

# Whisper 주요 코드 구조

☰ 태그	
👤 참석자	📄 PARKSEONGJUN 은지 하 기흥 김

## 파일 구조 및 주요 파일(Whisper/)

### 1. `whisper/`

- Whisper 모델과 관련된 주요 소스 코드를 포함한 디렉토리

### 2. `whisper/__init__.py`

- 패키지 초기화 파일로, `whisper` 디렉토리를 패키지로 인식시키고 초기 설정을 수행

### 3. `whisper/decoding.py`

- 음성 인식 Decoding 코드. 입력된 음성 데이터를 텍스트로 변환하는 과정을 수행

### 4. `whisper/model.py`

- Whisper 모델의 구조와 관련된 클래스와 함수 정의. 모델의 아키텍처와 레이어 구성 등을 포함

### 5. `whisper/tokenizer.py`

- Text to Token, Token to Text 변환 기능을 담당하는 토큰나이저 관련 코드

### 6. `whisper/transcribe.py`

- Speech Data(Audio Data) To Text 변환 메인 함수를 포함한 파일. 파일 경로를 입력으로 받아 음성 인식을 수행

### 7. `whisper/utils.py`

- 유틸리티 함수. 로깅, 시간 측정, 파일 입출력 등의 보조 기능을 수행

### 8. `whisper/audio.py`

- Audio Data Processing 코드. 오디오 파일을 불러오거나 변환하는 기능을 수행

### 9. `whisper/align.py`

- 음성 인식 결과와 원본 텍스트를 정렬하는 기능을 수행

### 10. `requirements.txt`

- 프로젝트에 필요한 Python 패키지 명시 (pip)

## 주요 파일

#### whisper/decoding.py

- **디코딩 클래스와 함수들:** 음성 데이터를 디코딩하여 텍스트로 변환하는 역할을 합니다. 다양한 디코딩 전략 (예: 그리디 디코딩, 빔 서치)이 포함되어 있을 수 있습니다.

#### whisper/model.py

- **WhisperModel 클래스:** Whisper 모델의 전체 구조를 정의합니다. 여기에는 모델의 레이어, 순전파 및 역전파 함수, 모델 초기화 등이 포함됩니다.
- **블록 및 레이어 정의:** 트랜스포머 블록, 어텐션 메커니즘 등 모델의 각 구성 요소가 정의되어 있습니다.

#### whisper/tokenizer.py

- **Tokenizer 클래스:** 텍스트를 토큰으로, 토큰을 텍스트로 변환하는 기능을 제공합니다. BPE(Byte Pair Encoding) 같은 토큰나이제이션 방법을 사용할 수 있습니다.

#### whisper/transcribe.py

- **transcribe 함수:** 음성 파일을 입력으로 받아 텍스트로 변환하는 메인 함수입니다. 오디오 파일을 읽고, 전처리하며, 모델을 통해 텍스트로 변환하는 전체 과정을 처리합니다.

#### whisper/utils.py

- **유틸리티 함수들:** 로깅, 시간 측정, 파일 입출력 등의 기능을 제공하여 메인 코드의 가독성과 유지보수성을 높입니다.

#### whisper/audio.py

- **오디오 처리 함수들:** 오디오 파일을 로드하고 전처리하는 기능을 제공합니다. 예를 들어, 샘플링 레이트를 변경하거나 오디오 파일을 정규화하는 작업 등이 포함됩니다.

## 주요 파일 용도

#### whisper/\_\_init\_\_.py

- **패키지 초기화:** 이 파일은 `whisper` 디렉토리를 패키지로 인식시킵니다. 이 파일이 존재하면 `whisper` 디렉토리는 Python 패키지가 되며, 다른 코드에서 `import whisper` 로 이 패키지를 사용할 수 있게 됩니다.

#### whisper/decoding.py

- **디코딩 로직:** 이 파일에는 음성 인식 결과를 디코딩하는 함수들이 포함되어 있습니다. 주요 역할은 모델이 출력하는 확률 분포를 실제 텍스트로 변환하는 것입니다. 일반적인 디코딩 방법으로는 그리디 디코딩, 빔 서치 등이 있습니다.
  - **Greedy Decoding:** 각 단계에서 가장 높은 확률을 가진 토큰을 선택합니다.
  - **Beam Search Decoding:** 여러 경로를 고려하면서 가장 높은 확률을 가진 경로를 선택합니다.

#### whisper/model.py

- **모델 정의:** 이 파일은 Whisper 모델의 전체 아키텍처를 정의합니다. 트랜스포머 기반의 모델 구조를 사용하며, 다양한 레이어와 블록으로 구성됩니다.
  - **WhisperModel 클래스:** 모델의 전반적인 구조를 정의하는 클래스입니다.
  - **Layer Definitions:** 트랜스포머 블록, 어텐션 메커니즘, 피드포워드 네트워크 등의 구성 요소가 포함됩니다.
  - **Forward Pass:** 입력 데이터를 받아서 모델을 통해 순전파를 수행하는 함수가 포함되어 있습니다.

#### whisper/tokenizer.py

- **토크나이저:** 텍스트 데이터를 모델에서 사용할 수 있는 토큰 형태로 변환하고, 모델의 출력을 텍스트로 변환하는 기능을 제공합니다.
  - **Tokenizer 클래스:** 텍스트를 토큰으로 변환하고, 토큰을 다시 텍스트로 변환하는 기능을 담당합니다.
  - **Vocabulary Management:** 모델이 사용하는 어휘 사전을 관리하고, 각 단어에 고유의 토큰 ID를 부여합니다.

#### whisper/transcribe.py

- **음성 파일 변환:** 음성 파일을 텍스트로 변환하는 메인 함수가 포함된 파일입니다.
  - **Transcribe Function:** 주어진 음성 파일을 불러와서, 모델을 통해 텍스트로 변환하는 전체 과정을 수행합니다.
  - **Preprocessing:** 오디오 데이터를 전처리하고 모델에 입력할 수 있는 형태로 변환합니다.
  - **Inference:** 모델을 사용하여 음성 데이터를 텍스트로 변환합니다.
  - **Postprocessing:** 모델의 출력을 후처리하여 최종 텍스트 결과를 생성합니다.

#### whisper/utils.py

- **유틸리티 함수들:** 다양한 보조 기능을 제공하여 메인 코드의 가독성과 유지보수성을 높입니다.
  - **Logging:** 코드 실행 중 발생하는 로그를 기록하는 기능을 제공합니다.
  - **Time Measurement:** 코드 실행 시간을 측정하는 함수들이 포함되어 있습니다.
  - **File I/O:** 파일 읽기 및 쓰기 관련 함수들이 포함되어 있습니다.

#### whisper/audio.py

- **오디오 처리:** 오디오 데이터를 처리하는 함수들이 포함되어 있습니다.
  - **Load Audio:** 오디오 파일을 불러오는 함수입니다.
  - **Resampling:** 오디오 샘플링 레이트를 변경하는 기능을 제공합니다.
  - **Normalization:** 오디오 데이터를 정규화하여 일정한 음량을 유지하도록 합니다.

## whisper/align.py

- **텍스트 정렬:** 음성 인식 결과와 원본 텍스트를 정렬하는 기능을 제공합니다.
  - **Alignment Algorithms:** 두 텍스트 시퀀스를 비교하여 유사한 부분을 정렬하는 알고리즘이 포함되어 있습니다.
  - **Error Metrics:** 인식 결과와 원본 텍스트의 일치도를 평가하는 지표를 제공합니다.

## Inference Flow

1. 음성 파일 불러오기 ( `audio.py` ):
  - 오디오 파일을 불러와서 필요한 전처리를 수행합니다.
2. 모델 준비 ( `model.py` ):
  - Whisper 모델을 로드하고, 필요한 경우 초기화 및 설정을 합니다.
3. 텍스트 변환 준비 ( `tokenizer.py` ):
  - 텍스트를 토큰으로 변환하거나 토큰을 텍스트로 변환하기 위해 토큰나이지어를 설정합니다.
4. 음성 인식 수행 ( `transcribe.py` ):
  - 오디오 데이터를 입력으로 받아 모델을 통해 텍스트로 변환합니다.
  - 전처리된 오디오 데이터를 모델에 입력하여 인퍼런스를 수행하고, 결과를 디코딩합니다.
5. 결과 처리 및 저장 ( `utils.py` ):
  - 인식 결과를 후처리하고, 필요한 경우 로그를 기록하며, 최종 텍스트를 저장합니다.

## Inference Structure

단계	설명	파일	주요 함수/클래스
1	오디오 파일 불러오기	<code>whisper/audio.py</code>	<code>load_audio(filepath)</code>
2	모델 로드	<code>whisper/model.py</code>	<code>WhisperModel.from_pretrained(model_name)</code>
3	토큰나이지어 준비	<code>whisper/tokenizer.py</code>	<code>Tokenizer.load(vocab_path)</code>
4	오디오 데이터 전처리	<code>whisper/audio.py</code>	<code>preprocess_audio(audio_data)</code>
5	모델에 입력	<code>whisper/transcribe.py</code>	<code>model.transcribe(audio_data)</code>
6	디코딩	<code>whisper/decoding.py</code>	<code>decode(logits)</code>
7	후처리	<code>whisper/transcribe.py</code>	<code>postprocess_text(decoded_text)</code>
8	결과 저장 및 반환	<code>whisper/utils.py</code>	<code>save_transcription(output_path, text)</code>

### 1. 오디오 파일 불러오기 ( `whisper/audio.py` )

- 함수: `load_audio(filepath)`
- 설명: 지정된 경로에서 오디오 파일을 불러오고, 모델이 처리할 수 있는 형식으로 변환합니다.

### 2. 모델 로드 ( `whisper/model.py` )

- 함수/클래스: `WhisperModel.from_pretrained(model_name)`
- 설명: 사전 학습된 Whisper 모델을 로드합니다. 다양한 모델 변형을 로드할 수 있습니다.

### 3. 토큰나이저 준비 ( `whisper/tokenizer.py` )

- 함수/클래스: `Tokenizer.load(vocab_path)`
- 설명: 텍스트를 토큰으로 변환하고, 반대로 토큰을 텍스트로 변환하는 토큰나이저를 준비합니다.

### 4. 오디오 데이터 전처리 ( `whisper/audio.py` )

- 함수: `preprocess_audio(audio_data)`
- 설명: 오디오 데이터를 모델에 입력할 수 있는 형태로 전처리합니다. 여기에는 샘플링 레이트 변경, 정규화 등이 포함됩니다.

### 5. 모델에 입력 ( `whisper/transcribe.py` )

- 함수: `model.transcribe(audio_data)`
- 설명: 전처리된 오디오 데이터를 모델에 입력하여 텍스트로 변환합니다.

### 6. 디코딩 ( `whisper/decoding.py` )

- 함수: `decode(logits)`
- 설명: 모델이 출력한 로그잇(logits)을 실제 텍스트로 변환합니다. 그리디 디코딩, 빔 서치 등의 방법이 사용됩니다.

### 7. 후처리 ( `whisper/transcribe.py` )

- 함수: `postprocess_text(decoded_text)`
- 설명: 디코딩된 텍스트를 후처리하여 최종 결과를 생성합니다. 여기에는 불필요한 공백 제거, 문장 부호 추가 등이 포함될 수 있습니다.

### 8. 결과 저장 및 반환 ( `whisper/utils.py` )

- 함수: `save_transcription(output_path, text)`
- 설명: 최종 텍스트 결과를 지정된 경로에 저장하거나 반환합니다.

---

## Whisper Fine-Tuning

- Video

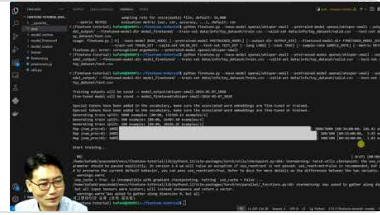
## [전이학습+딥러닝 서비스]\_21. (Finetune) Whisper 파인튜닝 학습 수행(실제 학습)

강의목록(자료):

[https://github.com/kafa46/acin\\_academy/tree/master/202\\_fine\\_tunning/](https://github.com/kafa46/acin_academy/tree/master/202_fine_tunning/)


꼰대 교수님 홈페이지: <https://prof.acin.kr/>

 <https://www.youtube.com/watch?v=de49OqoS5yU>



### • Code

acin\_academy/202\_fine\_tunning/whisper at master · kafa46/acin\_academy  
acin\_academy. Contribute to kafa46/acin\_academy development by creating an account on GitHub.

 [https://github.com/kafa46/acin\\_academy/tree/master/202\\_fine\\_tunning/whisper](https://github.com/kafa46/acin_academy/tree/master/202_fine_tunning/whisper)

kafa46/  
**acin\_academy**

acin\_academy



1 Contributor 0 Issues 6 Stars 3 Forks