여 출력하는 모델 개발, 학습
- **WAVEFORM**
  - 녹음된 음성 데이터를 분석하여 주파수 스펙트럼을 생성하는 알고리즘 구현

# Outputs and Outcomes

---

## Sprint 중요성

방언 사용자들이 표준어를 자연스럽게 정확하게 구사하도록 돕는 과정

사용자 친화적인 애플리케이션 인터페이스, 방언-표준어 매핑 데이터베이스,  
TTS 모델, 딥보이스 기술, 그리고 주파수 스펙트럼 분석 알고리즘을 통해  
사용자는 표준어 학습에 필요한 도구를 제공받아 교육 및 직업 기회가 확대됨  
또한, 이 과정은 방언 사용자의 서비스 접근성 개선에 도움

# Roles and Responsibilities

---

## 권주연

### 딥보이스 모델 개발자

- 표준어 아나운서 음성 데이터와 베타 테스트 방언 음성 데이터를 수집, 전처리
- 방언 음성 → 표준어 아나운서 음성 변환하는 딥보이스 모델 설계, 음성 합성 및 변환 기술 적용
- 수집된 데이터를 통해 모델을 학습, 병렬 연산을 활용해 자연스럽게 명료한 음성 변환을 실현
- 모델의 성능을 다양한 지표로 평가, 사용자 피드백 기반 최적화
- 웹 페이지 자동화를 통한 TTS 시스템 통합 구현

## 김지윤

### 방언-표준어 매핑, 데이터 수집 및 전처리

- 인터넷, 지역 언어 연구 자료, 사용자 제공 데이터 등 여러 출처에서의 표준어 및 방언 발화 데이터를 수집
- 수집된 데이터를 정리, 클렌징 → 분석 가능한 형태로 변환
  - 전처리 과정: 노이즈 등 불필요한 정보 제거, 샘플링 레이트 조정, 정규화
  - 수집된 방언 데이터를 표준어와 매핑 → 데이터베이스 구축
- 수집된 방언 데이터를 표준어와 매핑 → 데이터베이스 구축
- 각 방언 발화와 이에 대응하는 표준어를 연결하는 매핑 작업 기록, 관리



# Roles and Responsibilities

## 이상승

### 딥페이크 및 딥보이스 모델 개발자

- 영상 데이터 수집, 전처리, 얼굴 특징 추출, 발음 구간 분할, 노이즈 제거
- Deep face lab, Wav2lip 기반 딥페이크 기술 활용
  - 발음에 따른 입모양(방언→표준어)을 실시간 변환하는 모델 설계, 개발
- 수집된 데이터를 바탕으로 GPU 클러스터를 이용해 모델을 학습, 자연스러운 입 모양 전환을 구현
- 모델 성능을 다양한 지표로 평가, 개선, 튜닝
- 표준어 음성 → 나의 음성으로 변환하는 딥보이스 모델 설계,
- 음성 합성 및 변환 기술 적용

# Roles and Responsibilities

---

## 한은영

### 음성 시각화 및 어플리케이션 개발자, LLM 연결

- 음성 데이터의 주파수 특성 분석 및 시각화
- 딥페이크, 딥보이스 모델 통합
  - 사용자 친화적인 어플리케이션 설계 및 개발
  - 사용자가 쉽게 방언을 표준어로 변환, 결과 확인
- 프론트엔드/백엔드 시스템 설계 및 구현, 모델의 실시간 처리 기능 최적화 및 사용자 인터페이스(UI)와 사용자 경험(UX)을 고려하여 디자인
- 앱 성능 테스트 및 디버깅 → 안정성과 효율성 확보

# Sprint Planning

Story	To Do	In progress	Done
<div>데이터 수집 및 전처리</div>			<div>데이터 수집 계획 수립: 방언 및 표준어 데이터 소스 식별 및 접근 방법 결정</div> <div>데이터 정제: 소음 제거, 데이터 포맷 통일, 필요 없 는 부분 잘라내기</div> <div>데이터 레이블링: 음성 데이터에 정확 한 방언과 표준어 태그 부여</div> <div>데이터 수집 실행: 웹 스크래핑, 공개 데이 터셋 접근, 사용자 녹음 데이터 수집</div> <div>데이터 세트 구성: 훈련, 검증 및 테스트 데이터셋으로 분 할</div>
<div>음성 인식 및 변환 모델 개발</div>		<div>음성 인식 모델 개발: 음성 인식 API선택, API 통합 및 커스텀 모델 개 발</div> <div>모델 최적화: 과적합 방지, 정확도 향 상을 위한 파라미터 조 정</div>	<div>모델 테스트 및 평가: 개발한 모델을 사용하 여 실제 데이터셋에 테 스트 실행</div> <div>음성 변환 딥러닝 모델 개발: 딥보이스 기술 리서치, 모델 설계 및 훈련</div>

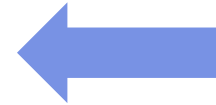
# Sprint Planning

Story	To Do	In progress	Done
<div>애플리케이션 UI/UX 설계 및 개발</div>	<div>모듈 통합: 개발한 모듈이 정상적으로 어플리케이션 상에서 작동 할 수 있도록 모듈 통합</div>	<div>인터랙션 구현: 음성 녹음, 재생 및 사용자 피드백 메커니즘 통합</div> <div>프론트엔드 개발: React Native를 사용한 모바일 어플리케이션 개발</div>	<div>그래픽 시각화 구현: 음성 파동을 시각적으로 표현하는 기능 개발</div> <div>UI/UX 디자인: 와이어프레임 및 목업 디자인</div>
<div>LLM 구성 및 개발</div>	<div>고급 기능 및 통합 테스트: 사용자 인터페이스 개선 및 추가적인 LLM 기능 통합</div>	<div>데이터 통합 및 학습: 데이터 전처리 및 정제, LLM의 파인 튜닝을 위한 학습 파이프라인 구축</div>	<div>기초 구성 : 기본 대화 처리 기능 구현 및 간단한 사용자 인터페이스 구축</div>

# Demo of Deliverables

방언-표준어 데이터  
매핑 작업

	A	B
1	dialect	standard
2	들통시	뒷간
3	등짐재이	등짐장이
4	등병	웅덩이
5	등북	웅덩이
6	등서리	벼랑
7	등짐쟁이	등짐장이
8	등골뻘	등골뻘
9	등껍데기	등껍질
10	등껍디	등껍질
11	등공	등
12	등날뻘이	등성이
13	등다리	등
14	등덜뻘이	등때기
15	등도리	등
16	등때	발두둑
17	등말	등목
18	등바니	살강
19	등보리	두둑보리
20	등빨	등판
21	등빨	덩치
22	등뻘	등뻘



'경상남도 방언(명사).csv'의 일부

형용사            1,117 단어  
 부사            1,050 단어  
 명사 및 동사   13,738 단어  
 어미            36 단어

도합 15,941 단어 (약 16,000 단어)

매핑 및 csv 파일로의 변환 완료

# Demo of Deliverables

매핑된 csv 파일 기반  
번역 프로그램

The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script named '기초구현(최종).py'. The script includes functions for uploading files to Google Cloud Storage (GCS), transcribing audio files, and converting dialect text to standard text. A test code block at the bottom demonstrates the usage of the 'process\_audio\_file' function with a specific audio file path.

```
92     f.write(text)
93
94     # 저장된 파일을 GCS에 업로드
95     upload_to_gcs(TRANSLATED_TEXT_BUCKET_NAME, file_path, file_name)
96
97     # 해당 함수는 파일 경로를 입력받아 음성을 텍스트로 변환하고 방언을 표준어로 변환
98     def process_audio_file(file_path):
99         # WAV 파일의 샘플 레이트 확인
100         sample_rate = get_sample_rate(file_path)
101
102         # GCS에 파일 업로드
103         gcs_uri = upload_to_gcs(GCS_BUCKET_NAME, file_path, os.path.basename(file_path))
104
105         # 음성을 텍스트로 변환 (비중거 방식 사용)
106         transcribed_text = transcribe_gcs_long_running(gcs_uri, sample_rate)
107
108         # 방언을 표준어로 변환
109         converted_text = convert_dialect_to_standard(transcribed_text)
110
111         # 변환된 텍스트를 파일로 저장 및 GCS에 업로드
112         save_translated_text(converted_text, f"{os.path.basename(file_path)}[:-4]}_translated.txt")
113
114         return {
115             'transcribed_text': transcribed_text,
116             'converted_text': converted_text
117         }
118
119     # 테스트 코드
120     if __name__ == "__main__":
121         # 테스트 파일 경로
122         test_file_path = r"C:\Users\jiyun\Desktop\2024\2024 1st semester\capstone design\dialect_data\방언1.wav"
123         # 오디오 파일 처리
124         result = process_audio_file(test_file_path)
125
126         # 결과 출력
127         print("Transcribed Text: ", result['transcribed_text'])
128         print("Converted Text: ", result['converted_text'])
129
```

The right sidebar contains a 'Usage' section with instructions on how to get help and a 'Console' window showing the execution of the script. The console output indicates that the script is running successfully and provides the transcribed and converted text.

Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Help Variable Explorer Files

Console 2/4 X

IPython 8.20.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/jiyun/Desktop/2024/2024 1st semester/capstone design/untitled4.py', wdir='C:/Users/jiyun/Desktop/2024/2024 1st semester/capstone design')

Waiting for operation to complete...

Operation not completed yet. Sleeping for 30 seconds...

Transcribed Text: 여기 친구지찌집 한 겹시 주이소

Converted Text: 여기 부추헌 한 겹시 주세요

In [2]:

Python Console History

conda: base (Python 3.11.5) Completions: conda/base LSP: Python Line 124, Col 48 UTF-8 CRLF RW Mem 55%

# Demo of Deliverables

Text에서  
방언 검출 후  
표준어 번역

매핑 적용 번역

- Dialect Text: 여기 정구지 찌짐 한 접시 주이소
- Standard Text: 여기 부추 전 한 접시 주세요



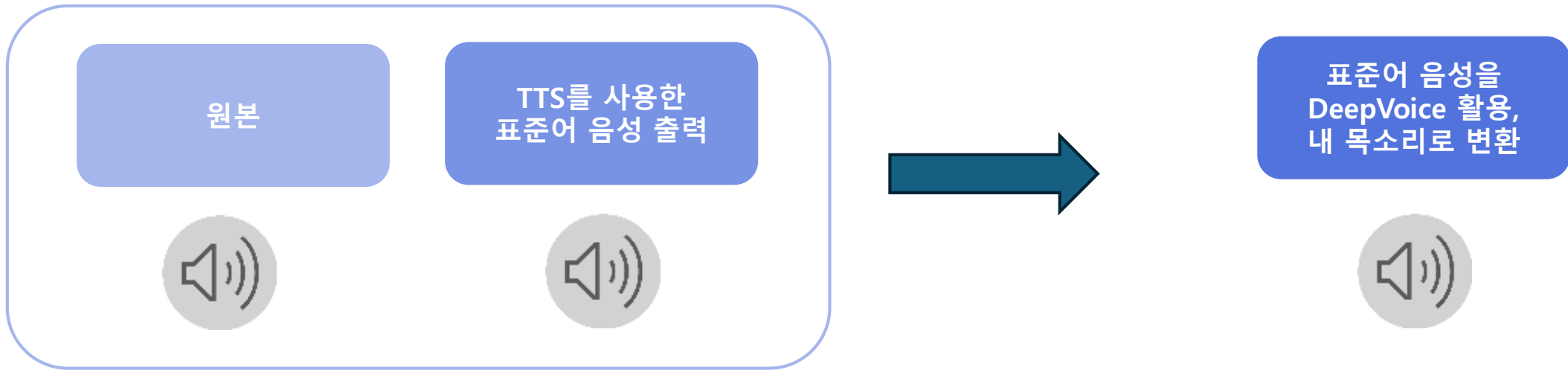
교정된 문장에  
대한 아나운서 tts  
다운로드 자동화

1. 로그인 페이지 열기
2. 자동 로그인
3. 클로바 더빙 페이지로 이동
4. 입력 필드에 교정된 문장  
자동 입력
5. 저장 버튼 클릭
6. 다운로드



# Demo of Deliverables

---

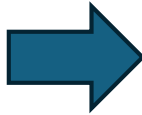




# Demo of Deliverables

## LLM을 활용한 대화 상황 유도

```
1 class ChatBot():
2     def __init__(self, model='gpt-3.5-turbo'):
3         self.model = model
4         self.messages = []
5
6     def ask(self, question):
7         self.messages.append({
8             'role': 'user',
9             'content': question
10        })
11        res = self.__ask__()
12        return res
13
14    def __ask__(self):
15        completion = openai.ChatCompletion.create(
16            # model 지정
17            model=self.model,
18            messages=self.messages
19        )
20        response = completion.choices[0].message['content']
21        self.messages.append({
22            'role': 'assistant',
23            'content': response
24        })
25        return response
26
27    def show_messages(self):
28        return self.messages
29
30    def clear(self):
31        self.messages.clear()
```

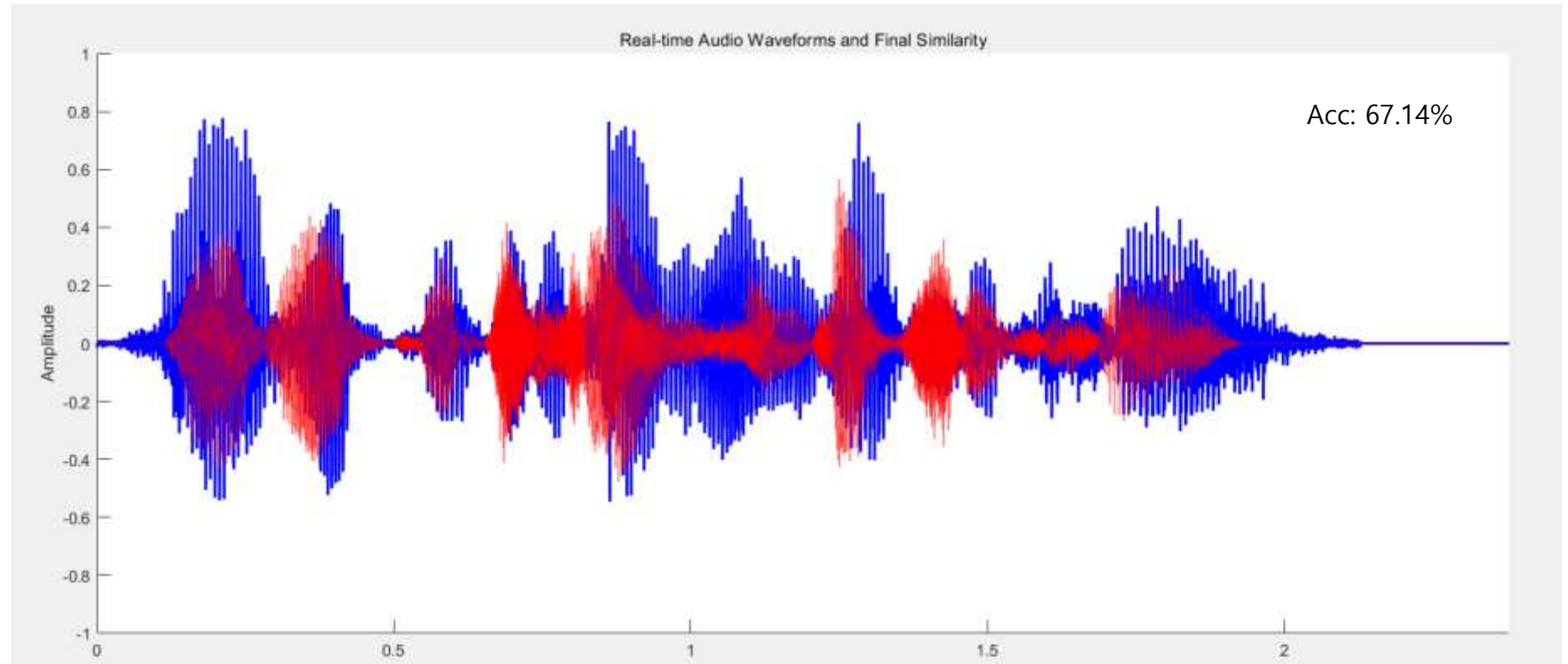


User: 너의 이름은 뭐야?  
Bot: 저의 이름은 A.I. Assistant입니다. 어떤 일을 도와드릴까요?  
User: 오늘 저녁 주전해줘  
Bot: 무슨 음식을 좋아하시나요? 기호에 따라 추천해드릴게요!  
User: 중식  
Bot: 중식 추천을 드린다면, "광장시장"에 있는 "진미평양냉면"의 "쟁반짜장"을 추천드립니다.

# Demo of Deliverables

Waveform을 통한  
나의 발화와 표준어  
사이의 간극  
확인 및 학습

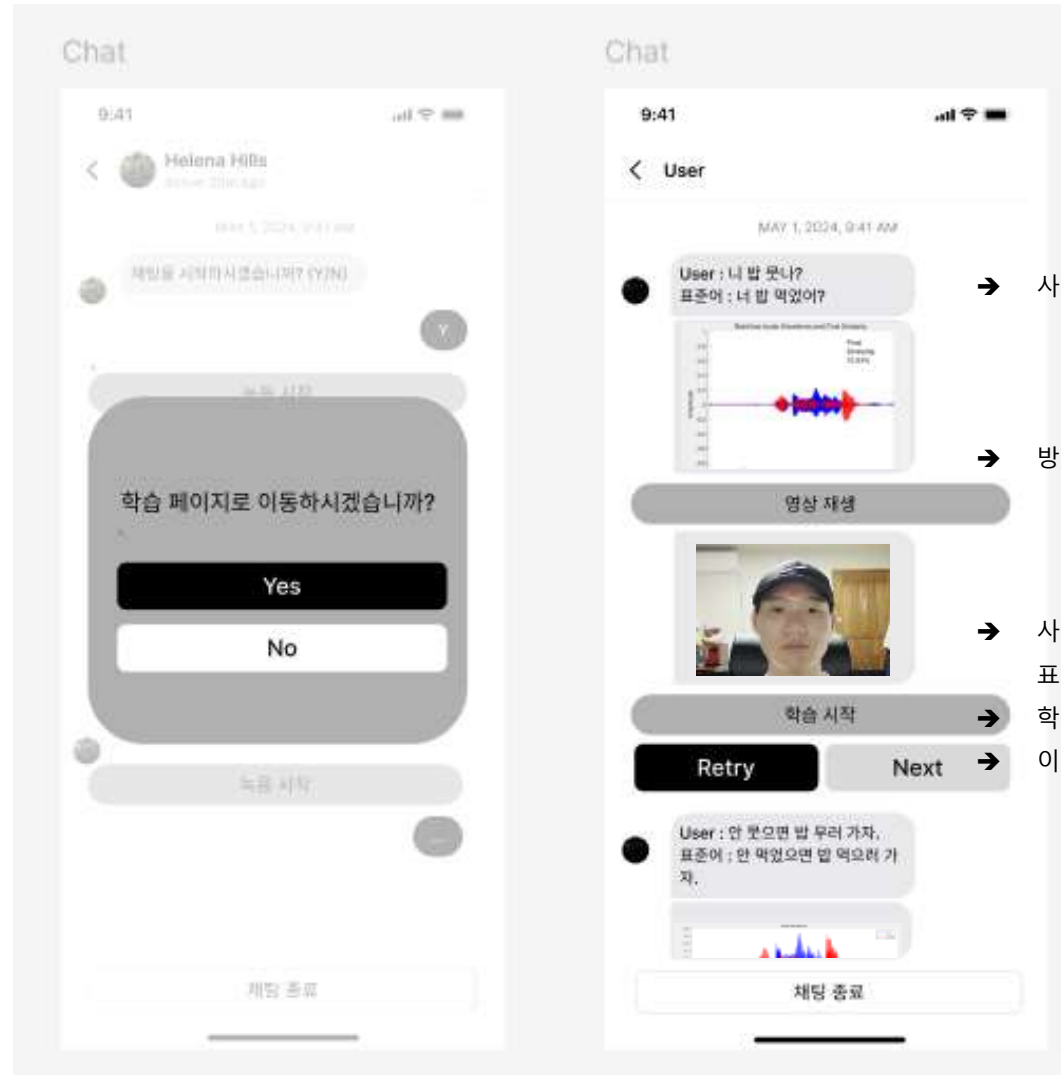
— 방언 음성  
— 표준어 음성



# Demo of Deliverables

## UI/UX 디자인

- ➔ 학습 페이지로 사용자 이동
- ➔ 사용자가 대화 중 사용한 방언을 순서대로 한 문장씩 학습
- ➔ 사용자는 반복 학습 또는 다음 문장 학습으로 넘어갈 수 있음



➔ 사용자의 방언 텍스트와 표준어로 번역한 텍스트 표시

➔ 방언 발화와 표준어 발화의 웨이브폼 및 일치율 표시

➔ 사용자의 얼굴에 표준어 입모양 디페이크를 적용한 영상 표시

➔ 학습 시작 버튼을 누르면 해당 문장을 재 녹음

➔ 이후 반복 학습 또는 다음 문장 학습 버튼 표시

# Demo of Deliverables

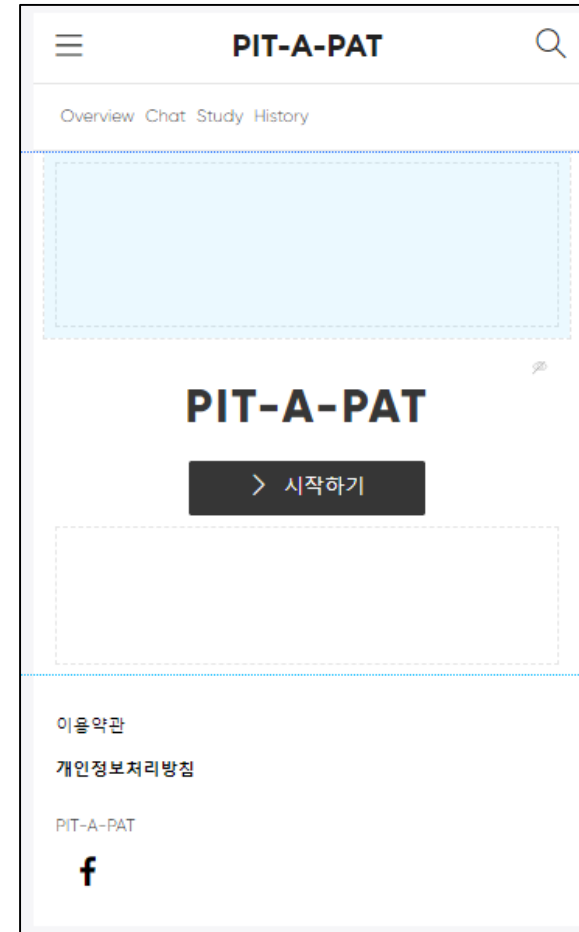
웹 서버 구축 및 디자인

PC ver.

<https://pitapatto.imweb.me/>



Mobile ver.



# Challenges and Solutions

---

## 문제점 및 해결책

기존 딥보이스 모델이 오래되어 모듈 간 충돌 발생,  
내 목소리를 정확하게 구현하는 데 어려움

-> Coqui-ai 모델을 학습하여 딥보이스 구현의 정확성과 일관성 향상

# Sprint Retrospective

---

## 개선점

### 다양한 기술의 결합: 개별 구현의 성과와 통합의 과제

- 딥보이스, 딥페이스, UI/UX 디자인 등 다양한 기술들이 개별적으로 구현됨.
- 기술을 유기적으로 결합하여 통합된 시스템으로 만드는 데에는 시간적 어려움이 있었음.
  - 전체적인 사용자 경험과 시스템의 완성도를 향상시킬 필요 있음.

# Next Sprint Goals

---

## 목표

- 딥페이크 기술을 활용한 아바타 생성으로 LLM모델과의 단조로운 대화가 아닌 실생활에 더욱 근접한 대화 형식 및 추구
- STT를 사용하여 방언을 사투리로 번역하는 과정에서의 딜레이 감소
- 웹 및 어플리케이션으로 모듈 통합 및 테스트 수행
- 웹 및 어플리케이션 UI/UX 개선
- 경상도 뿐만 아닌 전국의 방언 데이터 수집 및 맵핑
- 방언 교정 뿐만 아니라 표준어 사용자가 방언 사용을 필요로 할 경우, 방언을 학습 할 수 있도록 개선

# Sprint Retrospective

권주연

음성 변환의 자연스러움을 위해 노력. 다양한 지표와 사용자 피드백을 통해 지속적으로 개선. 특히 방언 음성을 표준어 아나운서 억양에 맞춰 변환하는 것에 집중. 또한, 웹 페이지 자동화를 통한 TTS 시스템 통합 작업도 수행하여 시스템의 효율성을 향상시킴.

김지윤

여러 출처에서 수집한 방언 및 표준어 데이터를 전처리하여 분석 가능한 형태로 변환. 특히, 수집된 방언 데이터를 표준어와 매핑하여 데이터베이스를 구축하고, 각 발화의 대응 관계를 체계적으로 기록하고 관리. 이러한 작업을 통해 데이터의 품질을 높이고, 모델 개발을 위한 기반 마련.

이상승

GPU 클러스터를 활용해 모델을 학습시키고, 자연스러운 음성 변환을 목표로 다양한 지표를 통해 성능을 평가하고 개선. 발음에 따른 입모양 변환 모델을 설계하여 실시간 변환을 구현.

한은영

음성 데이터의 주파수 특성 분석 및 시각화를 통해 딥페이크와 딥보이스 모델을 통합하는 방법을 배움. 사용자 친화적인 어플리케이션을 설계하고 개발하며, 실시간 처리 기능과 효율적인 UI/UX를 구현. 또한, 앱 성능 테스트와 디버깅을 통해 안정성과 효율성을 확보하는 경험을 쌓음.



권주연

- 학습 데이터 다양성을 높여, 다양한 방언과 억양을 더 잘 반영할 필요 O
- 사용자 피드백 수집 과정을 체계화하여 더 빠르고 정확하게 반영할 수 있는 시스템을 구축

김지윤

- 방언 데이터의 매핑량 증가 ~> 방언 교정의 퀄리티 향상
- 실행시간 단축 ~> 번역 모듈 최적화

이상승

- 성능 평가 지표의 다양성을 확대, 모델의 세부적인 개선 포인트를 더 명확히
- 실시간 변환 과정에서 발생하는 지연 시간 최소화를 위한 최적화 작업

한은영

- 주파수 특성 분석의 정밀도를 높여 더 정확한 모델 통합을 도모
- 어플리케이션의 사용자 피드백 루프를 강화하여 UI/UX 개선을 지속적으로 반영
- 성능 테스트를 더욱 체계화, 다양한 환경에서의 디버깅을 통한 안정성 향상