

Aplikacja do transkrypcji plików audio na pliki tekstowe w różnych wersjach językowych i odwrotnie.

Skrzecz Przemysław 19419

VIII 2023

Spis treści

Wstęp.....	1
Zamiana mowy na tekst.....	1
Przykładowy kod.....	2
Podsumowanie.....	4
Źródła.....	4

Wstęp

Celem aplikacji jest możliwość przetworzenia plików audio tak, aby głos dostępny na nagraniach został rozpoznany i przetworzony na tekst. Następnie tekst ten przetłumaczyć na wybrane przez użytkownika języki. Ostatnim krokiem jest synteza przetłumaczonych tekstów na pliki audio. Aplikacja pozwala stworzyć w ten sposób pewnego rodzaju lektora do nagrań audio bądź filmów. Przeznaczenie aplikacji to głównie osoby indywidualne, twórcy internetowi, którzy potrzebują lub chcieliby dotrzeć do większej liczby odbiorców. W tym streszczeniu skupię się na przedstawieniu działania zamiany mowy na tekst.

Zamiana mowy na tekst

Zamiana mowy na tekst jest pierwszym i najważniejszym etapem aplikacji, każde złe rozpoznanie na tym etapie spowoduje coraz to większe błędy na następnych etapach. Speech To Text (STT) jest technologią, która umożliwia zamianę dźwięku dokładniej mowy na tekst pisany. Technologia ta wykorzystuje algorytmy sztucznej inteligencji, aby dobrze rozpoznawać mowę. Proces rozpoznawania mowy jest na tyle złożonym procesem, że można o nim napisać książkę i to nie jedną, zainteresowanych lekturą na temat przetwarzania głosu odsyłam do literatury. Osobiście polecam książkę panów D. Jurafsky i J. H. Martin, pod tytułem „Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft Summary of Contents”. Książka ta posiada ogromną ilość wiedzy na temat przetwarzania dźwięku oraz rozpoznawania mowy. W tym streszczeniu pozwolę pokrótce opowiedzieć mechanikę działania bez zagłębiania się w szczegóły. Sztuczna inteligencja wspomniana wcześniej działa na modelach, które uczone są, jak rozpoznawać wzorce dźwiękowe oraz przyporządkowywać im odpowiednie słowa lub frazy.

Proces STT składa się z kilku kroków:

Pierwszy z nich to przetwarzanie sygnału akustycznego polega to na odczytywaniu i analizowaniu sygnału dźwiękowego z mikrofonu lub pliku audio. Sygnał dzielony jest na segmenty zwane ramkami, które poddawane są operacjom matematycznym, takim jak filtracja, normalizacja czy ekstrakcja cech. Matematyka jaka za tym stoi jest oparta na analizie częstotliwościowej sygnału, czyli rozkładaniu go na składowe sinusoidalne o różnych częstotliwościach i amplitudach. Dokładne równania znajdują się w książce podanej powyżej, uważam że bezsensowne kopiowanie ich tutaj, ponieważ i tak będą one nie do zrozumienia bez szerszego kontekstu.

Następny etap to rozpoznawanie fonemów, które polega na identyfikacji podstawowych jednostek dźwiękowych. Fonemy są reprezentowane przez symbole alfabetyczne lub numeryczne. Rozpoznawanie działa na zasadzie porównywania ramki sygnału z wcześniej wyuczoną bazą fonemów.

Trzeci etap to rozpoznawanie słów, polega on na łączeniu fonemów w słowa lub frazy. Rozpoznawanie odbywa się poprzez modele statystyczne, które służą do oceny prawdopodobieństwa wystąpienia danej sekwencji słów w danym języku i kontekście. Najczęściej używanymi modelami są tu ukryte modele Markowa (HMM) i modele językowe oparte na n-gramach.

Ostatnim etapem jest generowanie tekstu, jest to etap, który odpowiada za przekształcenie rozpoznanych słów w tekst pisany. Etap ten może obejmować również dodatkowe operacje takie jak poprawa błędów ortograficznych, interpunkcyjnych czy gramatycznych.

STT napotyka jednak na wiele problemów technologicznych, które wpływają na jakość i dokładność wyników. Niektóre z tych problemów to:

- Zaszumienie i zakłócenia sygnału dźwiękowego, które utrudniają rozróżnienie słów i dźwięków.
- Różnorodność języków, akcentów, dialektów i stylów mówienia, które wymagają dostosowania modeli STT do konkretnych kontekstów i użytkowników.
- Niejednoznaczność i niepoprawność gramatyczna języka mówionego, która sprawia, że trudno jest zinterpretować znaczenie i intencję wypowiedzi.
- Szybkość i zmienność mowy, która wymaga szybkiego i elastycznego przetwarzania sygnału dźwiękowego.

Aby rozwiązać te problemy, STT stosuje różne techniki i strategie, takie jak:

- Użycie filtrów i algorytmów redukcji szumów, aby poprawić jakość sygnału dźwiękowego.
- Użycie wielojęzycznych i domenowych modeli STT, aby uwzględnić różnice językowe i tematyczne.
- Użycie technik uczenia maszynowego i głębokiego uczenia się, aby poprawić dokładność i adaptację modeli STT do danych treningowych i testowych.
- Użycie technik segmentacji i rozpoznawania mówcy, aby zidentyfikować granice słów i osobowości mówiących.
- Użycie technik analizy semantycznej i syntaktycznej, aby zrozumieć znaczenie i strukturę wypowiedzi.

Przykładowy kod

Poniżej znajduje się fragment kodu z aplikacji odpowiedzialny za przetwarzanie mowy na tekst napisane w języku Python wraz z opisem działania.

Na początku importujemy potrzebne biblioteki oraz tworzymy obiekt rozpoznawania mowy sr

```
import os
import speech_recognition as sr
from threading import Thread

r = sr.Recognizer()

def googleSTT(lang):
    countAudioFile = 0
    countTextFile = 0

    for path in os.listdir(r'splitted/audio'):
        if os.path.isfile(os.path.join(r'splitted/audio',path)):
            countAudioFile += 1

    for path in os.listdir(r'splitted/text'):
        if os.path.isfile(os.path.join(r'splitted/text',path)):
            countTextFile += 1

    if countTextFile>0:
        print("There are files in splitted/text directory")
        return

    for i in range(countAudioFile):
        with sr.AudioFile("splitted/audio/"+str(i)+"-audio.wav") as source:
            r.adjust_for_ambient_noise(source)
            audio = r.record(source)

            try:
                result = r.recognize_google(audio,language=lang)

                with open('splitted/text/'+str(i)+'-text.txt','a', encoding='utf8') as
f:
                    f.write(result)

            except sr.UnknownValueError:
                print("Google could not understand audio")

            except sr.RequestError as e:
                print("Google error; {0}".format(e))
```

Funkcja googleSTT przyjmuje jeden parametr lang jest to parametr odpowiadający za język oryginalny nagrania zostanie on wykorzystany przy rozpoznawaniu, tak aby rozpoznawanie prowadzone przez usługę Google nie musiało samo określać języka oryginalnego nagrania. Funkcja sprawdza czy w folderze znajdują się pliki, z których można rozpoznać mowę, jeśli nie funkcja zwraca odpowiednią informację do konsoli oraz przerywa dalszą pracę. Jeśli odpowiednie pliki znajdują się w folderze oraz folder docelowy do którego mają trafić rozpoznania jest pusty, rozpoczyna się proces rozpoznawania mowy. Proces ten jest wykonywany w pętli dla każdego pliku audio znajdującego się w folderze, najlepsze rozpoznanie jest później zapisywane w folderze splitted/text jako plik tekstowy. W razie wystąpienia błędu funkcja zwraca kod błędu do konsoli. Funkcja wykorzystuje gotowe rozwiązanie w celu rozpoznawania mowy, zostało ono jednak odpowiednio dostosowane tak, aby działało dobrze w aplikacji.

Podsumowanie

Aplikacja pozwala w zautomatyzowany sposób tłumaczyć nagrania na inne języki, tworząc przy tym pewnego rodzaju lektora. Aktualny postęp i rozwój sztucznej inteligencji oraz narzędzi działających przy jej pomocy powodują, że takie rozwiązanie może zostać rozbudowane o dodatkowe funkcje takie jak czytanie własnym głosem. Istnieją już odpowiednie narzędzia pozwalające na dokonanie takich przekształceń jedno z nich pozwala na to w pełni za darmo oraz online <https://app.kits.ai/>. Ciągły rozwój pozwala również na coraz to lepsze rozpoznawanie mowy, tłumaczenie oraz zamianę tekstu na mowę. Podsumowując aplikacja zapewnia możliwości dla twórców, aby lepiej docierać do odbiorców treści nie zważając na bariery językowe.

Źródła

D. Jurafsky i J. H. Martin, „Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft Summary of Contents”.

<https://learn.microsoft.com/pl-pl/azure/ai-services/speech-service/speech-to-text>

<https://cloud.google.com/speech-to-text>