요구사항 분석서

다중 음성 환경에서의 주 발화자 인식 및 텍스트 변환 기술

`23.10.03.

202011376 조현서

201711780 김용진

201711140 김준원

목차

1. 개 요

1.1 프로젝트 기획 배경

“다중 음성 환경에서의 고도화된 실시간 음성인식 (STT) 기술”

현대 사회에서 음성 인식 기술은 우리 일상의 많은 부분에서 활용되고 있습니다. 그러나 다중 음성 환경, 특히 많은 사람들이 함께 있는 공간에서의 음성 인식은 여전히 큰 문제로 남아 있습니다. 이 프로젝트의 주제는 이러한 환경에서 주 발화자의 목소리를 정확히 식별하고, 그 목소리를 텍스트 변환(STT)에 효과적으로 활용하여 인식 정확도를 향상시키는 것입니다.

누구나 한 번쯤은 스마트폰이나 TV의 음성 어시스턴트 기능을 사용하려 할 때, 주변 소음으로 인해 명확한 명령 입력이 어려워 추가적인 노력을 기울여야 했던 경험들이 있을 것입니다. 운전 중에는 네비게이션에 음성으로 목적지를 입력하려 했을 때, 동승자의 대화나 차량 내외부의 잡음으로 인해 명확한 음성 인식이 어려웠던 순간들도 있었을 것입니다. 이러한 상황들은 현 음성 인식 기술의 한계를 뚜렷하게 드러내며, 이를 극복할 필요성을 느끼게 합니다.

이에 본 프로젝트는 다양한 배경 소음 속에서도 주 발화자의 목소리의 특성을 명확히 인식할 수 있는 고도화된 STT 기술의 개발을 목표로 합니다. 이를 통해 사용자는 복잡한 환경에서도 음성 인식 기술을 원활하게 활용할 수 있게 되며, 이는 음성 기반의 서비스와 기술이 보다 폭넓게 확산되는 데 기여할 것입니다.

1.2 기술 동향 및 핵심 기술

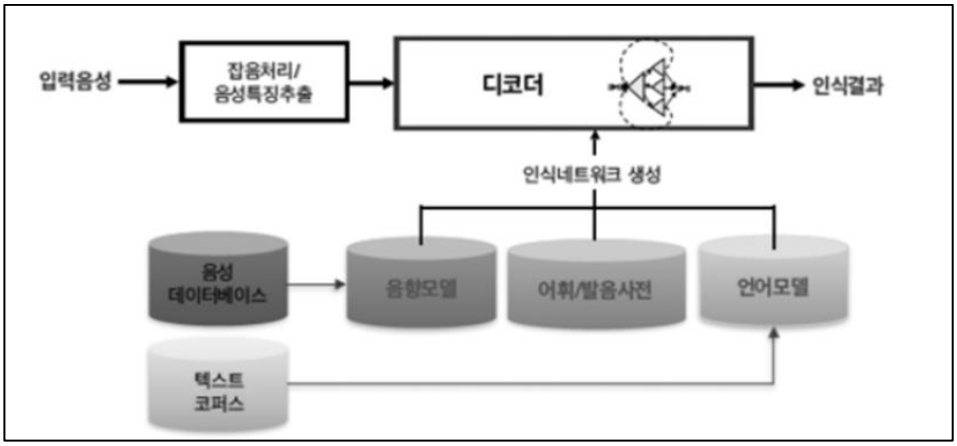
1.2.1 Speech-to-Text (STT)

Speech-to-Text(STT)는 사람의 음성을 텍스트 형식으로 변환하는 기술로, 다양한 응용 분야에서 중요한 역할을 합니다. 이 기술은 통화 내용의 자동 변환, 음성 명령 인식, 오디오 콘텐츠의 자막 생성 등에서 활용되며, 특히 스마트폰, 스마트 스피커, 자동차 내비게이션 시스템 등에서 일상적으로 사용됩니다.

STT의 기본 원리는 입력된 음성의 특징을 추출하고, 이를 텍스트와 연관짓는 것입니다. 이 과정에서는 음향 모델(Acoustic Model)이 음성의 특징을 단어나 발음으로 해석하고, 언어 모델(Language Model)은 해당 단어들이 어떻게 조합되어 문장을 이루는지 예측합니다.

전통적인 STT 시스템은 음향 모델링과 언어 모델링을 별도로 수행했으나, 최근의 딥러닝 기반 접근법에서는 이 두 과정을 통합하여 처리하는 End-to-End 모델이 주목받고 있습니다. 이러한 모델은 데이터의 양이 많아짐에 따라 성능이 획기적으로 향상되었으며, 특히 순환 신경망(RNN), 합성곱 신경망(CNN), 변환자(Transformer)와 같은 아키텍처를 활용하여 높은 정확도를 달성하고 있습니다.

STT 기술의 발전은 사용자 경험의 질을 크게 향상시키며, 자연어 처리 분야의 다른 기술과 결합될 때 더욱 강력한 시너지를 발휘합니다.



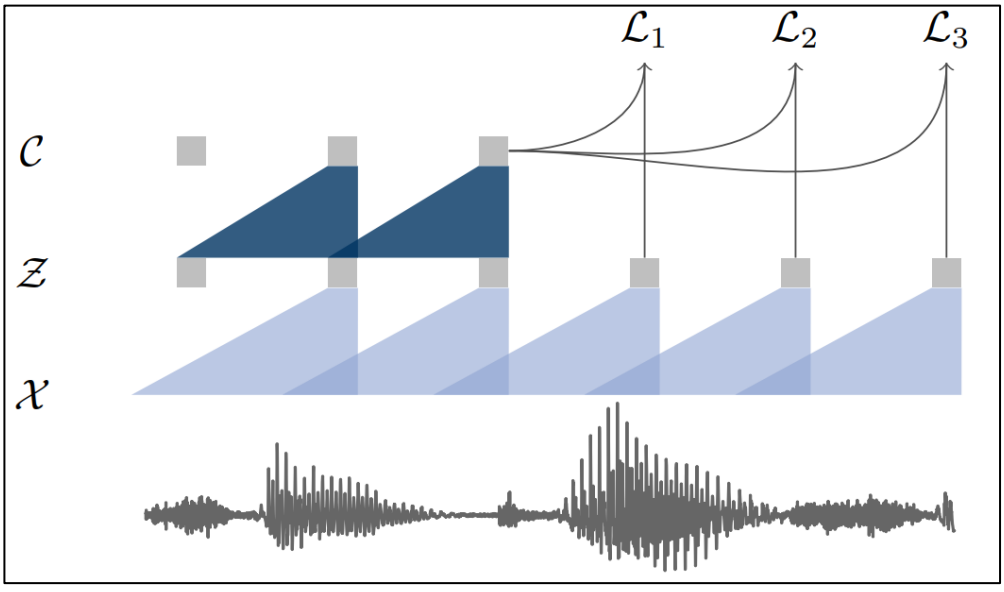
1.2.2 음성 벡터화 (Speech Vectorization)

음성 벡터화는 음성 데이터를 복잡한 고차원 구조에서 간결한 저차원 벡터 형태로 변환하는 과정을 말합니다. 이렇게 변환된 벡터는 딥러닝 모델의 입력으로 활용되어, 음성 인식, 감정 분석, 발화자 구분 등의 다양한 작업에 적용됩니다.

wav2vec는 Facebook AI에서 제안된 음성 벡터화 기술로, 원본 음성 데이터의 핵심 특징을 효율적으로 추출하여 벡터 형태로 나타내는데, 이를 통해 데이터의 크기는 줄이면서도 중요한 정보는 보존하는 것이 가능합니다.

wav2vec의 주요 특징은 대량의 라벨 없는 음성 데이터를 활용한 비지도 학습 방식을 채택하고 있다는 점입니다. 이를 통해 전통적인 음성 특징 추출 방법보다 더 다양하고 정밀한 정보를 포착할 수 있습니다. 특히, wav2vec를 통해 얻어진 벡터는 음성 인식과 같은 작업에서 높은 성능을 보여주며, 이 기술의 효율성을 입증하고 있습니다.

음성 벡터화는 딥러닝을 활용한 음성 처리 분야에서 모델의 학습을 더욱 정밀하고 효율적으로 만들어주는 핵심 기술로, wav2vec와 같은 혁신적인 방법론은 이 분야의 발전에 큰 도움을 주고 있습니다.



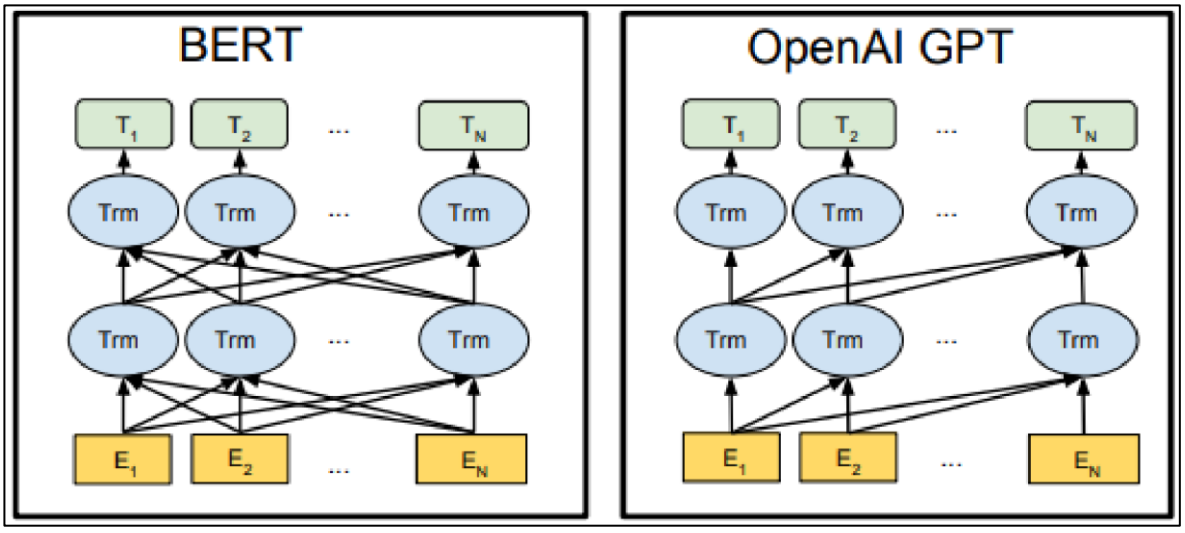
1.2.3 언어 모델 (Language Model)

언어 모델(Language Model, LM)은 단어나 문장의 확률을 예측하는 모델로, 주어진 단어들에 대해 다음 단어가 무엇일지 예측하는 데 사용됩니다. 이는 음성 인식, 기계 번역, 텍스트 생성 등 다양한 자연어 처리(NLP) 작업에서 핵심적인 역할을 합니다.

기본적으로 언어 모델은 주어진 문맥에서 단어의 나열이 얼마나 자연스러운지를 수치로 나타냅니다. 예를 들어, "나는 오늘 ___ 먹었다"라는 문장에서 빈 칸에 어떤 단어가 올 확률이 가장 높은지를 예측하는 것이 언어 모델의 한 예입니다.

전통적인 언어 모델은 n-gram 기반으로 작동하며, 주어진 단어들의 연속성을 기반으로 다음 단어의 확률을 계산합니다. 그러나 최근에는 딥러닝 기반의 언어 모델, 특히 순환 신경망(RNN)과 변환자(Transformer) 아키텍처가 주목받고 있습니다. 이러한 모델들은 더 큰 문맥을 고려하여 더 정확하고 다양한 예측을 할 수 있습니다.

OpenAI의 GPT나 Google의 BERT와 같은 최신 언어 모델은 대규모의 텍스트 데이터를 학습하여 높은 성능을 달성하고 있습니다. 이러한 모델들은 STT 시스템에서도 디코딩 단계에서 후보 문장들의 자연스러움을 평가하는 데 사용되어, 음성 인식의 정확도를 높이는 데 기여하고 있습니다.



1.2.4 Noise Reduction and Voice Isolation (소음 감소 및 목소리 분리)

소음 감소 및 목소리 분리는 오디오 처리 분야에서 중요한 연구 주제 중 하나로, 다양한 환경에서 발생하는 배경 소음을 제거하고 원하는 목소리나 소리만을 분리하여 드러내는 기술을 의미합니다.

1.2.4.1 소음 감소 (Noise Reduction):

소음 감소는 주변 환경의 잡음이나 간섭 소리를 최소화하여 오디오의 명료성을 향상시키는 기술입니다. 전통적인 방법으로는 스펙트럼 분석을 통해 소음과 목소리의 주파수 영역을 구분하고, 소음 영역을 제거하는 방식이 있습니다. 최근에는 딥러닝 기반의 방법이 주목받고 있으며, 특히 심층 신경망을 활용하여 복잡한 배경 소음도 효과적으로 제거할 수 있게 되었습니다.

1.2.4.2 목소리 분리 (Voice Isolation):

목소리 분리는 여러 음성이나 소리가 섞여 있을 때 특정 목소리만을 분리하는 기술입니다. 이 기술은 회의나 다중 대화 환경에서 주요 발화자의 목소리를 강조하거나, 특정 소리를 분석하기 위해 사용됩니다. 변환자(Transformer)나 순환 신경망(RNN) 같은 딥러닝 모델을 활용하여 여러 음성 중 원하는 목소리를 정밀하게 분리할 수 있습니다.

소음 감소 및 목소리 분리 기술은 음성 인식, 통화 품질 향상, 오디오 콘텐츠 제작 등 다양한 분야에서 활용되며, 특히 다중 음성 환경에서의 STT 성능 향상에도 큰 기여를 하고 있습니다. 이러한 기술의 발전은 사용자들에게 더욱 풍부하고 명료한 오디오 경험을 제공하게 됩니다.

1.3 프로젝트 주요 기능 및 특징

1.3.1 다중 발화 상황 인지 기능 (Overlapping Detection)

다중 발화 상황 인지 기능은 여러 사람이 동시에 말하는 상황, 즉 발화가 중첩되는 상황을 정밀하게 탐지하는 기능입니다. 이 기능은 주로 주발화자의 목소리 외에 다른 사람들의 목소리가 함께 들어올 때, 그 상황을 정확하게 인지하고 알려주는 역할을 합니다.

이 기능의 핵심은 '탐지'에 있습니다. 즉, 다중 발화가 일어나는 상황을 파악하고 그 정보를 사용자나 다른 시스템에 전달하는 것이 주 목적이며, 해당 상황에서의 발화 내용을 분석하거나 구분하는 작업은 이 기능의 범위에서 제외됩니다. 이를 통해 사용자는 현재의 오디오 환경이 단일 발화 상황인지, 아니면 다중 발화 상황인지를 명확하게 알 수 있게 됩니다.

1.3.2 실시간 주 발화자 인식 기능

주 발화자 인식 기능은 다중 발화 상황에서도 주 발화자의 목소리를 정밀하게 구분하고 필터링하는 기술입니다. 여러 사람이 동시에 말하는 환경에서도, 이 기능은 주 발화자의 고유한 목소리 특성을 파악하여 그 목소리만을 선별적으로 추출합니다.

이 기능은 주 발화자의 목소리 특성과 패턴을 미리 학습하고, 실시간으로 들어오는 오디오 스트림에서 그 특성과 일치하는 부분만을 인식하고 필터링합니다. 이로써, 다른 사람들의 목소리나 배경 잡음 등이 섞여 있더라도 주 발화자의 목소리만을 명확하게 들을 수 있게 됩니다.

1.4 조원 구성 및 역할 분담

- **조현서**
  - 프로젝트 총괄
  - 언어 모델을 이용한 음성 벡터의 디코딩 구축
  - 전체 구조 설계
- **김용진**
  - 다중 발화 환경에서의 주 발화자 인식 기술 동향 파악 및 구축
  - 데이터셋 구축
  - 음성 데이터 벡터화 기술 구축
- **김준원**
  - 모델 경량화 기술 동향 파악 및 구축
  - 음성 데이터 벡터화 기술 구축
  - 활동 보고서 작성

1.5 일정

- **10월**
  - 주요 기능 구체화
  - 최신 연구 동향 파악
- **11월**
  - 데이터 셋 구축
  - 아이디어 및 기술 후보군 선정

- **12월**
  - 음성 데이터 벡터화 구축
  - 벡터화 모듈 학습 및 하이퍼파라미터 파인 튜닝
- **1월**
  - 다중 발화자 인식 기술 구축
  - 해당 모듈 학습 및 하이퍼파라미터 파인 튜닝
- **2월**
  - 다중 발화자 환경에서 노이즈 제거 기술 구축
  - 해당 모듈 학습 및 하이퍼파라미터 파인 튜닝
- **3월**
  - 인코딩을 언어 모델과 결합하여 텍스트 변환 기술 구축
  - Loss 함수 튜닝 및 end-to-end 학습 진행
- **4월**
  - 다양한 다중 음성 환경에서 모델 파인 튜닝
  - End-to-end 모델 성능 개선
- **5월**
  - 모델 경량화를 통한 실시간성 확보
- **6월**
  - 결과 보고서 작성
  - 시연 제작

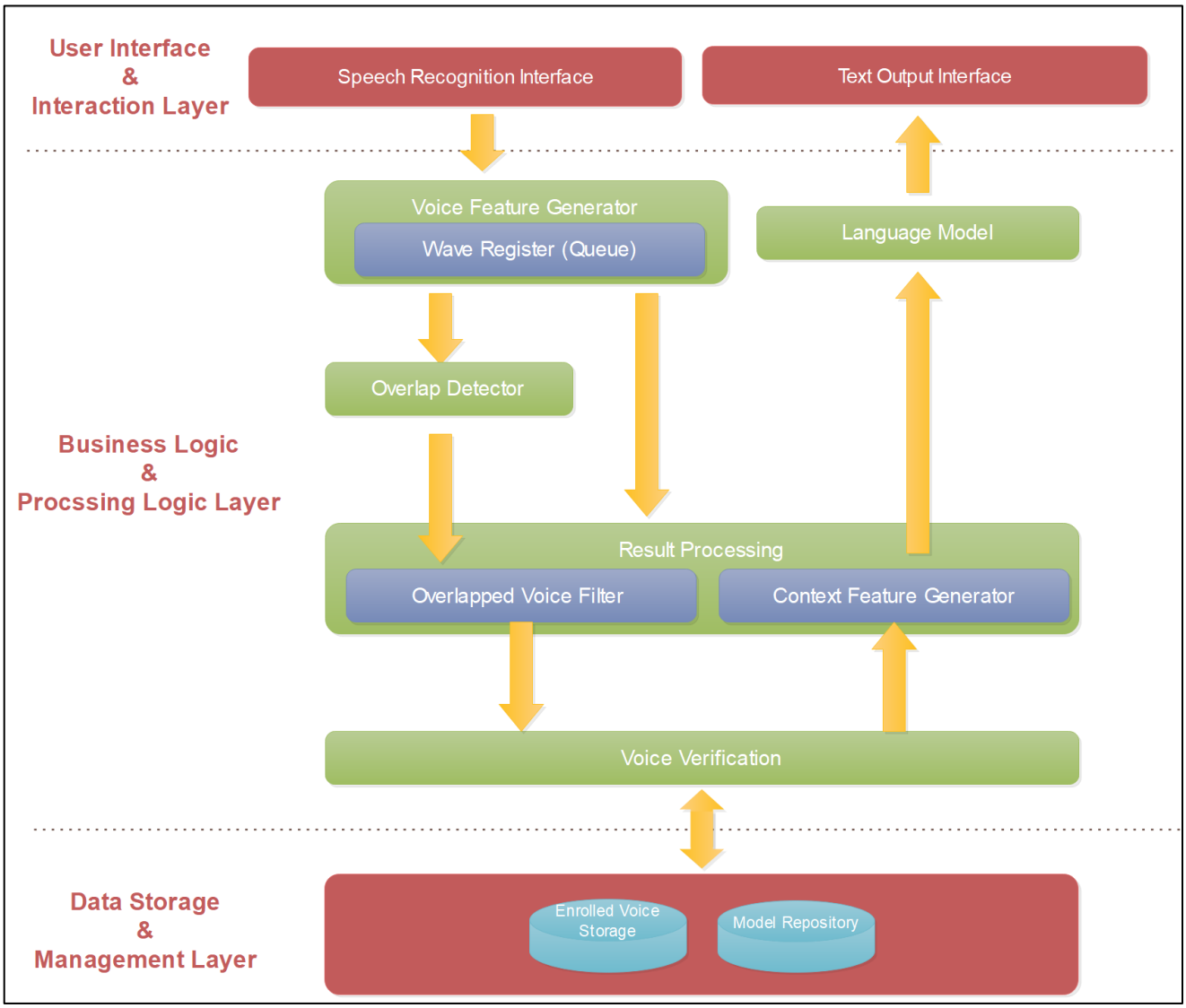
2. 기능적 요구사항

Usecase Diagram 은 주로 사용자와 시스템 간의 상호작용을 중심으로 그려지는 도구입니다. 그러나 본 프로젝트는 특정 사용자와의 상호작용보다는 시스템 내부의 모듈 기능과 성능 향상에 초점을 맞추고 있습니다.

이러한 프로젝트의 특성상, 시스템의 전체적인 구조와 각 모듈 간의 관계를 명확하게 표현하는 소프트웨어 아키텍처가 Usecase Diagram 보다 더 적합하다고 판단하였습니다. 소프트웨어 아키텍처는 각 구성요소와 그 사이의 연결을 분명히 드러내 줄 수 있어, 본 프로젝트의 핵심인 실시간 음성인식 기술의 복잡한 구조와 특성을 효과적으로 설명하는 데 도움이 됩니다.

따라서, 프로젝트의 목적과 방향성을 더욱 명확하게 전달하기 위해, Usecase Diagram 대신 소프트웨어 아키텍처를 도입하여 프로젝트의 전반적인 구조와 기능을 표현하기로 결정하였습니다.

2.1 소프트웨어 아키텍쳐



2.2 전체 구조

저희 시스템의 아키텍쳐는 크게 "User Interface & Interaction Layer", "Business Logic & Processing Logic Layer", "Data Storage & Management Layer"의 세 부분으로 구성됩니다.

2.2.1 User Interface & Interaction Layer

이 계층은 사용자와의 직접적인 상호작용을 담당합니다. 사용자로부터 음성 입력을 받아들이고, 처리된 결과인 텍스트를 사용자에게 표시하는 역할을 합니다.

2.2.2 Business Logic & Processing Logic Layer

이 계층은 시스템의 핵심 로직을 포함하고 있습니다. 사용자로부터 받은 음성 데이터는 이곳에서 전처리되며, 다중 음성 상황을 인지하고 필터링합니다. 또한, 주 발화자를 확인하고, 맥락에 따른 특징 정보를 생성하는 작업도 이 계층에서 이루어집니다. 그리고 언어 모델을 통해 맥락에 따른 특징 정보를 텍스트로 변환하는 작업도 이곳에서 처리됩니다. 저희가 주로 집중하고 개발하려는 기능들은 대부분 이 계층에 속합니다.

2.2.3 Data Storage & Management Layer

이 계층에서는 시스템에서 필요로 하는 다양한 데이터, 예를 들면 사용자의 음성 데이터나 모델의 파라미터 등이 저장되고 관리됩니다. 현재 프로젝트의 범위 내에서는 이 계층을 직접적으로 다루지는 않지만, 시스템의 안정적인 운영을 위해 필수적인 부분입니다.

2.3 구성 요소 (Component)

본 프로젝트에서 핵심적으로 연구 및 개발을 진행하고 있는 "Business Logic & Processing Logic Layer"의 주요 구성 요소에 대해 상세히 설명하겠습니다.

2.3.1 Voice Feature Generator (음성 특징 생성기)

이 구성 요소는 "User Interface & Interaction Layer"로부터 받아온 음성 데이터를 일시적으로 큐 데이터 구조에 보관합니다. 그 후, 이 데이터는 여러 Convolution Layer들을 통해 전처리되어 음성의 핵심 특징들을 추출합니다.

2.3.2 Overlap Detector (다중 발화 감지기)

이 구성 요소는 여러 목소리가 중첩되어 발화되는 상황, 즉 다중 발화 상황을 정확하게 감지하는 역할을 합니다.

2.3.3 Result Processing (결과 처리기)

이 구성 요소는 각 계층에서 출력된 중간 결과들을 적절히 처리합니다. 특히 다중 발화 상황에서 발생하는 목소리들을 필터링하고, 주 발화자의 목소리에 대해서는 전처리된 데이터를 바탕으로 맥락에 따른 특징을 생성합니다.

2.3.4 Language Model (언어 모델)

이 구성 요소는 주 발화자의 맥락에 따른 특징을 입력으로 받아, 그에 해당하는 텍스트를 출력합니다. 이 과정에서 복잡한 언어 패턴과 구조를 해석하여 음성을 텍스트로 정확하게 변환합니다

2.3.5 Voice Verification (음성 검증기)

이 구성 요소는 데이터 스토리지와 상호작용하면서, 전처리된 음성 특징이 사전에 등록된 특징과 일치하는지를 검증합니다. 이를 통해 시스템은 주 발화자의 목소리를 신뢰도 높게 인식하게 됩니다.

2.4 구성 요소 간의 관계 및 상호 작용

2.4.1 Voice Feature Generator와 Overlap Detector

Voice Feature Generator는 사용자의 음성을 받아 그 안에서 중요한 목소리 및 음성 특징을 추출합니다. 이렇게 추출된 음성 특징은 Overlap Detector로 전송되며, Overlap Detector는 이 정보를 분석하여 현재의 음성 상황이 다중 발화 상황인지 아닌지를 정밀하게 판단합니다.

2.4.2 Voice Feature Generator와 Overlap Detector, Result Processing

Overlap Detector가 다중 발화 상황을 감지하면, 이에 대한 정보를 Result Processing에 즉시 전달합니다. Result Processing은 이 정보를 참고하여 목소리 필터링 작업을 진행하며, 다중 발화 상황이 아닐 경우, Voice Feature Generator에서 전처리된 음성 특징을 그대로 사용하게 됩니다.

2.4.3 Result Processing, Language Model, 및 Voice Verification

Result Processing에서 정제된 음성 특징은 Voice Verification 과정으로 이동합니다. Voice Verification은 이 특징이 사전에 등록된 사용자의 목소리 특징과 얼마나 일치하는지를 검증하는 과정입니다. 만약 검증 과정에서 해당 음성 특징이 등록된 사용자의 목소리와 일치한다고 판단되면, 이 특징은 Language Model로 전송됩니다. Language Model은 이 특징을 기반으로 사용자의 발화 내용을 텍스트로 변환합니다. 반면, 일치하지 않는다면, 이는 주 발화자의 목소리가 아니라고 판단되어 해당 음성 정보는 처리되지 않고 버려집니다.

2.4.4 Voice Verification과 Data Storage & Management Layer

Voice Verification은 등록된 사용자의 음성 특징 정보를 Data Storage & Management Layer에서 가져옵니다. 그 후, 실시간으로 처리되는 음성 특징과 저장된 음성 특징을 비교하여 일치하는지 확인합니다.

2.4.5 Language Model과 User Interface & Interaction Layer

Language Model에서 음성을 텍스트로 변환한 결과는 User Interface & Interaction Layer로 전송됩니다. 이렇게 변환된 텍스트는 사용자에게 직접적으로 보여지게 되어, 사용자는 자신의 발화 내용이 어떻게 텍스트로 변환되었는지 확인할 수 있습니다.

3. 비기능적 요구사항

3.1 성능

본 프로젝트에서 중요하게 강조하는 성능 요구사항은 다음과 같습니다:

3.1.1 응답 시간

시스템은 사용자의 요청에 대해 신속하게 응답해야 합니다. 특히 실시간 음성 인식과 같은 기능에서는 지연 없이 결과를 제공하는 것이 중요합니다.

3.1.2 정확도

음성 인식 및 다른 관련 기능들은 높은 정확도를 유지해야 합니다. 잘못된 인식이나 오류는 사용자 경험을 저하시킬 수 있기 때문에 최소화되어야 합니다.

3.1.3 자원 최적화

시스템은 할당된 자원(CPU, 메모리, 저장 공간 등)을 효율적으로 사용해야 합니다. 불필요한 자원 낭비는 최소화하며, 필요한 작업에 충분한 자원을 할당해야 합니다.

3.1.4 확장성

시스템은 발화자 수나 등록된 목소리 데이터의 증가에 유연하게 대응할 수 있어야 합니다. 필요에 따라 추가 자원을 투입하여 성능을 유지하거나 향상시킬 수 있어야 합니다.

이러한 성능 요구사항은 프로젝트의 목표와 특성을 반영하여 세부적으로 조정 및 구체화될 수 있습니다.

3.2 사용성

본 프로젝트는 전통적인 사용자 인터페이스의 사용성에 중점을 두지 않습니다. 대신, 프로젝트의 핵심은 특정 모듈의 기능 향상과 성능 최적화에 있습니다. 이는 사용자의 직접적인 인터랙션 없이도 시스템이 원활하게 작동하며, 사용자의 요구와 환경에 맞게 효율적으로 응답해야 함을 의미합니다. 따라서, 내부적인 사용성은 시스템의 안정성, 반응성 및 오류 처리 능력에 중점을 둡니다.

3.3 이식성

본 프로젝트의 결과물은 다양한 플랫폼과 응용 프로그램에 쉽게 적용될 수 있어야 합니다. 스마트폰, 스마트 홈 기기, 자동차 네비게이션, 청각 장애인을 위한 STT 서비스 등 다양한 환경에서의 이식성을 고려하여 모듈을 설계하고 개발합니다. 이를 위해 표준화된 인터페이스, 모듈화된 구조, 플랫폼 독립적인 코드 작성 등의 방법을 적용합니다.

3.4 신뢰성

신뢰성은 시스템이 예상대로, 그리고 지속적으로 정확하게 작동하는 능력을 의미합니다. 본 프로젝트에서는 다음과 같은 요소를 고려하여 신뢰성을 확보합니다:

3.4.1 오류 처리

시스템은 예기치 않은 입력이나 상황에도 안정적으로 작동해야 합니다. 이를 위해 오류 처리 메커니즘을 강화합니다.

3.4.2 테스트

다양한 시나리오와 환경에서의 테스트를 통해 시스템의 안정성을 검증합니다.

3.4.3 백업 및 복구

데이터 손실이나 시스템 장애 시, 빠르게 복구할 수 있는 메커니즘을 구축합니다.

3.5 법적 책임

법적 책임은 프로젝트와 관련된 법률 및 규정을 준수하는 것을 의미합니다. 본 프로젝트에서는 다음과 같은 사항을 고려합니다:

3.5.1 데이터 보호

사용자의 개인 정보나 민감한 데이터를 처리할 때, 관련 법률 및 규정을 준수합니다. GDPR, CCPA와 같은 데이터 보호 규정에 따라 데이터를 수집, 저장, 처리합니다.

3.5.2 저작권

사용하는 코드나 라이브러리, 데이터에 대한 저작권을 확인하고, 라이선스를 준수합니다.

3.5.3 접근성

청각 장애인을 위한 STT와 같은 서비스를 제공할 때, 장애인 차별 금지법 등 관련 법률을 준수하여 모든 사용자가 동등하게 서비스를 이용할 수 있도록 합니다.

이러한 법적 책임을 준수함으로써, 프로젝트는 법적 리스크를 최소화하고, 사용자의 신뢰를 얻을 수 있습니다.

REFERENCES

1. "Deep Speech: Scaling up end-to-end speech recognition"., A. Hannun et al., arXiv (2014).
2. "WaveNet: A Generative Model for Raw Audio"., A. van den Oord et al., CoRR (2016).
3. "Listen, Attend and Spell"., W. Chan et al., ICASSP (2016).
4. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition"., D.S. Park et al., Interspeech (2019).
5. "wav2letter: an End-to-End ConvNet-based Speech Recognition System"., R. Collobert et al., arXiv (2016).
6. "wav2vec: Unsupervised Pre-training for Speech Recognition"., A. Baevski et al., Interspeech (2019).
7. "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations"., A. Baevski et al., NeurIPS (2020).
8. "End-to-end Speech Recognition with Neural Transducer"., N. Jaitly et al., NIPS (2016).
9. "Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning"., S. Kim et al., ICASSP (2017).
10. "Transformer-based Acoustic Modeling for Hybrid Speech Recognition"., A. Zeyer et al., ICASSP (2020).
11. "Attention is All You Need"., A. Vaswani et al., NeurIPS (2017).
12. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"., J. Devlin et al., NAACL-HLT (2019).
13. "GPT-3: Language Models are Few-Shot Learners"., T. Brown et al., NeurIPS (2020).
14. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter"., V. Sanh et al., NeurIPS (2019).
15. "A Simple Framework for Contrastive Learning of Visual Representations (SimCLR)"., T. Chen et al., ICML (2020).
16. "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images"., D. Ulyanov et al., ICML (2016)