



Documento: TranscriptAudio

1. Introdução

A classe TranscriptAudio é responsável por converter arquivos de áudio MP3 para WAV e transcrever o conteúdo de arquivos de áudio WAV usando o reconhecimento de fala Sphinx4. Esta classe utiliza as bibliotecas FFmpeg e JavaCV para a conversão de formatos e Sphinx4 para a transcrição de áudio.

2. Estrutura da Classe

java

package transcript.audio;

*import edu.cmu.sphinx.api.Configuration;
import edu.cmu.sphinx.api.StreamSpeechRecognizer;
import edu.cmu.sphinx.api.SpeechResult;
import org.bytedeco.javacv.FFmpegFrameGrabber;
import org.bytedeco.javacv.FFmpegFrameRecorder;
import org.bytedeco.ffmpeg.global.avcodec;*

*import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;*

public class TranscriptAudio {

*// Converte MP3 para WAV
public static void convertMp3ToWav(String mp3FilePath, String wavFilePath) throws Exception {
 try (FFmpegFrameGrabber grabber = new FFmpegFrameGrabber(mp3FilePath);
 FFmpegFrameRecorder recorder = new FFmpegFrameRecorder(wavFilePath, 1)) {
 grabber.setAudioChannels(1);
 grabber.start();
 recorder.setAudioCodec(avcodec.AV_CODEC_ID_PCM_S16LE);
 recorder.setSampleRate(grabber.getSampleRate());
 recorder.setAudioChannels(grabber.getAudioChannels());
 recorder.setFormat("wav");
 recorder.start();*

*// Ler e gravar todos os samples
while (true) {
 var samples = grabber.grabSamples();
 if (samples == null) break;
 recorder.record(samples);
}*

*recorder.stop();
grabber.stop();*



```
}  
}  
  
// Transcreve o áudio WAV usando Sphinx4  
public static void transcribeAudio(String audioFilePath, String outputTextFilePath) throws Exception {  
    // Verifica se o arquivo WAV existe  
    File audioFile = new File(audioFilePath);  
    if (!audioFile.exists()) {  
        throw new FileNotFoundException("Arquivo não encontrado: " + audioFilePath);  
    }  
  
    // Configuração do Sphinx4  
    Configuration configuration = new Configuration();  
    configuration.setAcousticModelPath("resource:/edu/cmu/sphinx/models/en-us/en-us");  
    configuration.setDictionaryPath("resource:/edu/cmu/sphinx/models/en-us/cmudict-en-us.dict");  
    configuration.setLanguageModelPath("resource:/edu/cmu/sphinx/models/en-us/en-us.lm.bin");  
  
    // Inicializa o reconhecedor de fala  
    try (InputStream inputStream = new FileInputStream(audioFilePath);  
        OutputStream outputStream = new FileOutputStream(outputTextFilePath)) {  
  
        StreamSpeechRecognizer recognizer = new StreamSpeechRecognizer(configuration);  
        recognizer.startRecognition(inputStream);  
  
        // Transcreve o áudio  
        SpeechResult result;  
        while ((result = recognizer.getResult()) != null) {  
            outputStream.write(result.getHypothesis().getBytes());  
            outputStream.write("\n".getBytes());  
        }  
  
        recognizer.stopRecognition();  
    }  
}  
  
public static void main(String[] args) {  
    String mp3FilePath = "teste.mp3";  
    String wavFilePath = "audio.wav";  
    String outputTextFilePath = "Output.txt";  
  
    try {  
        // Converte MP3 para WAV  
        convertMp3ToWav(mp3FilePath, wavFilePath);  
  
        // Transcreve o áudio WAV  
        transcribeAudio(wavFilePath, outputTextFilePath);  
        System.out.println("Transcrição concluída com sucesso.");  
    } catch (Exception e) {  
        System.err.println("Erro ao transcrever o áudio: " + e.getMessage());  
        e.printStackTrace();  
    }  
}
```

2.1. convertMp3ToWav(String mp3FilePath, String wavFilePath)

- **Descrição:** Este método converte um arquivo MP3 para o formato WAV. Utiliza FFmpegFrameGrabber para ler o áudio MP3 e FFmpegFrameRecorder para gravar o áudio no formato WAV.
- **Parâmetros:**
 - mp3FilePath - Caminho do arquivo MP3 de entrada.
 - wavFilePath - Caminho do arquivo WAV de saída.
- **Exceções Lançadas:**
 - Exception - Caso ocorra algum erro durante o processo de conversão.



2.2. transcribeAudio(String audioFilePath, String outputTextFilePath)

- **Descrição:** Este método transcreve o conteúdo de um arquivo de áudio WAV para um arquivo de texto. Configura o reconhecedor de fala Sphinx4 com o modelo acústico, o dicionário e o modelo de linguagem apropriados e escreve o resultado da transcrição em um arquivo de texto.
- **Parâmetros:**
 - audioFilePath - Caminho do arquivo WAV de entrada.
 - outputTextFilePath - Caminho do arquivo de texto de saída.
- **Exceções Lançadas:**
 - FileNotFoundException - Se o arquivo WAV de entrada não for encontrado.
 - Exception - Caso ocorra algum erro durante o processo de transcrição.

2.3. main(String[] args)

- **Descrição:** O método principal define os caminhos dos arquivos MP3, WAV e de saída. Chama os métodos convertMp3ToWav e transcribeAudio para realizar a conversão e transcrição do áudio, respectivamente. Trata exceções e exibe mensagens de erro ou sucesso.

3. Dependências

- FFmpegFrameGrabber e FFmpegFrameRecorder do JavaCV para conversão de formatos de áudio.
- StreamSpeechRecognizer do Sphinx4 para reconhecimento de fala.

4. Conclusão

A classe TranscriptAudio fornece uma solução eficiente para converter e transcrever arquivos de áudio. A utilização das bibliotecas JavaCV e Sphinx4 permite a integração direta com processos de reconhecimento de fala e manipulação de arquivos de áudio.