SMART UP

길진성 김태성 김현욱 정선진

음성을 통한 강의 내용 요약 시스템

목차

1. 선정한 주제

음성을 통한 강의 내용 요약 시스템

1-1. 주제 개요

강의 음성 데이터를 통해 AI 를 학습시켜 강의에 특화된 stt 기술 개발
기존의 문서 요약 알고리즘들의 장점을 이용하여 보다 정교한 문서 요약 알고리즘 개발
개발한 문서 요약 알고리즘을 api 로 생성하여 라즈베리파이에서도 지연을 최소화
(stt 를 통해 음성을 텍스트로 변환 -> 문서 요약 알고리즘을 통해 내용 요약 -> 요약된 내용은 컴퓨터로 전송)

2. 주제를 선정한 이유 및 근거

기존에 음성을 텍스트로 변환하여 요약하는 기술은 이미 존재하지만 저희 팀은 강의라는 특수한 상황에 집중하여 개발할 생각입니다. 강의에 특화된 stt 기술을 개발하여 기존의 stt 보다 강의 음성에서의 인식률을 높이고 기존의 문서 요약 알고리즘의 장점들을 차용하여 새로운 문서 요약 알고리즘을 개발할 것입니다. 또한 문서 요약 알고리즘을 api 로 생성하고 처리 시간을 단축시켜 강의가 진행되는 중에 요약본을 확인할 수 있도록 할 예정입니다.

3.문제 해결의 가치

가치

-효율적인 학습

학생들은 수업이나 강의를 들을 때 많은 양의 정보를 받아들이게 됩니다. 이 정보를 이해하고 기억하는 것은 어려울 수 있습니다. 강의 내용 요약 시스템은 이러한 정보를 요약하고 정리하여 학습자가 쉽게 이해하고 기억할 수 있도록 도와줍니다.

-시간 절약

학생들은 제한된 시간 동안 많은 수업을 들어야 합니다. 강의 내용 요약기는 핵심적인 내용을 간결하게 전달하므로 시간을 절약에 유리하여 학생들이 더 많은 주제를 다룰 수 있습니다.

-정보 필터링

긴 강의에서 핵심 정보를 추출하는 것이 중요합니다. 강의 내용 요약기를 사용하면 대량의 정보를 필터링하여 핵심 내용에 초점을 맞출 수 있습니다.

4.알고리즘의 성능 평가

-음성 인식

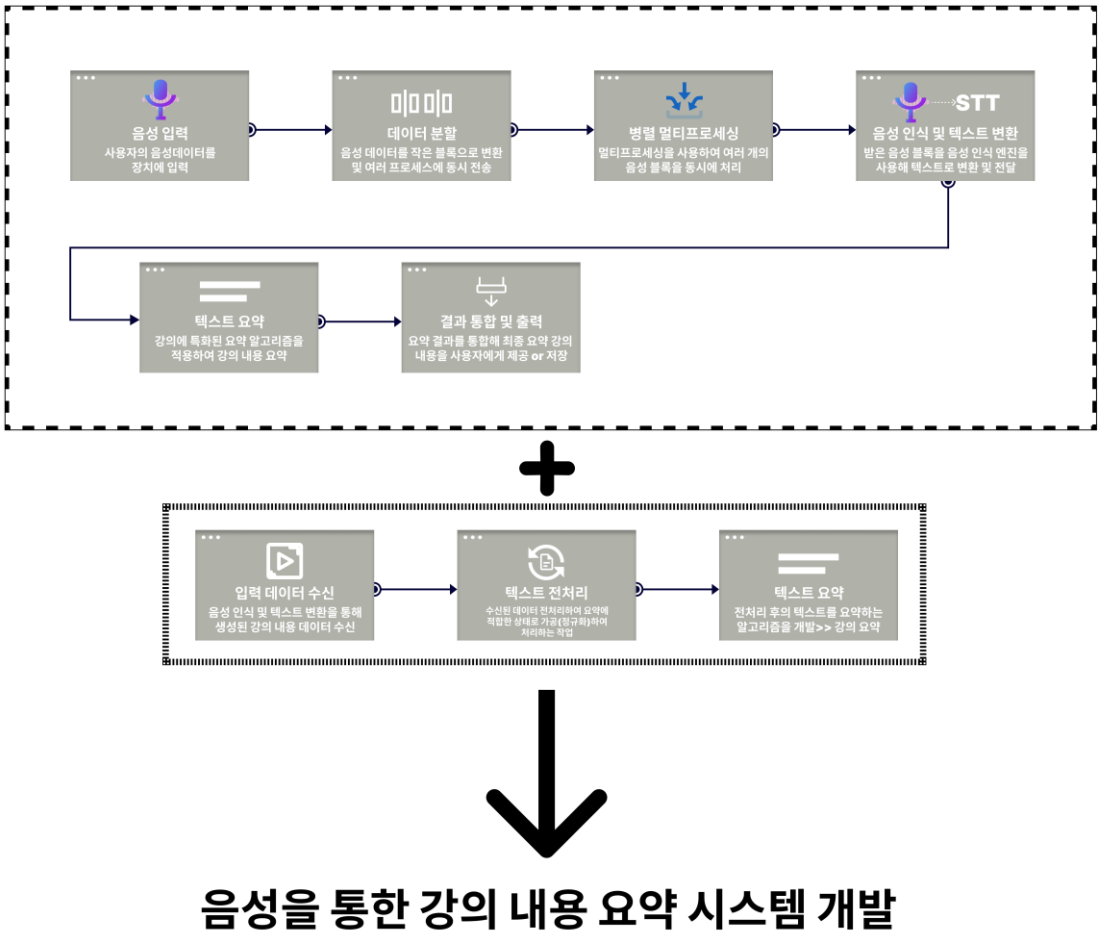
stt 성능 비교에 대해서는 CER 을 사용할 예정입니다.
음성 인식 성능 비교에는 WER(Word Error Rate)와 CER(Character Error Rate)가 존재합니다.둘 사이의 차이점은 WER 은 단어가 토큰이 되며, CER 은 문자가 토큰이 되는 것입니다.한국어 음성인식에서는 WER 보다 CER 이 더 중요한 척도로 여겨집니다. 한국어는 교착어(첨가어)로 조사를 사용하고 다른 언어와 비교하여 형태소의 구조가 복잡하며, 단어와 단어 사이의 경계가 모호합니다. 이러한 언어 구조의 특성으로 인해 단어 수준에서의 평가가 어렵습니다. 따라서, 문자 단위의 오류를 측정하는 CER 이 한국어 음성인식에서 더 정확한 평가 방법으로 간주되어 CER 을 사용할 예정입니다.

-문서 요약 알고리즘

요약 성능 비교에 대해서는 rouge-score 를 사용할 예정입니다.
rouge-score 는 문서와 문서에 대한 요약본을 제공합니다.이를 통해 문서를 알고리즘으로 요약한 요약본과 제공된 요약본과의 유사도로 성능 지표를 비교할 수 있습니다.rouge-score 에는 4 가지 항목(Rouge-N ,Rouge-L ,Rouge-W ,Rouge-S)이 있습니다.
1)Rouge-N 은 참조 요약과 생성된 요약간의 단어 수의 일치를 측정합니다.
2)Rouge-L 은 가장 긴 공통 부분 문자열을 계산하여 요약과 참조 사이의 유사성을 측정합니다.
3)Rouge-W 는 주어진 참조 요약의 중요 단어를 고려하여 요약의 중요성을 측정합니다.
4)Rouge-S 는구문 유사성을 고려하여 요약의 유사성을 측정합니다.

2. 주제 구현방법 및 보조자료

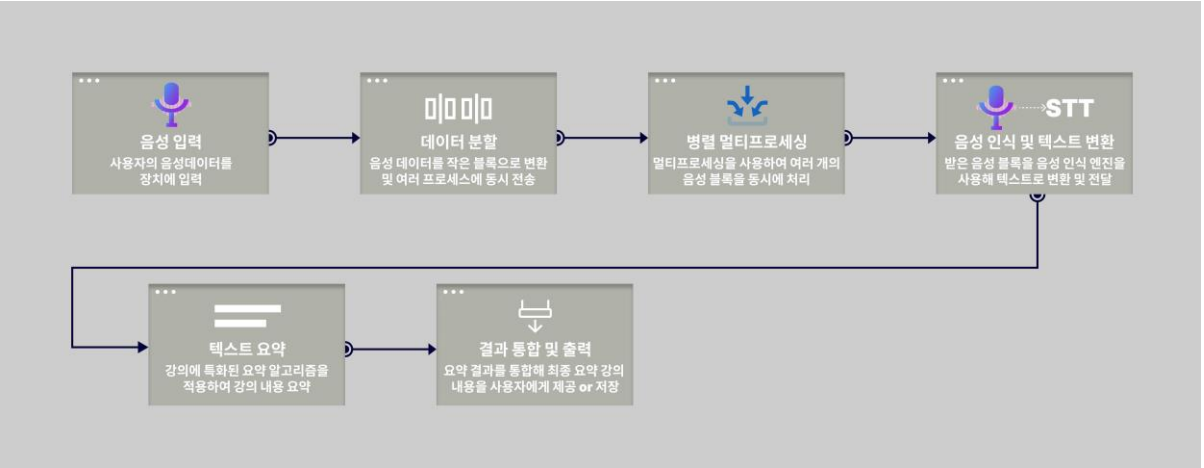
- 구현 과정에 대한 이미지 (Flow Chart)



음성을 통한 강의 내용 요약 시스템 개발

-소프트웨어 구현방법

1. 음성 최적화 구현



개발 개요

음성 입력:
사용자가 강의나 회의 등의 음성 데이터를 장치에 입력합니다.

데이터 분할:
입력된 음성 데이터는 라우터를 통해 작은 블록 또는 프레임으로 분할됩니다. 이렇게 분할된 데이터는 여러 프로세스에 동시에 전송됩니다.

병렬 멀티프로세싱:
라즈베리 파이에서 멀티프로세싱을 사용하여 여러 개의 음성 블록을 동시에 처리합니다. 각 프로세스는 하나의 음성 블록을 받아서 음성 인식 및 텍스트로의 변환 작업을 수행합니다.

음성 인식 및 텍스트 변환:
각 프로세스는 받은 음성 블록을 음성 인식 엔진을 사용하여 텍스트로 변환합니다. 이 단계 에서 배포된 강의용 음성 데이터셋을 이용해 목적에 특화된 음성인식 알고리즘을 개발합니다. 이렇게 변환된 텍스트는 후속 단계로 전달됩니다.

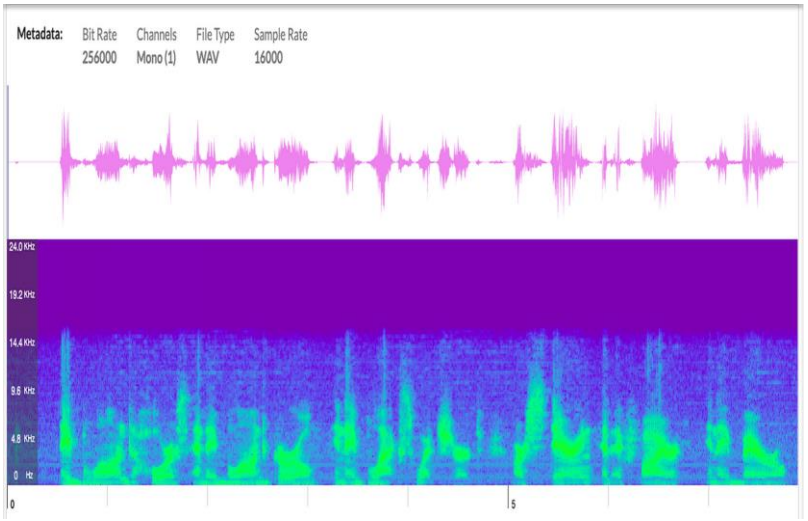
텍스트 요약:

텍스트 데이터는 강의에 특화된 요약 알고리즘을 적용하여 강의 내용을 요약합니다.

결과 통합 및 출력:
각 프로세스에서 생성된 요약 결과를 통합하여 최종 요약된 강의 내용을 생성합니다. 이러한 결과는 사용자에게 제공되거나 저장됩니다.
이와 같은 과정을 거쳐 음성 데이터가 분할되고 병렬로 처리되며,
최종적으로 강의 내용이 요약되어 사용자에게 제공됩니다.

※ 개발 단계

1. 데이터 수집 및 전처리

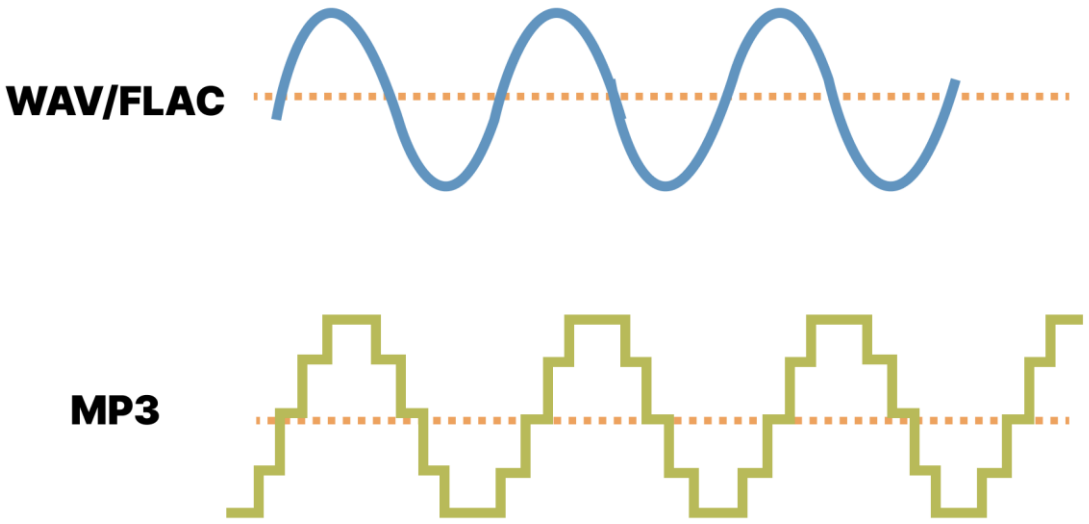


데이터 수집:

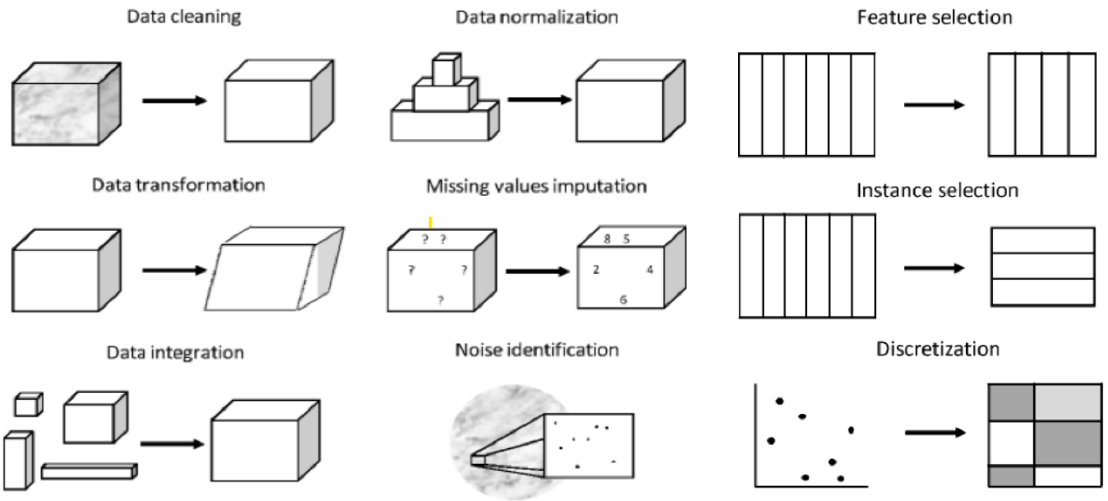
AI Hub

AIHub 또는 다른 오픈 데이터 포털에서 강의용 음성 데이터셋을 검색하고 다운로드합니다.
데이터셋의 크기, 형식, 언어 등을 고려하여 적합한 데이터셋을 선택합니다.

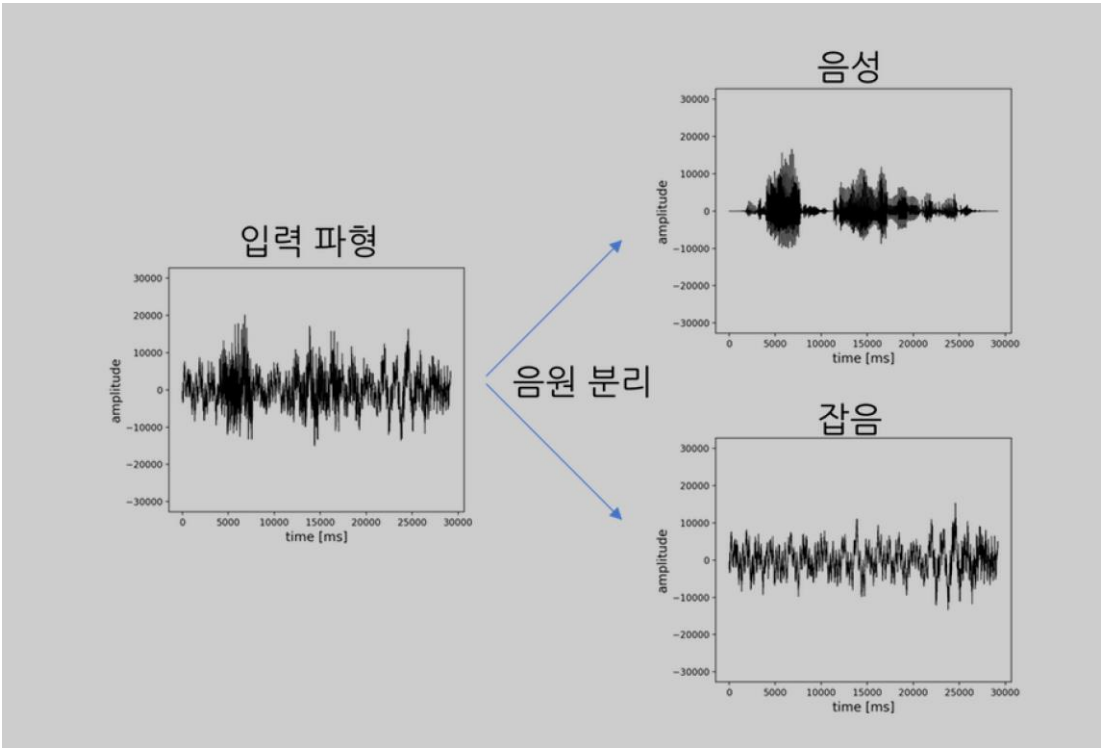
데이터 형식 변환:



다운로드한 음성 데이터의 형식을 확인하고, 프로젝트에 사용할 수 있는 형식으로 변환합니다. 일반적으로 음성 데이터는 오디오 파일 형식으로 제공됩니다. 주로 사용되는 형식은 WAV, MP3 등이 있습니다.
필요에 따라 음성 데이터를 텍스트로 변환하는 작업도 수행할 수 있습니다. 이를 위해 음성 데이터에 대한 대본이 제공되어야 합니다.

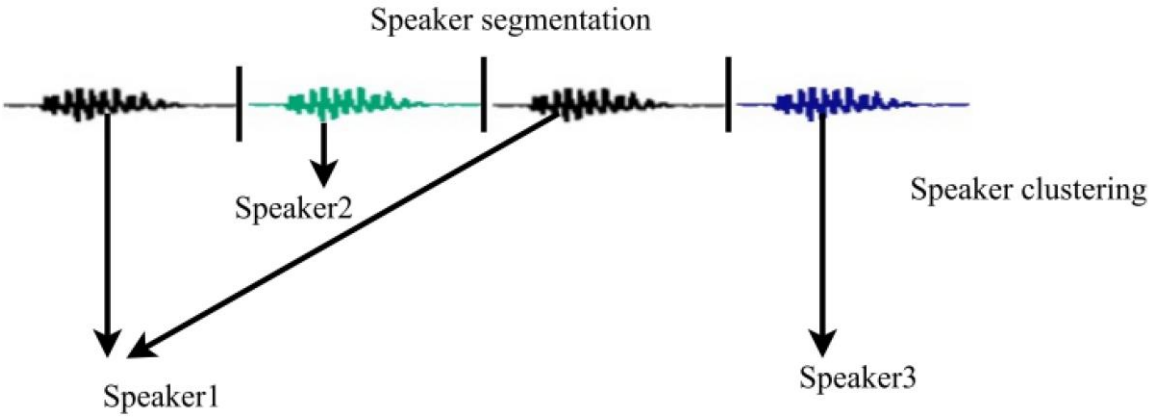


데이터 전처리:



데이터의 품질을 향상시키기 위해 전처리 작업을 수행합니다.
노이즈 제거: 음성 데이터에 포함된 백색 소음 또는 환경 소음을 감소시키기 위해 노이즈 제거 기술을 적용합니다.

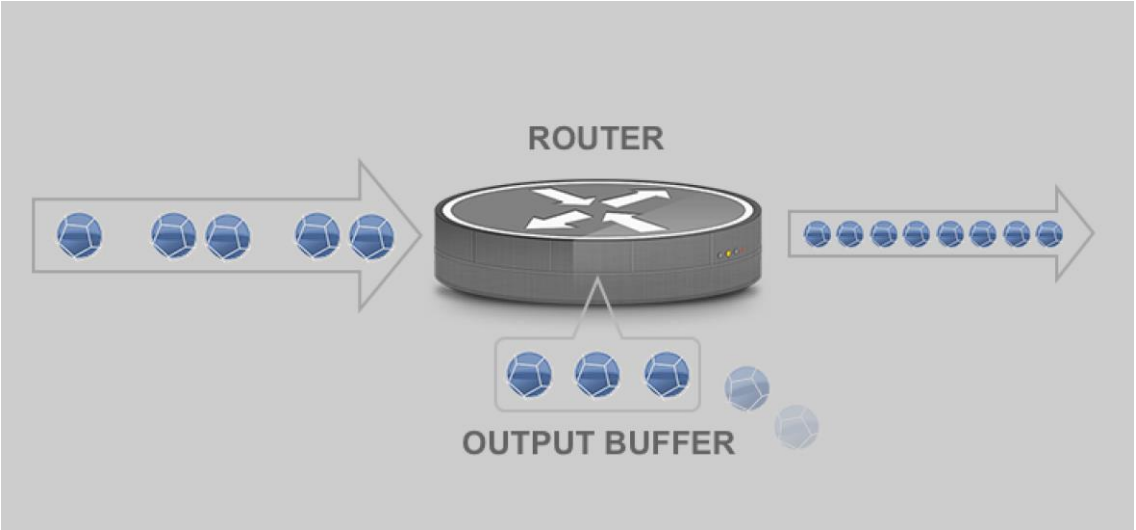
음성 신호 정규화:

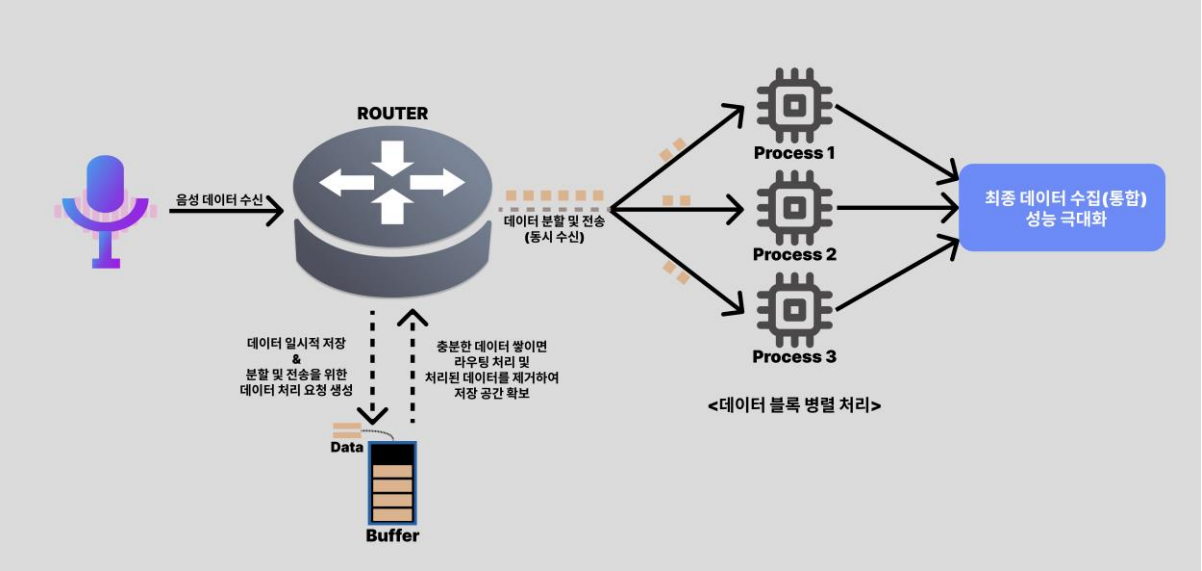


음성 신호의 크기를 조정하여 일관된 음량 수준을 유지합니다.
음성 세그멘테이션: 음성 데이터를 각각의 발화 단위로 분할하여 처리하기 쉽도록 합니다.

데이터 표준화: 데이터셋의 형식을 표준화하여 일관된 형식으로 데이터를 관리합니다.

2. 라우터를 통한 데이터 분할





데이터 수신 및 버퍼링:

라우터는 외부 소스로부터 들어오는 음성 데이터를 수신합니다. 이 데이터는 일시적으로 버퍼에 저장됩니다. 버퍼는 데이터를 일시적으로 보관하여 프로세스가 처리할 준비가 될 때까지 기다리는 역할을 합니다.

데이터 분할 요청 수신:

버퍼에 충분한 데이터가 쌓이면, 라우터는 분할 및 전송을 위한 데이터 처리 요청을 생성합니다. 이 요청은 데이터를 작은 블록 또는 프레임으로 분할하고, 각각의 블록을 병렬로 처리할 프로세스에게 전달하는 목적을 가지고 있습니다.

데이터 분할:

생성된 데이터 처리 요청에 따라 라우터는 수신된 음성 데이터를 작은 블록 또는 프레임으로 분할합니다. 일반적으로 이러한 블록의 크기는 일정한 길이를 가지며, 작업의 효율성과 데이터 처리 속도를 고려하여 결정됩니다.

분할된 데이터 전송:

분할된 데이터는 각각의 프로세스로 전송됩니다. 이때, 라우터는 데이터를 효율적으로 분배하여 각 프로세스가 동시에 데이터를 받을 수 있도록 합니다. 이는 시스템의 전체 처리량을 극대화하기 위한 중요한 단계입니다.

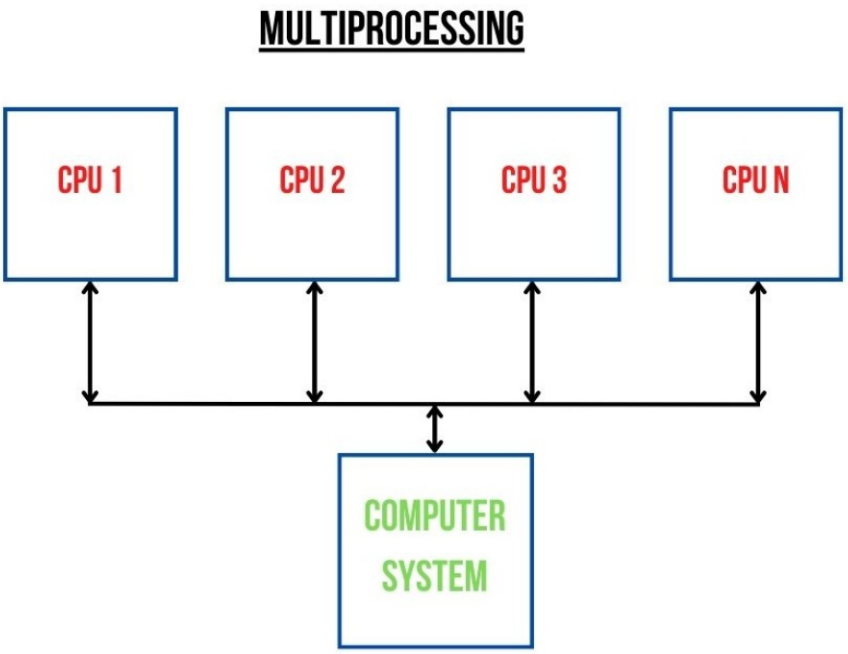
프로세스 활성화:

데이터가 전송되면, 각각의 프로세스는 받은 데이터를 처리하기 위해 활성화됩니다. 각 프로세스는 자신에게 할당된 데이터 블록을 처리하고, 그 결과를 다음 단계로 전달합니다.

병렬 처리 및 결과 수집:

각각의 프로세스는 동시에 데이터 블록을 처리하며, 병렬 처리를 통해 시스템의 성능을 극대화합니다. 각 프로세스가 처리한 결과는 결과를 수집하는 단계에서 통합되어 최종적인 결과를 생성합니다.

3. 멀티프로세싱을 위한 데이터 처리:



프로세스 생성:

라우터를 통해 분할된 데이터 블록을 처리할 각각의 프로세스를 생성합니다. 이때, 프로세스의 개수는 시스템의 자원 및 병렬 처리의 효율성을 고려하여 결정됩니다.

프로세스 초기화:

각각의 프로세스는 데이터 처리를 위한 초기화 작업을 수행합니다. 이는 필요한 라이브러리 및 리소스를 로드하고, 작업에 필요한 변수 및 데이터 구조를 초기화하는 등의 작업을 포함합니다.

데이터 수신 및 처리:

각 프로세스는 라우터를 통해 분할된 데이터 블록을 수신합니다. 이후, 프로세스는 받은 데이터를 처리하여 원하는 작업을 수행합니다. 이는 음성 인식, 텍스트 요약 등의 작업을 포함할 수 있습니다.

병렬 처리:

각각의 프로세스는 독립적으로 데이터를 처리하며, 병렬로 실행됩니다. 이를 통해 시스템의 전체 처리량을 높일 수 있으며, 음성 데이터의 처리 속도를 향상시킬 수 있습니다.

결과 수집 및 통합:

각 프로세스가 처리한 결과는 결과를 수집하는 단계에서 통합됩니다. 이를 통해 각각의 프로세스가 생성한 결과를 종합하여 최종적인 결과를 생성할 수 있습니다.
프로세스 종료:

모든 데이터가 처리되고 결과가 통합되면, 각각의 프로세스는 종료됩니다.

4. 음성 인식 및 텍스트 변환

음성 데이터 수신:

각각의 프로세스는 라우터를 통해 분할된 음성 데이터를 수신합니다. 이 음성 데이터는 사용자의 음성 발화를 담고 있습니다.

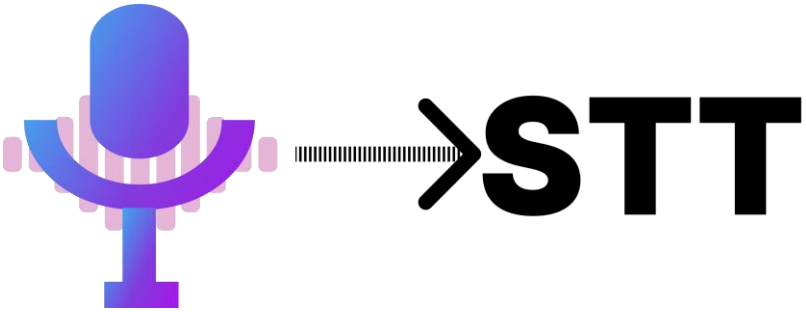
Kaldi 호출:

각각의 프로세스는 받은 음성 데이터를 Kaldi 음성 인식 엔진에 전달합니다.

음성 데이터 처리:

Kaldi 는 받은 음성 데이터를 분석하고, 해당 음성에 대한 텍스트 정보를 추출합니다. 이를 위해 Kaldi 는 음성 신호를 주파수 영역으로 변환하여 음성의 특징을 추출하고, 이를 텍스트로 변환하는 작업을 수행합니다.

텍스트 변환:

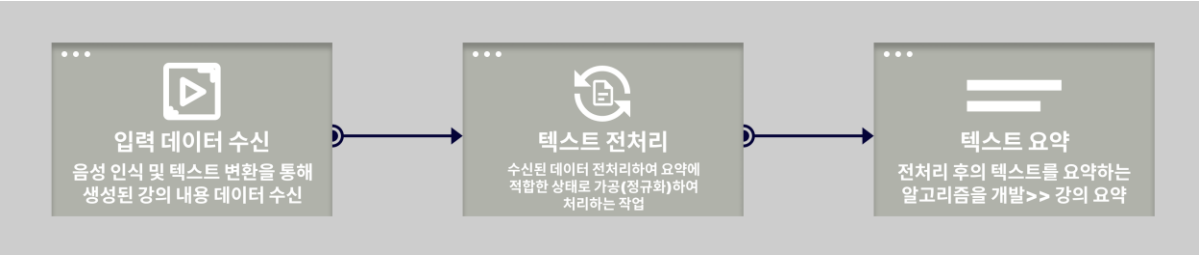


Kaldi 는 추출된 음성 정보를 텍스트로 변환합니다. 이렇게 변환된 텍스트는 원본 음성 발화의 내용을 담고 있습니다.

결과 출력:

텍스트로 변환된 결과는 프로세스에서 생성되고, 이를 필요에 따라 저장하거나 다른 시스템으로 전달합니다. 이렇게 변환된 텍스트는 다음 단계에서 요약 알고리즘에 입력으로 사용될 수 있습니다.

2. 요약 알고리즘 구현



개발 개요

음성 인식 및 텍스트 변환 이후 추가

입력 데이터 수신:

음성 인식 및 텍스트 변환 과정에서 생성된 강의 내용을 포함한 텍스트 데이터를 수신합니다.

텍스트 전처리:

수신된 텍스트 데이터를 전처리하여 요약에 적합한 형태로 가공합니다. 이 과정에는 문장 토큰화, 단어 토큰화, 불용어 제거, 특수 문자 제거 등이 포함됩니다. 또한, 텍스트 데이터를 정규화하여 단어의 형태를 변형하여 처리하는 등의 작업이 이루어집니다.

텍스트 요약:

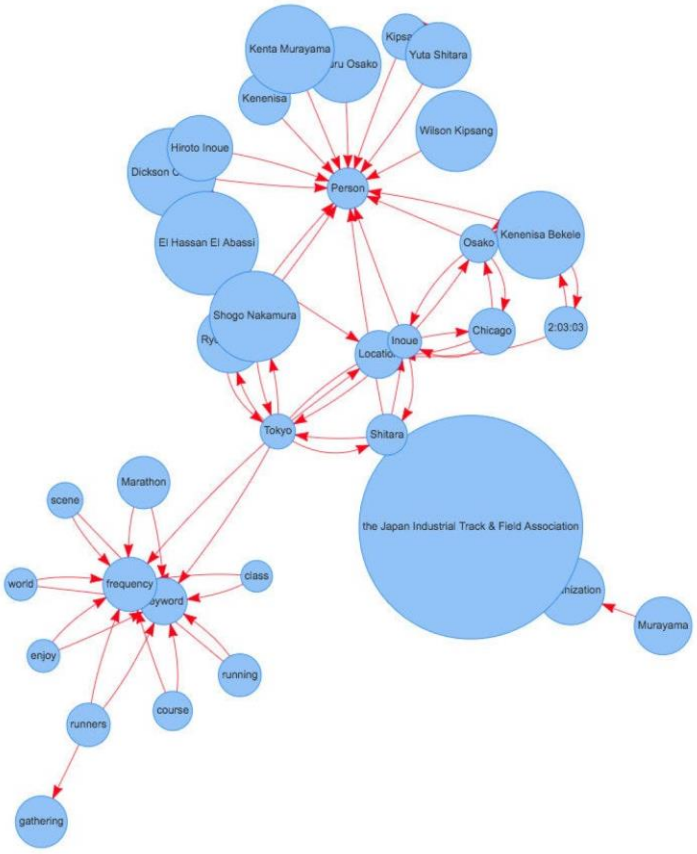
전처리 후의 텍스트를 요약하는 알고리즘을 개발합니다. 알고리즘을 통하여 전달 받은 강의 내용을 요약합니다

※개발 단계

1. 알고리즘 비교 분석

추출적 요약 알고리즘들의 비교 분석

TextRank:



장점:
그래프 기반의 추출적 요약 알고리즘으로, 문장 간의 관계를 고려하여 중요한 문장을 추출합니다.

문서 내에서 문장의 중요도를 그래프의 랭크를 통해 계산하므로 간단하고 직관적입니다.
또한 문서의 구조를 고려하여 요약 결과를 생성하므로 자연스러운 문장을 얻을 수 있습니다.

단점:

그래프를 구성하고 중요도를 계산하기 위한 계산량이 크므로 대규모 문서에는 적용하기 어려울 수 있습니다.
문장 간의 관계를 제대로 반영하지 못할 수 있으며, 문맥을 고려하지 않을 수 있습니다.

TF-IDF (Term Frequency-Inverse Document Frequency):

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

장점:

각 단어의 중요성을 평가하여 문서 내에서 중요한 단어를 추출합니다.
그래서 간단하고 직관적인 방법으로, 계산량이 적고 빠르게 결과를 얻을 수 있습니다.
또한 단어의 빈도와 역 문서 빈도를 고려하여 중요한 단어를 추출하므로, 문서의 핵심을 잘 파악할 수 있습니다.

단점:

단어 수준의 추출이므로 문장의 의미나 문맥을 고려하지 않을 수 있습니다.
문서의 길이가 길어질수록 흔한 단어가 중요하다고 평가될 수 있으며, 이는 요약의 품질을 저하시킬 수 있습니다.

Text Contraction:

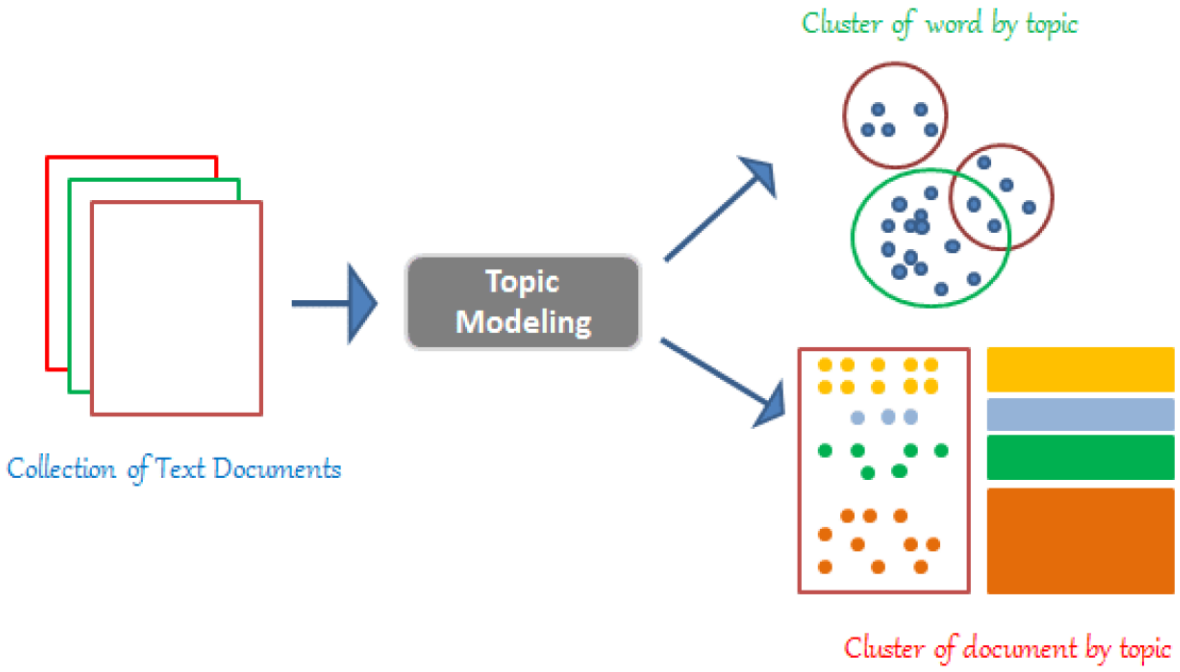
장점:

문장 내에서 중요하지 않은 정보를 제거하여 문장의 길이를 줄이는 방법입니다.
그렇기 때문에 간단하고 직관적인 방법으로, 문장 내의 불필요한 단어나 구절을 쉽게 식별하여 제거할 수 있습니다.
그리고 문장의 요약과 함께 중요한 정보를 유지하면서 문장을 간결하게 만들어줍니다.

단점:

일부 중요한 정보를 잃을 수 있으며, 문장의 의미가 왜곡될 수 있습니다.
모든 문장에 일괄적으로 적용되는 방법이기 때문에 문서의 특성에 따라 적절하지 않을 수 있습니다.

LDA (Latent Dirichlet Allocation):



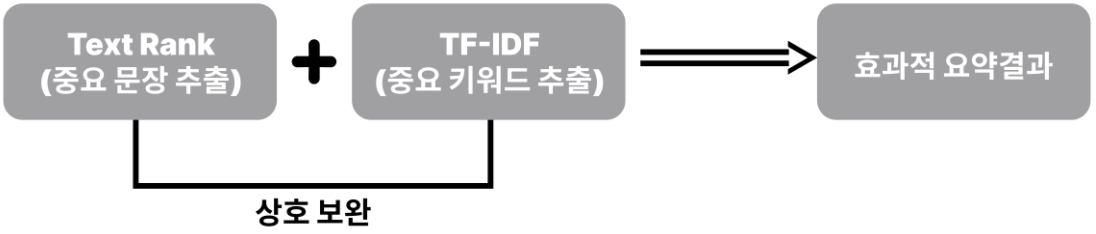
장점:

토픽 모델링 알고리즘으로, 숨겨진 주제를 식별하여 문서의 주제를 파악할 수 있습니다.
문서 내에서 각 주제의 확률을 계산하여 주제와 관련된 문장을 추출할 수 있습니다.
또한 문서의 구조를 고려하여 관련성 있는 문장을 선택하므로 요약의 품질이 높을 수 있습니다.

단점:

모델링에 사용되는 매개변수 설정이나 주제의 수를 결정하는 것이 어려울 수 있습니다.
상당한 계산량이 요구되며, 대규모 문서에 적용하기에는 계산 비용이 높을 수 있습니다.
토픽 모델링 자체가 해석이 어려운 경우가 있으며, 어떤 주제가 추출되었는지 이해하기 어려울 수 있습니다.

2. 알고리즘 결합 계획



더 개선 된 알고리즘 개발을 위해 알고리즘 결합하고자 합니다.

중요한 문장을 추출하는 TextRank 와 중요한 키워드를 추출하는 TF-IDF 를 결합 함으로서

중요한 내용을 중점으로 효과적인 요약 결과를 얻을 수 있습니다

텍스트 랭크는 문장 간의 관계를 고려하여 중요 문장 추출 TF 는 문장 내 중요한 단어를 추출하여 요약 알고리즘이므로

두 알고리즘이 서로 보완하는 특성을 가지고 있어, 결합함으로써 요약의 전체적인 품질을 향상시킬 수 있습니다

알고리즘은 외부 api 로 작동 할 때가 파이 내부에서 동작하는 경우보다 리소스 사용이 훨씬 적고
음성 인식 알고리즘을 온전히 파이에서 사용가능하기 때문에 속도 면에서 더 빠릅니다

TEXTRANK 의 계산량이 크다는 단점은 백 페이지 이상의 대규모 문서에만 국한 되는 한계로 강의 요약 같은 경우 더 적은 양의 텍스트를 요약 하기 때문에 계산량의 단점이 희석됩니다.

3. 데이터 수집 및 전처리

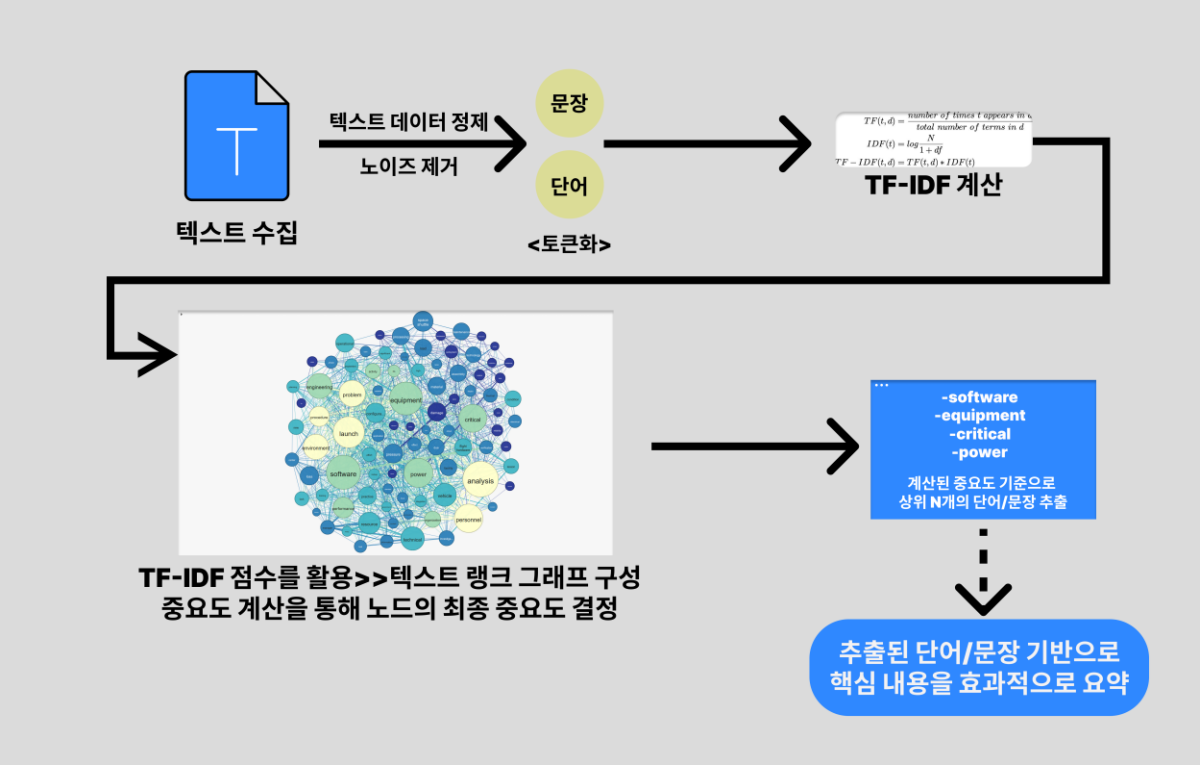
데이터 수집:

오픈 데이터 포털이나 인터넷 강의, 무료 강의 등을 통해 텍스트 데이터셋을 수집합니다.

데이터 전처리:

수집한 텍스트 데이터를 정제하고, 노이즈를 제거하여 데이터의 품질을 향상시킵니다. 또한, 문장 또는 단어로 토큰화합니다. 이 과정에서 TF-IDF 를 사용하여 각 단어의 중요도를 계산합니다.

4. 텍스트 요약을 위한 알고리즘 동작



TF-IDF 계산:

TF-IDF 를 사용하여 각 단어의 중요도를 계산합니다. TF-IDF 는 각 단어의 문서 내에서의 빈도수와 전체 문서 집합에서의 빈도수에 대한 정보를 결합하여 단어의 중요성을 평가합니다.
이렇게 계산된 TF-IDF 점수는 각 단어의 중요도를 나타내는 척도로 활용됩니다.

텍스트 랭크 그래프 생성:

TF-IDF 점수를 활용하여 텍스트 랭크 그래프를 생성합니다. 각 단어를 노드로, TF-IDF 점수를 가중치로 하는 그래프를 구성합니다.

그래프 순환 및 중요도 계산:

텍스트 랭크 알고리즘을 사용하여 그래프를 순환하고, 각 단어(또는 문장)의 중요도를 계산합니다.
TF-IDF 로 계산된 중요도와 텍스트 랭크에서 계산된 중요도를 종합적으로 고려하여 각 노드의 최종 중요도를 결정합니다.

중요한 단어 또는 문장 추출:

계산된 중요도를 기준으로 상위 N 개의 단어 또는 문장을 추출합니다.
TF-IDF 와 텍스트 랭크의 결합으로 인해 중요한 단어와 문장이 강조되며, 요약에 필요한 핵심 내용을 보다 효과적으로 추출할 수 있습니다.

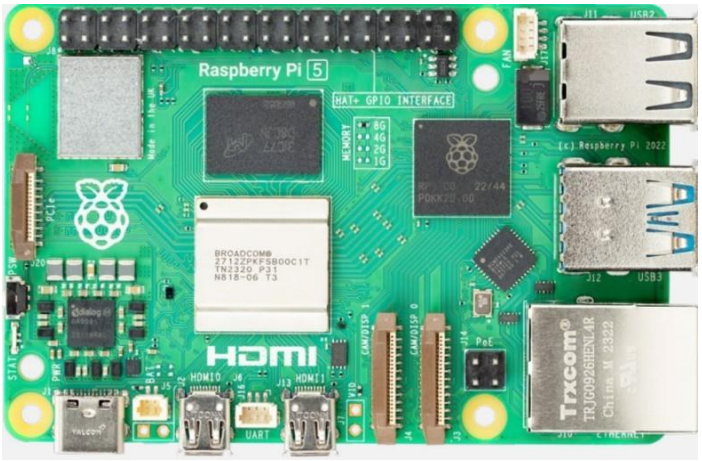
요약 결과 생성:

추출된 중요한 단어 또는 문장들을 조합하여 요약 결과를 생성합니다.
이를 통해 원본 텍스트의 핵심 내용을 보존하면서도 간결하게 요약된 결과를 얻을 수 있습니다.

3. 개발 계획 및 기자재

- 제품 구현 시 필요한 하드웨어 부품

(1) 라즈베리파이 보드



라즈베리파이 5

148,170 **원**

(2) 버튼스위치

[YwRobot] 아두이노 버튼 스위치 모듈 **v1 (WHITE) [ELB060671]**

1.500 원

아두이노 버튼 스위치 모듈 v1 (WHITE) [ELB060671] / 디바이스마트 (devicemart.co.kr)

(3) 마이크로폰 센서

[Seeed] ReSpeaker 4-Mics Linear array Kit for Raspberry Pi

40,400 원

ReSpeaker 4-Mic Linear Array Kit for Raspberry Pi [107990056] / 디바이스마트 (devicemart.co.kr)

- 팀원 간 업무 내용

길진성

회의록 작성
자재 구입
제품 기획
문서 요약 알고리즘 개발

김현욱

제품 기획
문서 요약 알고리즘 개발
모델 학습

김태성

제품 기획
회로 구현
문서 요약 알고리즘 개발
모델 학습

정선진

제품 기획
회로 구현
모델 학습
문서 요약 알고리즘 개발

- 작업 제작 추진 계획 및 일정

5 주	6 주	7 주	8 주
물품 구입 및 자세한 계획 수립 & 구조 장치의 하드웨어적 설계도(디자인)제작	강의 음성 데이터셋을 이용해 음성 인식 AI 학습		중간 고사 기간

9 주	10 주	11 주	12 주	13 주	14 주	15 주	16 주
문서 요약 알고리즘 개발				작동 여부 테스트와 성능 향상			

- 지원 경비 사용 계획

NO.	물품 이름	금액
1	***라즈베리파이 5***	148,170 원
2	[YwRobot] **아두이노 버튼 스위치 모듈** v1 (WHITE) [ELB060671]	1,500 **원**
3	[Seeed] ReSpeaker 4-Mics Linear array Kit for Raspberry Pi	40,400 원
4		
5		
6		
7		
	합계	190,070 원