Aplikacja do automatycznego generowania napisów do filmów i seriali na podstawie analizy audio i wideo.

Projekt z przedmiotu

Automaty, Języki i Obliczenia

Maciej Włosek

Spis treści

1. Wstęp

1.1 Cel projektu

Projekt ma na celu opracowanie aplikacji, która wykorzystuje techniki analizy audio i wideo do automatycznego generowania napisów dla mediów audiowizualnych. Aplikacja pozwoli na oszczędność czasu i zwiększenie efektywności tworzenia napisów. Dodatkowo aplikacja umożliwia wsparcie w tłumaczeniu pomiędzy różnymi językami oraz dostarcza napisy dla osób niesłyszących. Pozwala ona również na szybsze edytowanie wielkości, grubości czy też koloru liter. Będzie ona też pozwalała na wybranie dowolnego motywu czcionki.

2. Wybór technologiczny

2.1 Wybór technologii i narzędzi(node.js):

Node.js: Jest to środowisko wykonawcze JavaScript, które umożliwia uruchomienie aplikacji serwerowych opartych na JavaScript. W tym projekcie używamy Node.js do tworzenia aplikacji w środowisku backendowym.

2.2 fs(File System):

Jest to moduł wbudowany w Node.js, który umożliwia interakcję z systemem plików. Wykorzystujemy ten moduł do operacji na plikach, takich jak odczyt, zapis i usuwanie plików.

2.3 ffmpeg:

To narzędzie do manipulacji multimediów, które umożliwia konwersję, edycję i przetwarzanie plików wideo i audio. W tym projekcie wykorzystujemy bibliotekę „fluent-ffmpeg” jako interfejs do obsługi funkcji ffmpeg.

2.4 @google-cloud/speech:

Jest to biblioteka klienta dla usługi rozpoznawania mowy firmy Google Cloud. Wykorzystujemy ją do przetwarzania plików audio i generowania transkrypcji mowy.

2.5 @ffmpeg-installer/ffmpeg:

Jest to paczka npm zawierająca binarne pliki ffmpeg. Używamy tej paczki do ustawienia ścieżki do ffmpeg w naszej aplikacji.

2.6 deepl-translator:

Jest to moduł do tłumaczenia za pomocą usługi DeepL. DeepL to zaawansowana usługa tłumaczeniowa, która oferuje wysoką jakość tłumaczeń między wieloma językami.

3.Schemat i działanie aplikacji

1. Wprowadzenie danych wejściowych:
    - Użytkownik dostarcza plik wideo, który ma zostać przetworzony.
    - Aplikacja przyjmuje ścieżkę do pliku wideo jako parametr wejściowy.

2. Konwersja wideo na audio:
    - Wykorzystując bibliotekę ffmpeg, aplikacja konwertuje plik wideo na plik audio.
    - Konwersja obejmuje również zmianę formatu audio na odpowiedni dla analizy mowy.

3. Analiza mowy:
    - Wykorzystując usługę rozpoznawania mowy, taką jak Google Cloud Speech-to-Text, aplikacja przesyła plik audio do usługi w celu uzyskania transkrypcji mowy.

- Usługa rozpoznawania mowy zwraca tekstową reprezentację transkrypcji.

4. Generowanie napisów:
   - Na podstawie transkrypcji mowy, aplikacja generuje napisy w formie tekstu.
   - W przypadku dłuższych dialogów, napisy mogą być podzielone na krótsze fragmenty.

5. Tłumaczenie (opcjonalne):
   - Tekst napisów jest przetwarzany przez moduł tłumaczenia, aby uzyskać tłumaczenie w żądanym języku.

6. Generowanie pliku z napisami:
   - Aplikacja tworzy plik z napisami w odpowiednim formacie, na przykład SubRip (SRT), który zawiera numery linii, czas początkowy i końcowy dla każdego napisu oraz tekst napisu.
   - Plik z napisami jest zapisywany w wybranym miejscu na dysku.

7. Dodanie napisów do filmu:
   - Wykorzystując bibliotekę ffmpeg, aplikacja łączy pierwotny plik wideo z plikiem z napisami.
   - Napisy są dodawane do filmu jako warstwa napisów, zachowując oryginalny dźwięk i obraz.

8. Wygenerowanie wynikowego pliku wideo z napisami:
   - Aplikacja generuje nowy plik wideo, który zawiera oryginalny dźwięk i obraz z dodanymi napisami.
   - Wynikowy plik wideo jest zapisywany w wybranym miejscu na dysku.

9. Zakończenie działania aplikacji:
   - Aplikacja informuje użytkownika o zakończeniu procesu generowania napisów i podaje ścieżkę do wynikowego pliku wideo z napisami.

4.Implementacja

4.1 Omówienie funkcji:

```
function generateSubtitles(videoPath, outputPath) {
```

Główna funkcja, która przyjmuje ścieżkę do pliku wideo (videoPath) i ścieżkę do pliku wynikowego z napisami (outputPath). Ta funkcja inicjuje proces generowania napisów dla podanego wideo. Wykorzystuje bibliotekę fluent-ffmpeg do konwersji wideo na plik dźwiękowy, następnie przekazuje ten plik do usługi rozpoznawania mowy firmy Google w celu uzyskania transkrypcji. Ostatecznie generuje plik z napisami w formacie SRT i dodaje napisy do oryginalnego pliku wideo.

```
function calculateDialogueDuration(dialogue) {
```

Funkcja oblicza czas trwania dla danego dialogu na podstawie liczby słów w dialogu. Ustala średnią liczbę słów na minutę (wordsPerMinute) i oblicza czas trwania dialogu na podstawie liczby słów w dialogu oraz tej średniej.

```
function generateSubtitleContent(transcription) {
```

Funkcja generuje zawartość pliku z napisami w formacie SRT na podstawie transkrypcji. Dzieli transkrypcję na linie, a następnie na dialogi. Dla każdego dialogu oblicza czas początkowy i końcowy, formatuje te czasy oraz tekst dialogu i tworzy linie napisów w formacie SRT.

```
function splitIntoDialogues(text) {
```

Funkcja dzieli długi tekst na dialogi. Obecnie tekst jest dzielony na podstawie znaku nowej linii, ale można dostosować tę funkcję w zależności od preferencji i potrzeb.

```
function calculateTextDuration(text) {
```

Funkcja oblicza czas trwania dla danej linii tekstu na podstawie liczby znaków w tekście. Oblicza czas trwania na podstawie średniej liczby znaków na sekundę (charactersPerSecond) i długości tekstu.

```
function formatTime(seconds) {
```

Funkcja formatuje czas w sekundach na format hh:mm:ss,000, który jest używany w plikach SRT. Konwertuje czas w sekundach na godziny, minuty i sekundy, a następnie formatuje je jako tekst, dodając odpowiednie zera na początku

```
function padZero(number) {
```

Funkcja dodaje zero na początku liczby, jeśli jest mniejsza niż 10. Jest używana w funkcji formatTime() do formatowania czasu.

5. Wnioski

- Ważne jest, aby korzystać z odpowiednich modułów i bibliotek, które ułatwiają konkretną funkcjonalność. W tym przypadku, użyto modułów takich jak ffmpeg do konwersji wideo i audio, @google-cloud/speech do rozpoznawania mowy oraz modułu do tłumaczenia (np. google-translate-api), aby rozszerzyć funkcjonalność tłumaczenia napisów.

- Warto zorganizować kod w sposób modułowy, dzieląc go na funkcje o określonych zadaniach. Każda funkcja ma przypisane konkretne zadanie, co ułatwia czytelność i utrzymanie kodu.

- Komunikacja z usługami zewnętrznymi, takimi jak usługa rozpoznawania mowy lub tłumaczenia, odbywa się poprzez ich odpowiednie API i konfiguracje. W przypadku Google Cloud Speech-to-Text, wymagany jest klucz dostępu do API, który można dostarczyć jako parametr podczas inicjalizacji klienta.

6.Propozycje dalszej rozbudowy projektu

- Rozbuduj funkcje analizy mowy, aby uwzględnić inne aspekty, takie jak rozpoznawanie mówcy, detekcję emocji lub identyfikację językach.

- Udoskonal funkcję generowania napisów, na przykład poprzez dodanie lepszej segmentacji dialogów, oznaczanie czasu wystąpienia mówcy lub dodanie formatowania stylu napisów.

- Zaimplementuj interfejs użytkownika, który pozwoli użytkownikowi wybrać plik wideo, określić preferencje generowania napisów (np. język, opcje tłumaczenia) i wyświetlić wynikowy plik wideo z napisami.który można dostarczyć jako parametr podczas inicjalizacji klienta.

- Zoptymalizuj proces generowania napisów poprzez równoległe przetwarzanie wielu fragmentów audio lub wykorzystanie usług chmurowych do przetwarzania w skali.

- Dodaj obsługę innych formatów plików napisów, takich jak VTT (WebVTT) lub XML.

- Wsparcie dla innych silników rozpoznawania mowy. Obecnie kod wykorzystuje usługę Google Cloud Speech-to-Text, ale można rozważyć integrację z innymi dostawcami, takimi jak Amazon Transcribe, Microsoft Azure Speech to Text lub Sphinx.

- Integracja z platformami streamingowymi. Możliwość wykorzystania aplikacji do generowania i dodawania napisów do materiałów wideo na platformach streamingowych, takich jak YouTube, Netflix, Amazon Prime itp.

- Dodanie funkcji edytora napisów. Umożliwienie użytkownikowi edycji wygenerowanych napisów, takich jak poprawa błędów, dostosowywanie czasu wyświetlania napisów, dodawanie znaczników czasowych, formatowania tekstu itp.