

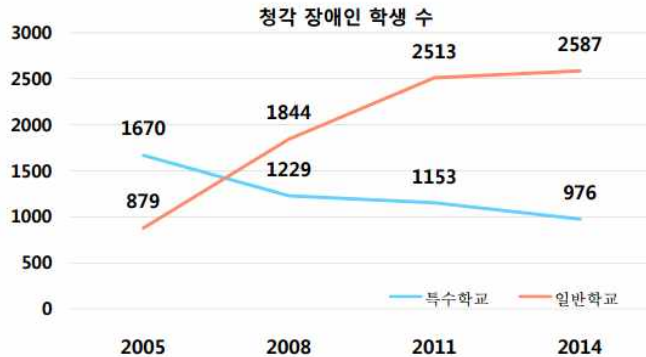
종합설계프로젝트: 결과보고서

| | | | | | |
|----------------------------|---|---------------|----|-------|-------------------------------|
| 과 제 명 | Talk to me : 음성인식을 이용한 실시간 인물 구분 타이핑 프로그램 | | | | |
| 기 간 | 2016.03.02 ~ 2016.06.20 | | | | |
| 팀 대 표 | 소속 | 컴퓨터학부 | | 성명 | 박소은 |
| | 연락처 | 010-3006-8495 | | Email | tdfq3578@naver.com |
| 과 제 참 여 학 생 | 소속(학과) | 학번 | 학년 | 성명 | 담당업무 |
| | 컴퓨터학부 | 2012105037 | 4 | 박소은 | 프로젝트 진행관리, 아날로그 음성 변환/입력 |
| | 컴퓨터학부 | 2012105028 | 4 | 김한별 | 프로그램 구현지휘, 알고리즘 분석/테스트 |
| | 컴퓨터학부 | 2012105040 | 4 | 박지인 | 문서 수집/제작/정리 타이핑 속도 조율 |
| | 컴퓨터학부 | 2012105057 | 4 | 송은영 | 예산 수립/신청/관리 화자 분류 결과 출력/전송 |
| 주요 수행내용 (결과) | <ul style="list-style-type: none"> 음성특징 학습 고속 푸리에 변환 & MFCC 학습 머신러닝 알고리즘 GMM(Gaussian Mixture Models) 학습 오픈 라이브러리 'LIUM speaker diarization'를 사용해 화자인식 네이버 음성인식 라이브러리를 이용해 음성 출력 | | | | |
| 기대효과 | <ul style="list-style-type: none"> 청각장애인의 자유로운 수업 수강 가능 청각장애인과 비장애인의 자유로운 대화 가능 회의 시 회의록 작성에 용이 주인으로 등록된 사람의 말만 듣는 기기 개발 가능 범죄사건 수사에 활용 가능 보안의 용도로 사용 가능 | | | | |

1. 과제 수행 배경 및 목적

1) 과제 수행 배경

가. 청각 장애인 수에 맞추지 못한 현행 학생 지원 서비스의 문제점



| 연도 | 특수학교 | 일반학교 | 일반학교 내 특수학급 | 일반학교 내 일반학급 | 총 |
|------|------|------|-------------|-------------|------|
| 2005 | 1670 | 879 | 494 | 385 | 2549 |
| 2008 | 1229 | 1844 | 795 | 1049 | 3073 |
| 2011 | 1153 | 2513 | 916 | 1597 | 3666 |
| 2014 | 976 | 2587 | 808 | 1779 | 3563 |

| 연도 | 특수학교 | 일반학교 | 일반학교 내 특수학급 | 일반학교 내 일반학급 | 총 |
|------|------|------|-------------|-------------|------|
| 2005 | 68% | 34% | 19% | 15% | 100% |
| 2008 | 40% | 60% | 26% | 34% | 100% |
| 2011 | 31% | 69% | 25% | 44% | 100% |
| 2014 | 27% | 73% | 23% | 50% | 100% |

특수학교 및 일반학교 청각장애인 지원 현황

장애학생이 수업과 학습을 하는 데 도움이 필요한 경우 대필, 실험보조, 학습보조와 같은 도움을 제공하기 위해 도우미제도를 운영하고 있습니다.

대필 : 수업이나 시험 시 필기가 어려운 학생을 위한 도움 제공
 실험보조 : 원활한 실험을 위해 도우미가 실험에 참여하여 장애학생을 보조
 학습보조(튜터) : 보충, 심화학습 지원

대학교 청각장애인 지원 현황

- 교내 학습도우미 지원 및 속기사 지원 등의 서비스가 미흡하다.
- 청각장애인의 평등한 학습권 보장이 어려운 상황이다.
- 청각장애인과 비장애인 학생 사이의 의사소통이 어려운 상황이다.

나. 유사 목적 서비스의 한계



유사한 목적을 가지고 시행되고 있는 '쉐어타이핑'

- 속기사와 청각장애인의 시간이 맞지 않으면 서비스를 이용할 수 없다.
- 청각장애인과 비장애인 사이의 의사소통에 도움이 되지 못한다.

다. 인물 구분 음성 인식 분야에 대한 도전의 필요성

- 인물 구분 음성 인식으로 주인의 말만을 알아듣는 기기를 개발할 수 있다.
- 인물 구분 음성 인식으로 범죄사건 수사에 도움이 될 수 있다.
- 보안의 용도로 인물 구분 음성 인식 기술이 쓰일 수 있다.

라. 머신러닝 알고리즘 분야에 대한 도전의 필요성

- 고도로 학습된 인공지능 개발에 대한 관심이 높아지고 있다.
- 고도의 연산 결과를 추출해낼 수 있다.

2) 과제 수행 목적

가. 청각장애인의 자유로운 수업 수강

- 실시간 타이핑 프로그램으로, 강연자의 말을 실시간으로 텍스트로 받아볼 수 있다.
- 인물 구분 음성 인식으로, 정해진 인물인 강연자의 말만을 출력한다.

나. 청각장애인과 비장애인과 자유로운 대화

- 실시간 타이핑 프로그램으로, 여러 사람들의 말을 실시간으로 텍스트로 받아볼 수 있다.
- 인물 구분 음성 인식으로, 대화에 참여하는 사람들의 말을 화자별로 구분한다.
- 대화를 이해하기 쉽도록, 화자의 이름과 말한 내용을 채팅창 형식으로 출력한다.

다. 아날로그 음성의 디지털화

- 잡음 속에서도 사람의 목소리인 아날로그 음성을 정확하게 받아들이는다.
- 받아들이는 아날로그 음성의 특성을 반영하여 디지털화한다.
- 적절한 자료 처리와 저장 형식을 찾는다.

라. 인물 구분 음성 분석과 학습

- 음성 속에서 인물 구분에 도움이 되는 목소리의 특성 이론을 학습한다.
- 음성을 분석하여 인물 구분에 영향을 미치는 특성을 추출한다.

마. 머신러닝 학습을 통한 음성 구분

- 화자 식별에 도움이 되는 머신러닝 알고리즘을 학습한다.
- 머신러닝 알고리즘을 활용하여, 음성의 주인을 학습하도록 구현한다.
- 음성의 특성별로 가중치를 다르게 두어, 정확한 인물 구분이 가능하도록 구현한다.

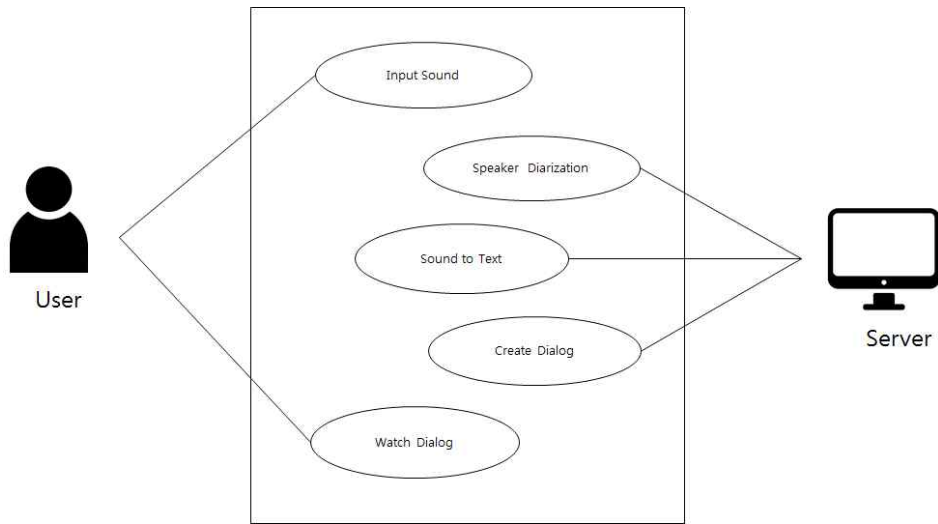
바. 음성을 텍스트로 전환

- 음성의 주인의 이름과 그 내용이 함께 화면에 출력된다.
- 음성의 주인과 다른 음성을 받아들이면 출력되지 않는다.



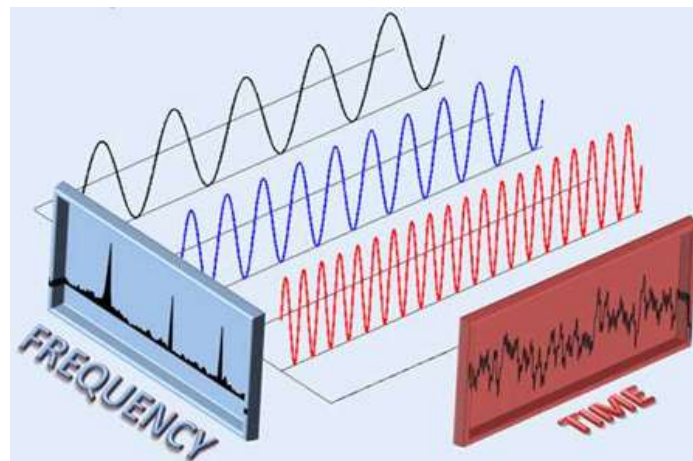
2. 과제 수행 내용

1) Use Case Diagram



사용자는 소리를 입력하고 대화창을 볼 수 있다. 서버는 입력된 소리의 화자를 인식하고 텍스트로 변환하여 대화창에 뿌려준다.

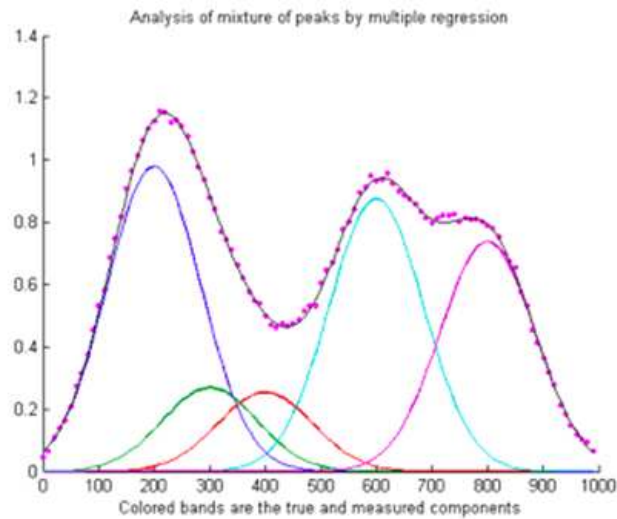
2) MFCC



<푸리에 변환>

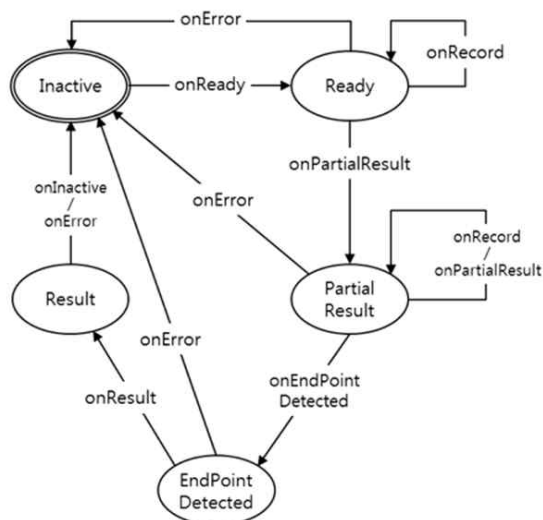
소리는 여러 주파수의 합으로 이루어져있다. 시간영역의 신호를 여러 주파수의 합으로 나타내기 위해 푸리에변환을 사용한다. 푸리에 변환을 빠르게 하기 위해 모든 신호를 변환시키지 않고 샘플링 한 후 변환하는 것이 고속 푸리에 변환이다. 이 스펙트럼의 크기를 mel-scale 필터를 거치고 로그를 취하는 등 몇 가지 처리를 더 거쳐 얻어낸 MFCC값을 음성인식을 위한 의미 있는 파라메타로 사용한다.

3) GMM & UBM



평균을 중심으로 뭉쳐있는 가우시안 확률분포를 여러개 합하여 만들어지는 모델이다. unsupervised learning에 속한다. em알고리즘으로 파라메타를 찾아 데이터가 어느 가우시안에 속할지 추정한다. UBM은 불특정 다수의 화자를 나타내는 큰 GMM으로 화자 종속적 특징 데이터를 이용해 MAP적응을 수행하여 화자모델을 트레이닝하는 단계를 거치게 된다. 이것을 이용해 새로운 목소리가 들어왔을 때, 기존의 가우시안에 속할 확률이 일정치 이상이 되면 그 목소리의 주인으로 인식되고 일정수치에 미치지 못하면 타인으로 인식된다.

4) Naver open api



네이버 음성인식 open api를 사용해서 안드로이드로 입력된 소리를 text로 변환한다. 사진은 open api의 state diagram이다.

3. 과제 수행 내역

1) 3월 2일 활동 일지

가. 목표

- 팀 편성 이후, 자기소개 시간가지기
- 주제에 대한 아이디어 논의하기

나. 회의록

- 핀테크에 대한 주제를 잡기로 결정했습니다.
- 웨어러블, 주식예측, 버스요금 지불, 소비패턴분석 등의 키워드가 나왔습니다.

다. 진행 사항

- 주제에 대한 큰 틀을 잡았습니다.
- 큰 주제인 핀테크와 관련된 아이디어들을 각자 더 생각해보기로 했습니다.



2) 3월 6일 활동 일지

가. 목표

- 팀원 변동 이후, 주제 틀 변경하여 확정하기

나. 회의록

- IoT, 어플로 제어하는 멀티탭
- 아두이노를 이용한 여행가방
- VR을 활용한 길 찾기
- 청각장애인을 위한 음성인식 IoT
- 지도 기반 버킷리스트 작성, 위치 기반 알람

다. 진행 사항

- 장애인과 비장애인을 위한 실시간 대화형 속기 프로그램으로 주제를 확정했습니다.

3) 3월 7일 활동 일지

가. 목표

- 주제 발표 준비하기

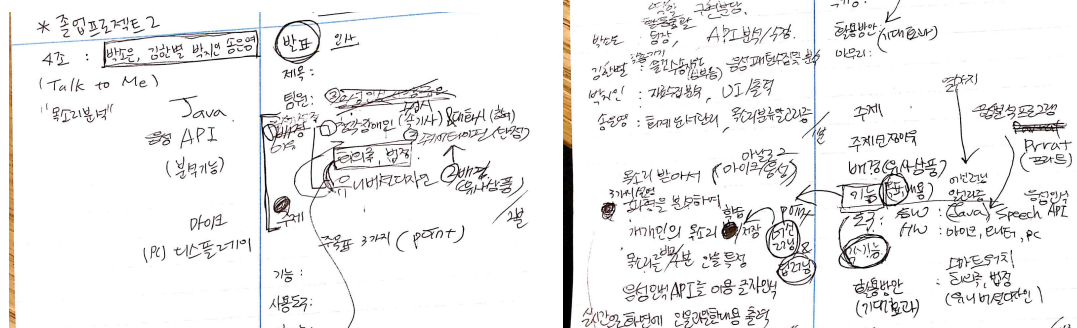
- 확정된 주제 구체화하기

나. 회의록

- 주제 발표와 관련하여 역할 분담을 하였습니다.
- 주제와 비슷한 목적을 가진 유사 서비스를 찾아보았습니다.
- 주제에 대한 회의록 작성이나 스마트 워치관련 개발과 같은 활용방안을 제시했습니다.
- 목소리를 받아와 분석하자는 방향을 잡았습니다.
- 목소리를 구분하는 데에 머신러닝을 사용하여 학습하자는 의견이 나왔습니다.
- 음성인식 API를 이용하여 음성을 텍스트로 변환하기로 결정했습니다.
- 실시간으로 인물의 이름과 말한 내용을 출력하도록 구현하기로 했습니다.
- 프로젝트를 위해 마이크와 PC가 필요하며, JAVA를 사용하여 구현할 것을 예상했습니다.

다. 진행 사항

- 주제 발표에 대한 내용 및 계획을 세우고 역할을 분담했습니다.
- 구체적인 주제 관련 내용들을 정리했습니다.



4) 3월 12일 활동 일지

가. 목표

- 수행계획서 작성에 대해 부분별로 역할 분담하기
- 수행 계획 발표 준비하기

나. 회의록

- 수행계획서 내용은 박소은 학우와 박지인 학우가 맡게 되었습니다.
- 수행 계획 발표 자료는 송은영 학우가 제작하게 되었습니다.
- 프로젝트의 전반적인 탐색과 테스트는 김한별 학우가 맡게 되었습니다.
- JAVA를 사용하여 Web에서 구동시키려 했으나 알맞은 음성인식 API를 찾지 못했습니다.
- 네이버 음성인식 API를 사용하기위하여 안드로이드를 사용하기로 결정했습니다.

다. 진행 사항

- 수행계획서 작성과 수행 계획 발표, 수행 도구 탐색에 관련된 역할을 분담했습니다.
- 프로젝트의 음성인식 API와 관련된 도구를 결정하였습니다.

5) 3월 22일 활동 일지

가. 목표

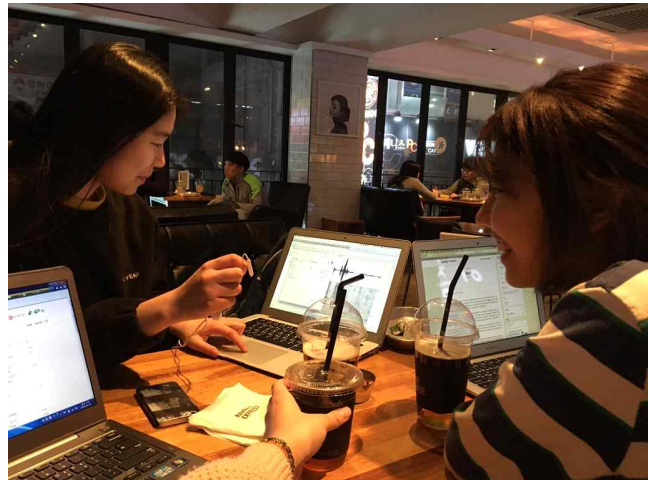
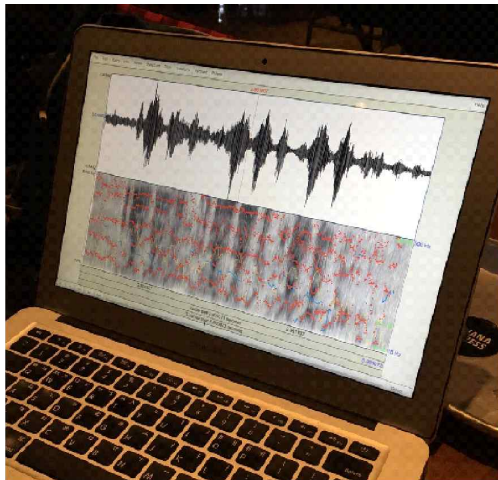
- Praat를 이용하여 목소리의 특성을 관찰하기
- 인물 구분을 위한 음성인식에 필요한 목소리 특성 분석 및 학습하기

나. 회의록

- 인물 구분 음성 인식 분야가 화자 식별(Speaker Diarization)로 불리는 것을 알게 되었습니다.
- Praat를 이용하여 목소리의 특성을 관찰해보며, formant가 목소리 구분에 가장 가능성이 있는 특성임을 예측해보았습니다.
- Praat를 이용하여 추출된 Shimmer와 Jitter 값을 이해해보는 것이 좋겠다는 의견이 나왔습니다.
- Praat만을 사용하여 목소리를 분석하는 데에 어려움이 있다고 판단하여, 그와 관련된 이론들을 따로 더 학습해보자고 결정했습니다.

다. 진행 사항

- Praat를 이용하여 목소리의 특성을 관찰하고 분석, 학습해보았습니다.



6) 3월 27일 활동 일지

가. 목표

- 고속 푸리에 변환에 대해 학습하기
- 파이썬 프로그램을 이용하여 고속 푸리에 변환 실습해보기
- 실습해서 나온 출력 분석해보기

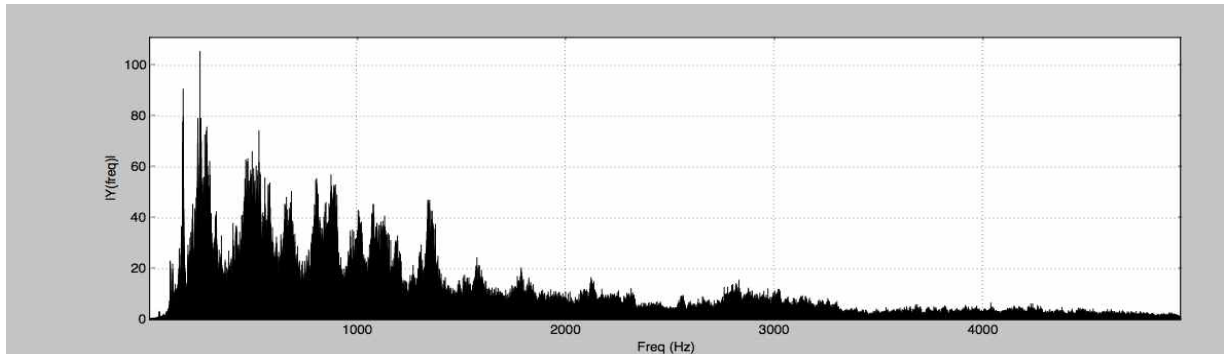
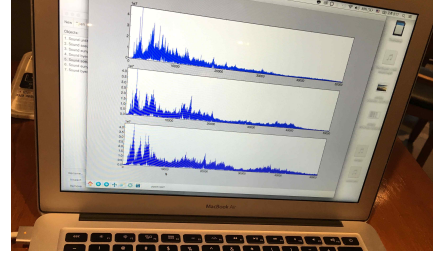
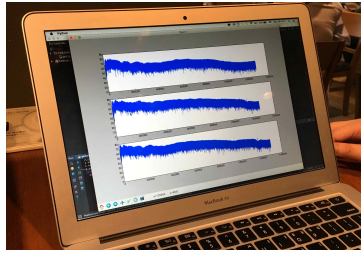
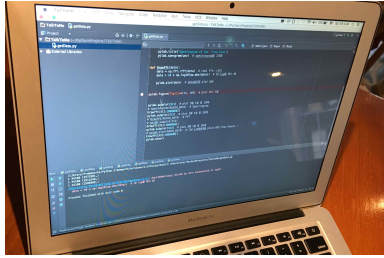
나. 회의록

- 다양한 음성을 녹음하여 고속 푸리에 변환 그래프를 그려보고 비교해보았습니다.
- 인물별로 고속 푸리에 변환 그래프에서의 차이점을 발견할 수 있었습니다.
- 고속 푸리에 변환으로 얻은 그래프의 x축과 y축을 이해하였습니다.
- 고속 푸리에 변환으로 얻은 그래프에 몇 가지 처리를 더해 MFCC 값을 얻어 보았습니다.

다. 진행 사항

- 고속 푸리에 변환에 대해 이해하였습니다.

- 고속 푸리에 변환이 인물 구분 음성 인식에 필요하다는 것을 확정했습니다.



7) 3월 29일 활동 일지

가. 목표

- 음성 분석과 관련된 이론 학습하기
- 화자 식별과 관련된 논문을 읽고 사용할 변수 값들을 알아보기

나. 회의록

- 화자 인식, 화자 식별, 화자 종속 음성 인식과 관련된 논문들을 찾아보았습니다.
- 경북대에서 논문 열람비용을 지원해주는 DBpia를 이용하여 논문을 열람하였습니다.
- '신경회로망에 의한 남녀 화자의 성별 인식 알고리즘'이라는 논문을 읽고, 음성 특징 벡터 4가지를 살펴보았습니다.
- '음성특징벡터 및 정규화 인식방법을 이용한 화자종속 음성인식'이라는 논문의 내용이 저희 팀의 주제와 유사하다는 것을 발견했습니다.
- 살펴본 논문들에서 발견되는 LPC와 CMS 등에 대해 이해해볼 필요가 있다고 판단했습니다.

다. 진행 사항

- 화자 식별 음성 인식 시, 필요하다고 생각되는 특징 값들을 알아보고 학습해보았습니다.



8) 4월 2일 활동 일지

가. 목표

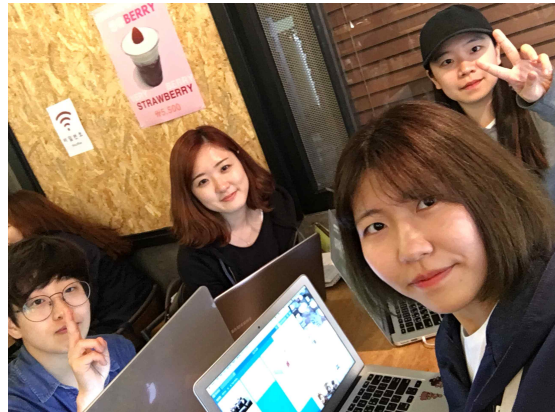
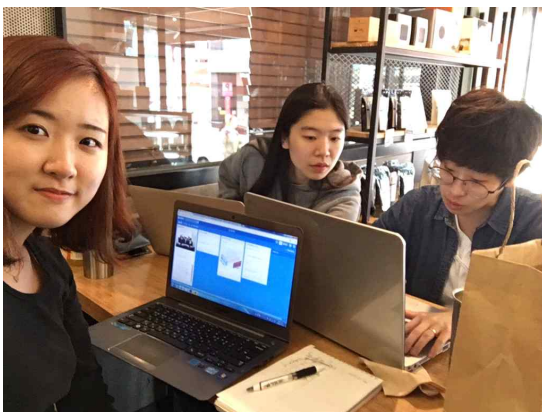
- 머신러닝 알고리즘을 학습하기
- 인물 구분 음성인식에 적합한 머신러닝 알고리즘 결정하기

나. 회의록

- 김한별 학우와 송은영 학우가 비지도 방식 Neural Network를 연구하는 역할을 맡았습니다.
- 박소은 학우와 박지인 학우가 음성을 Input Data로 변환하는 내용을 연구하는 역할을 맡았습니다.
- Supervised learning과 Unsupervised learning을 학습하여 이해해보았습니다.
- Unsupervised learning에서 hebbian learning rule을 이해해보며, 저희 팀의 주제를 구현하는 데에 이용할 수 있을 것이라고 예상해보았습니다.
- LIUM Speaker Diarization 오픈 라이브러리를 활용해보자는 의견이 나왔습니다.
- 위의 오픈 라이브러리를 사용한 VoicelD를 분석하여 사용해보기로 결정했습니다.

다. 진행 사항

- 인물 구분 음성인식에 적합한 머신러닝을 학습하였습니다.
- 구현해 사용해볼만한 머신러닝 알고리즘 기반 API를 찾았습니다.



9) 4월 25일 활동 일지

가. 목표

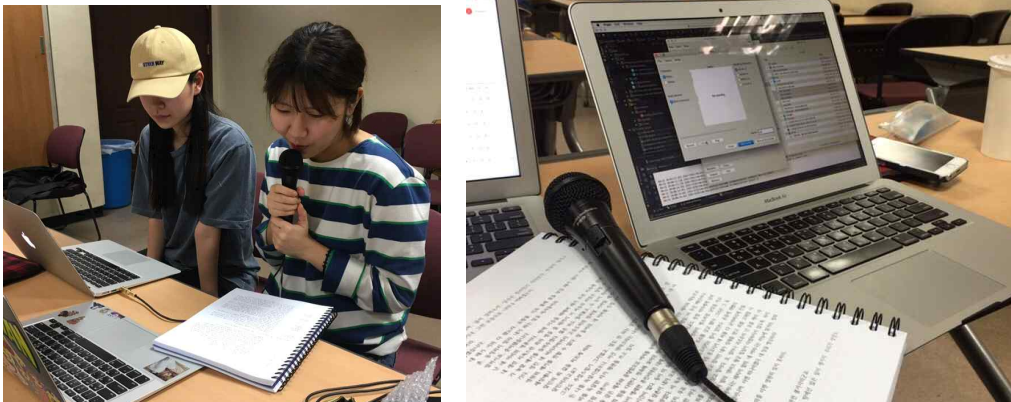
- 인물 구분 음성 분석 프로그램 테스트 작업하기

나. 회의록

- 지원받은 마이크를 활용하여, 인물 구분 음성 분석 프로그램을 테스트해보았습니다.
- 연극에 사용되는 대본을 활용하여 대화하듯이 음성을 녹음하여 사용하였습니다.
- 낮은 속도와 높지 않은 정확도였지만, 인물 구분 음성 분석 프로그램이 잘 실행되었습니다.
- 인물 구분 음성 분석 프로그램의 문제점과 해결 방안을 토의해보았습니다.
- 안드로이드 기기에서 네이버 음성인식 API를 사용할 수 있도록 구현하는 계획을 세웠습니다.

다. 진행 사항

- 인물 구분 음성 분석 프로그램을 테스트하고 보완 사항을 논의해보았습니다.
- 프로그램을 서버에서 구동하고 안드로이드 기기와 통신하도록 구현 계획을 세웠습니다.



10) 4월 26일 활동 일지

가. 목표

- 중간발표에 대한 준비하기
- 프로그램의 인물 구분관련 기능 설정하기
- 프로그램을 서버에서 구동하고 안드로이드 기기와 통신 테스트하기

나. 회의록

- 인물 구분 기능의 최적화를 위하여 여러 가지 트레이닝 셋을 사용해보았습니다.
- 좀 더 나아진 프로그램을 바탕으로 발표 내용을 완성했습니다.
- 중간발표에 대한 발표 자료를 제작하였습니다.
- 안드로이드에서 음성을 받아와 서버로 보내는 과정을 연구했습니다.
- 서버에서 받은 음성을 인물 구분 음성 인식 프로그램의 Input으로 사용해보았습니다.
- 서버에서 분석한 출력물을 안드로이드 기기로 전송해보았습니다.

다. 진행 사항

- 인물 구분 음성 인식 프로그램을 개선하여 중간발표를 준비했습니다.
- 프로그램과 안드로이드 기기간의 통신을 개선했습니다.



11) 5월 5일 활동 일지

가. 목표

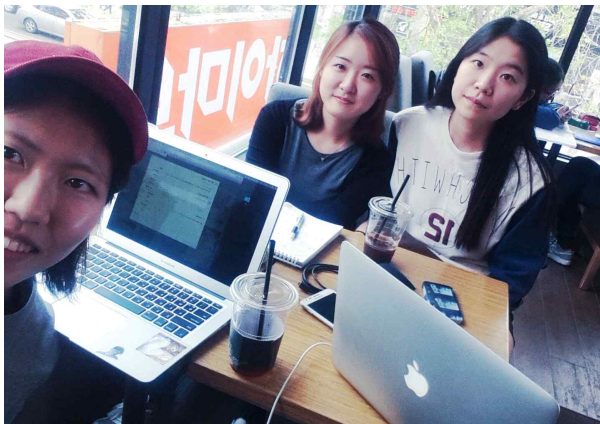
- 프로젝트의 현황 점검하기
- 당면한 문제점에 대한 해결책 논의하기

나. 회의록

- 너무 느린 프로그램의 문제점을 해결할 의견을 모아보았습니다.
- 새로운 주제 실현 접근 방식을 제안했습니다.
- 하나의 휴대폰에서 모든 사람의 목소리를 판별하기 때문에 속도를 크게 향상시킬 수 없다는 결론을 지었습니다.
- 여러 휴대폰을 두고, 각각의 휴대폰이 휴대폰 주인의 목소리만을 판별하여 화자의 이름과 말한 내용을 대화목록에 출력하여 속도를 향상시키자는 의견을 수용하였습니다.
- VoiceID를 사용하는 것이 속도를 저해할 수 있다는 판단을 내렸습니다.
- LIUM Speaker Diarization 오픈 라이브러리를 분석하여 다시 프로그램을 제작하기로 결정했습니다.

다. 진행 사항

- 주제 실현의 접근 방식을 변경하였습니다.
- 프로그램을 새로 제작하기로 결정하였습니다.



12) 5월 16일 활동 일지

가. 목표

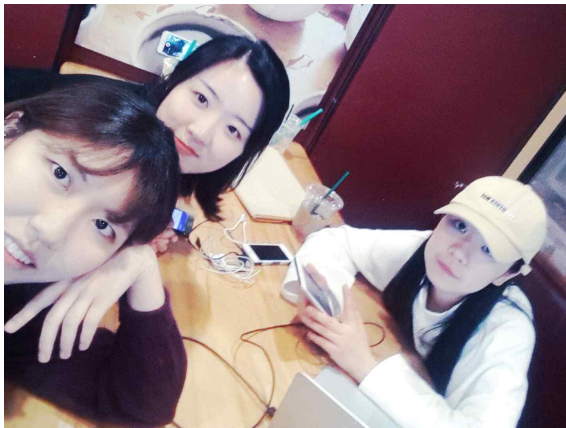
- 중간발표를 위한 시연영상 제작하기
- 중간발표를 위한 자료를 제작하기

나. 회의록

- 변경된 인물 구분 음성인식 프로그램의 구현 방법을 중간발표 내용에 표현하였습니다.
- 주제 실현을 위해 변경한 방법으로 시연영상을 촬영하였습니다.
- 변경된 인물 구분 음성인식 프로그램과 안드로이드 기기의 통신에 JAVA 서버를 이용하였습니다.

다. 진행 사항

- 변경된 프로그램을 소개하는 중간발표를 위한 자료와 시연영상을 제작했습니다.



13) 5월 22일 활동 일지

가. 목표

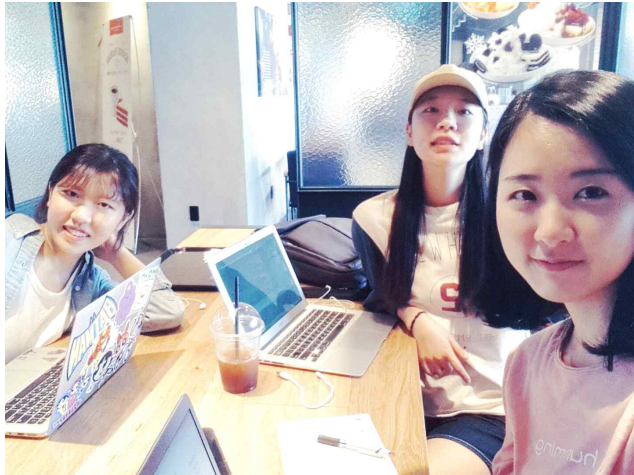
- 인물 구분 음성인식 프로그램 테스트하기
- 최종발표 계획 대비하기

나. 회의록

- 인물 구분 음성인식 프로그램을 이용하여, 여러 사람이 등장하여 대화하는 음성 파일을 분석해보았습니다.
- 최종발표에서 발표할 내용에 대해 논의해보았습니다.
- 프로젝트의 결과물에 대하여 구체적인 시나리오를 그려보았습니다.
- 최종발표에서 프로젝트의 결과물을 소개할 시연영상의 내용을 토의해보았습니다.

다. 진행 사항

- 인물 구분 음성인식 프로그램을 테스트해보고 개선점을 찾아보았습니다.
- 최종발표에 담을 내용을 토의해보았습니다.



14) 6월 12일 활동 일지

가. 목표

- 안드로이드 기기에서 네이버 음성인식 API기반 앱 테스트하기

나. 회의록

- JAVA 서버를 통해 안드로이드 기기에서 음성을 받아오는 것을 테스트했습니다.
- 음성을 받아올 때, 음성 인식에 용이한 형식으로 받아오도록 설정해주었습니다.
- 받은 음성을 서버에서 인물 구분 음성인식 프로그램의 Input으로 사용하여 분석해보았습니다.
- 분석된 정보를 바탕으로 화자의 이름과 말한 내용을 출력하였습니다.
- 여러 대의 안드로이드 기기를 사용하여 서버와의 통신을 테스트해보았습니다.
- 안드로이드 기기에서 앱을 테스트해보았습니다.
- 안드로이드 앱과 서버 간의 통신으로 원하는 정보만을 대화 목록에 출력하는 기능을 테스트해보고 원하는 작업을 수행하도록 수정하였습니다.

다. 진행 사항

- 프로젝트의 주제에 맞는 여러 기능들을 테스트하고 개선해야할 점을 수정해보았습니다.



15) 6월 14일 활동 일지

가. 목표

- 최종발표를 위한 시연영상 촬영하기
- 최종발표에 대한 발표자료 제작하기

나. 회의록

- 여러 번의 테스트를 거친 인물 구분 음성인식 프로그램과 여러 대의 안드로이드 기기를 사용하여, 이전에 만들어두었던 내용대로 시연영상을 촬영하였습니다.
- 함께 대화를 할 때, 각 휴대폰이 휴대폰 주인의 목소리만 인식하여 대화 목록에 출력하는 시연영상을 촬영하였습니다.
- 시연영상을 발표하기 좋게 여러 각도로 찍어 편집하였습니다.
- 프로젝트의 주제에 맞게 제작한 프로그램을 개발해온 내용들을 모두 최종발표 자료에 넣었습니다.

다. 진행 사항

- 완성된 프로그램을 소개하기위한 최종발표의 시연영상과 발표자료를 제작하였습니다.



4. 과제 수행 결과

1) GMM 생성



```
case Constant.SEND_BASE_WAVE:
    String base = "base_" + id;

    // 서버에 녹음된 파일 wav를 저장
    stream = new FileOutputStream(Constant.projectDir + "/working/" + base + ".wav");
    try {
        stream.write((byte[]) objectToWork.getObject(0));
    } finally {
        stream.close();
    }

    diarization(base); // Diarization
    rename(base); // segment 파일에 라벨 붙이기
    // ubm.gmm 을 활용한 pre training
    trainInit(base);
    fileCopy(Constant.projectDir + "/" + base + ".out.gmms", Constant.projectDir + "/working" + "/" + base + ".init.gmm");

    // EM 알고리즘을 이용한 training
    trainMap(base);
    fileCopy(Constant.projectDir + "/" + base + ".out.gmms", Constant.projectDir + "/voicedb" + "/" + base + ".gmm");

    gender = getGenderFromSegmentFile(Constant.projectDir + "/working/" + base + ".seg");
    writer.println("build complete");
    writer.flush();
```

프로젝트의 working 디렉토리에 전송받은 음성을 wav 파일 형식으로 저장한 후, GMM 생성 과정을 거친다. 결과물로 <id>.gmm 이 생성된다.

2) 화자인식

```
<Diarization>
java -Xmx2024m -jar lium_spkdiarization-8.4.1.jar --fInputMask=%s.wav --
sOutputMask=%s.seg --doCEClustering filebase
-> 결과물로 %s.seg 생성
```

```
1 |; cluster S0 | score:FS = -33.385052550324424 | [ score:FT = -34.70304825472198 | [ score:MS = -33.832848255022654 | [ score:MT = -34.807047359510904 ]
2 eunyoung 1 296 316 F S U S0
3 eunyoung 1 612 1512 F S U S0
4 eunyoung 1 2124 516 F S U S0
5 eunyoung 1 2672 754 F S U S0
6 eunyoung 1 3426 1364 F S U S0
7 eunyoung 1 4790 717 F S U S0
```

Female Studio, Femae Telephone, Male Studio, Male Telephone 중 Female Studio의 score가 가장 낮기 때문에 Female로 인식한다.

```
<ident file 생성>
S0를 label 로 대체하기
-> 결과물로 %s.ident.seg 생성
```

```
talktome -- vi eunyoung.ident.seg -- 161x16
1 |; cluster EUNYOUNG | score:FS = -33.385052550324424 | [ score:FT = -34.70304825472198 | [ score:MS = -33.832848255022654 | [ score:MT = -34.807047359510904 ]
2 eunyoung 1 296 316 F S U EUNYOUNG
3 eunyoung 1 612 1512 F S U EUNYOUNG
4 eunyoung 1 2124 516 F S U EUNYOUNG
5 eunyoung 1 2672 754 F S U EUNYOUNG
6 eunyoung 1 3426 1364 F S U EUNYOUNG
7 eunyoung 1 4790 717 F S U EUNYOUNG
```

임의로 붙는 cluster의 라벨 S0을 원하는 라벨로 대체한다.

```
<Train Init>
java -Xmx1024m -cp lium_spkdiarization-8.4.1.jar
fr.lium.spkDiarization.programs.MTrainInit --sInputMask=%s.ident.seg --
fInputMask=%s.wav --fInputDesc=audio2sphinx,1:3:2:0:0:0,13,1:1:300:4 --
emInitMethod=copy --tInputMask=ubm.gmm --tOutputMask=%s.init.gmm
filebase
-> 결과물로 %s.init.gmm 생성
```

UBM(Universal background model)을 이용해 기초 트레이닝을 시행한다.

```
<Train Map>
java -Xmx1024m -cp lium_spkdiarization-8.4.1.jar
fr.lium.spkDiarization.programs.MTrainMAP --sInputMask=%s.ident.seg --
fInputMask=%s.wav --fInputDesc=audio2sphinx,1:3:2:0:0:0,13,1:1:300:4 --
tInputMask=%s.init.gmm --emCtrl=1,5,0.01 --varCtrl=0.01,10.0 --
tOutputMask=%s.gmm filebase
-> 결과물로 %s.gmm 생성, 학습된 gmm 빌드 완료
```

%s.init.gmm과 음성 파일 및 세그먼트 파일로 MAP adaptation method를 적용하여 학습한다.

```
<Score>
java -Xmx256m -cp lium_spkdiarization-8.4.1.jar
fr.lium.spkDiarization.programs.MScore --sInputMask=%s.seg --fInputMask=
%s.wav --sOutputMask=%s.ident.score.seg --sOutputFormat=seg,UTF8 --
fInputDesc=audio2sphinx,1:3:2:0:0:0,13,1:0:300:4 --tInputMask=base.gmm --
sTop=8,ubm.gmm --sSetLabel=add --sByCluster filebase
```

-> 결과물로 %s.ident.score.seg 생성, score를 비교하여 본인 여부 판별

```
1 ;; cluster S0_EUNYOUNG [ score:EUNYOUNG = -31.9502610779388 ]
2 eunyoung 1 296 316 F S U S0_EUNYOUNG
3 eunyoung 1 612 1512 F S U S0_EUNYOUNG
4 eunyoung 1 2124 516 F S U S0_EUNYOUNG
5 eunyoung 1 2672 754 F S U S0_EUNYOUNG
6 eunyoung 1 3426 1364 F S U S0_EUNYOUNG
7 eunyoung 1 4790 717 F S U S0_EUNYOUNG
```

%s.gmm과 새로 들어온 음성을 비교하여 score를 매긴다.

```
case Constant.SEND_MESSAGE:
    String msgfile = "msg_" + id;
    // .pcm을 wav로 저장
    writeToWaveFile((byte[]) objectToWork.getObject(0), Constant.projectDir + "/working/" + msgfile + ".wav");

    // gender detection
    gender_detection(msgfile);
    Gender genderFromMsg = getGenderFromSegmentFile(Constant.projectDir + "/working/" + msgfile + ".seg");

    System.out.println("gender:" + gender + ", genderFromMsg:" + genderFromMsg);
    // 성별을 먼저 비교하고 성별이 같은 경우에만 더 세부적인 프로세스 수행
    if (gender == genderFromMsg) {
        diarization(msgfile); // diarization
        boolean result = score(msgfile); // 생성했던 base gmm과 매칭하여 score 획득
        if(result) {
            String msg = (String) objectToWork.getObject(1);
            sendAll(name + ">" + msg); // 집합 안산자를 통해서 사용자 이름과 내용이 다른 클라이언트들에게
        }
    }
    break;
```

음성을 wav파일로 저장하고, 먼저 gender detection을 수행한다.

```
private synchronized void gender_detection(String filebase) {
    try {
        String dir = Constant.projectDir + "/working";
        String[] segInitParams = {"-fInputMask=" + dir + "/%s.wav", "-fInputDesc=audio2sphinx,1:1:0:0:0:0,13,0:0:0",
            "-sInputMask=", "-sOutputMask=" + dir + "/%s.s.seg", filebase};
        MSegInit.main(segInitParams);

        String[] decodeParams = {"-fInputMask=" + dir + "/%s.wav", "-fInputDesc=audio2sphinx,1:3:2:0:0:0,13,0:0:0",
            "-sInputMask=" + dir + "/%s.s.seg", "-sOutputMask=" + dir + "/%s.g.seg", "-dPenalty=10,10,50",
            "-tInputMask=" + Constant.projectDir + "/sms.gmms", filebase};
        MDecode.main(decodeParams);

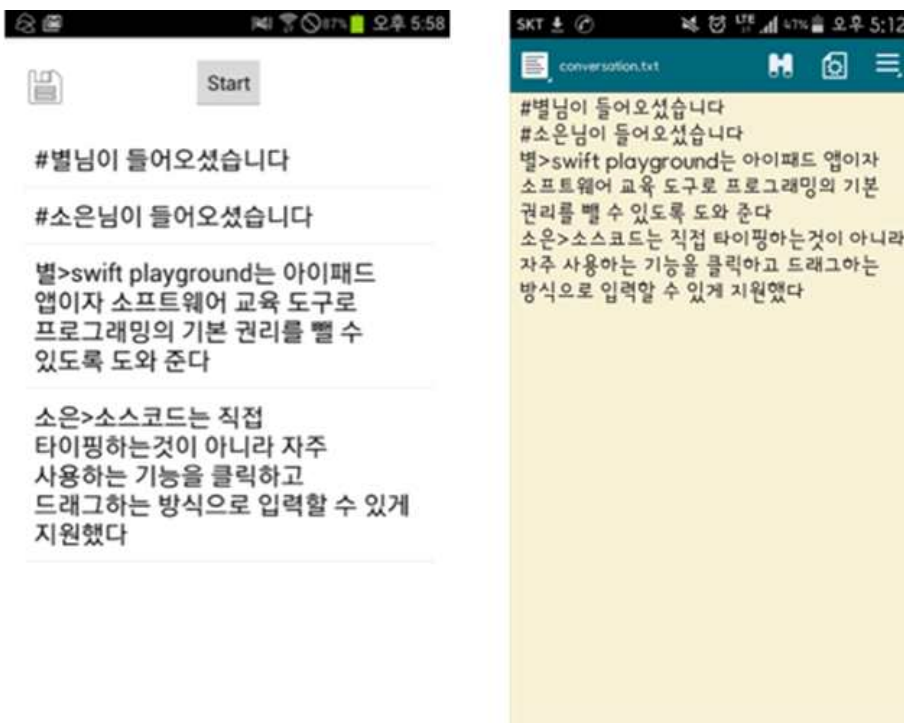
        String[] scoreParams = {"-help", "-sGender", "-sByCluster", "-fInputDesc=audio2sphinx,1:3:2:0:0:0,13,1:1:0:0",
            "-fInputMask=" + dir + "/%s.wav", "-sInputMask=" + dir + "/%s.g.seg", "-sOutputMask=" + dir + "/%s.seg",
            "-tInputMask=gender.gmms", filebase};
        MScore.main(scoreParams);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

sms.gmms와 gender.gmms 이용하여 diarization을 하는 것보다 빠른 gender detection을 수행한다. gender가 같은 경우 diarization을 수행하고 scoring 수행하여 본인 여부를 식별한다. 만일 본인이라고 판별한 경우에만 sendAll로 채팅방에 메시지를 전송한다.

3) 채팅방내용 저장 기능

```
case Constant.SAVE_REQUEST:
    writer.println("save");
    writer.flush();
    for (String str : conversations) {
        writer.println(str);
    }
    writer.println("finish");
    writer.flush();
    break;
```

서버에서 클라이언트로부터 저장 요청을 받으면 모든 대화들을 한 줄씩 클라이언트로 전송한다. 클라이언트는 text파일로 저장한다.



<저장결과>

4) Naver 음성인식 api

```
case R.id.finalResult:
    // NaverRecognizer 가 finalResult를 생성하면 서버로 음성 및 텍스트 전송
    String[] results = (String[]) msg.obj;
    mResult = results[0];
    txtResult.setText(mResult);
    if (!mResult.isEmpty()) {
        Log.d(TAG, "handleMessage: " + writer.getBytes().length);
        sendTalk(new OBJECT(Constant.SEND_MESSAGE, writer.getBytes(), mResult));
    }
    break;
```

NaverRecognizer가 final result단계에 오게 되면 화자식별을 위해 텍스트를 서버로 전송한다.

```

private void receive() {
    try {
        BufferedReader reader = new BufferedReader(new InputStreamReader(socket.getInputStream()));

        while (isRunning) {
            String str = reader.readLine();// str변수에 라인단위로 저장
            Log.d(TAG, "receive: " + str);

            if (str.equals("build complete")) { //GMM build 완료
                // Handler 이용하여 android UI 업데이트
                handler.sendMessage(0);
                BuildGMMActivity.isBaseGmmSet = true;
                continue;
            }
            if (str.equals("save")) { // 안드로이드의 TalkToMe 폴더에 대화 파일 저장
                BufferedWriter writer = new BufferedWriter(new FileWriter(Environment.getExternalStorageDirectory() + "/TalkToMe/conversation.txt"));
                str = reader.readLine();
                while (!str.equals("finish")) {
                    writer.write(str);
                    str = reader.readLine();
                }
                writer.close();
                continue;
            }
            if (isCancelled()) {
                break;
            }
            if (str.equals(speaker + "[out]")) { // 만일 client가 나가면 스레드 종료
                break; // 종료
            }

            publishProgress(new MessageItem(str)); // 대화 메시지 추가
        }

        Log.d(TAG, "receive: finish");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

클라이언트 측에서 소켓통신은 별도의 스레드로 send와 receive를 수행한다. 캡처한 코드는 receive 메소드 코드로 서버 측에서 보낸 메시지에 따라,

1. 클라이언트 GMM을 생성할 기초 음성 파일을 전달하고 만일 서버에서 프로세스가 완료되면 **build complete** 메시지를 받고, 화면에 Complete 메시지를 띄우고 다음 액티비티로 넘어갈 수 있다.
2. **save** 메시지를 받으면 안드로이드 단말의 TalkToMe 폴더에 텍스트 파일로 대화 기록을 모두 저장한다.
3. 만일 client가 어플리케이션을 종료하여 **[out]** 메시지를 받으면 소켓 통신을 하던 스레드를 종료한다.
4. 그 외의 메시지는 대화방에 새로운 메시지로 추가한다.

5. 과제 결과의 활용 방안

1) 과제 결과의 기대효과

가. 청각장애인의 자유로운 수업 수강 가능

- 청각장애인이 수업을 들을 때, 속기사의 도움 없이도 자유롭게 수강이 가능하다.
- 현장 강의나 인터넷 강의 등 수업의 형태에 구애받지 않고, 해당 프로그램을 활용할 수 있다.

나. 청각장애인과 비장애인과 자유로운 대화 가능

- 청각장애인이 비장애인과 대화를 나눌 때, 상대방의 입 모양을 유심히 보지 않아도 대화가 가능하다.
- 음성이 텍스트로 작성되어, 앞서 말한 내용을 천천히 읽으며 이해할 수 있다.

2) 과제 경과의 활용 방안

가. 회의 시 회의록 작성에 용이

- 회의에 참여하는 각 인물들이 말하는 내용을 화자별로 정리가 가능하다.
- 회의에 참여하는 사람들이 앞서 말한 내용을 돌이켜 읽어볼 수 있다.

나. 주인으로 등록된 사람의 말만 듣는 기기 개발 가능

- 기기가 화자를 구분할 수 있도록 개발할 수 있다.
- 등록된 화자인지를 판별하여 명령을 받아들이도록 구현할 수 있다.

다. 범죄사건 수사에 활용 가능

- 범인의 목소리의 특성을 분석하고 비슷한 음성을 구별해낼 수 있다.
- 범인의 목소리를 토대로 용의자의 범위를 줄일 수 있다.

라. 보안의 용도로 사용 가능

- 특정 인물만 입장이 가능하게 하기위하여 해당 인물의 목소리를 이용할 수 있다.
- 특정 인물만 접속해야하는데 그렇지 않은 인물이 접속하려할 경우, 음성 증거를 수집할 수 있다.

6. 경진대회/전시회 출품 경력 및 수상실적

없음.