Projekt WIMU 25Z

Przeglądarka embeddingów audio

Design Proposal

Zespół nr 6
Rafał Kuśmierz
Jakub Sałek
Aleksander Gryzik

Prowadzący projekt
mgr inż. Tomasz Radzikowski

Planowana funkcjonalność aplikacji

Aplikacja będzie narzędziem do porównywania i wizualizacji embeddingów audio–tekstowych. Pozwoli użytkownikom wgrywać pliki dźwiękowe, wpisywać teksty oraz analizować jak modele embeddingowe CLAP i SLAP interpretują znaczenie wprowadzonych danych.

Główne funkcje aplikacji:

1. Analiza pojedynczej pary danych wprowadzonych przez użytkownika
   Użytkownik będzie mógł wybrać model CLAP lub SLAP, wgrać plik audio i wpisać tekst. Aplikacja następnie obliczy embeddingi dla obu modalności oraz wyświetli:
   - wartość podobienstwa (cosine similarity),
   - heatmapę różnic między embeddingami,
   - wizualizację punktów w przestrzeni ukrytej.
2. Porównanie pracy modeli CLAP i SLAP na plikach wprowadzonych przez użytkownika
   Dla tej samej lub nowej pary plików audio i tekstu użytkownik będzie mógł porównać działanie obu modeli. Aplikacja pokaże różnice w similarity score, embeddingach oraz rozmieszczeniu punktów na wspólnej wizualizacji.
3. Porównanie pracy modeli CLAP i SLAP dla odgórnie wgranego zestawu plików
   W ramach demonstracji różnic w działaniu modeli zostanie przygotowany zestaw przykładowych nagrań – czysto instrumentalnych, jak i o charakterze emocjonalnym. Aplikacja obliczy embeddingi tych dźwięków za pomocą modeli CLAP i SLAP, a następnie zaprezentuje ich rozmieszczenie na wykresach 2D. Pozwoli to wizualnie porównać w jaki sposób oba modele grupują dane.

4. Ranking podobieństwa
   Aplikacja umożliwi tworzenie rankingów podobieństwa między tekstami a dźwiękami:
   - Text → Audio: użytkownik wpisuje opis, a aplikacja sortuje wgrane nagrania według dopasowania.
   - Audio → Text: użytkownik wgrywa dźwięk, a aplikacja wskazuje najlepiej pasujące teksty.
5. Pseudo-captioning
   Aplikacja będzie umożliwiać dopasowanie nagrania dźwiękowego do istniejącej bazy opisów tekstowych. Do celów demonstracyjnych zostanie przygotowana wbudowana lista tagów i opisów (np. instrumenty, emocje, style). Po wgraniu pliku audio aplikacja obliczy jego embedding, porówna go z embeddingami tekstów z bazy i zwróci ranking najlepiej dopasowanych opisów. Pozwoli to zobaczyć, jak model interpretuje znaczenie dźwięku i jakie skojarzenia semantyczne tworzy między modalnościami.
6. Ranking z lokalnej bazy użytkownika
   Użytkownik będzie mógł utworzyć własną bazę embeddingów audio przechowywaną lokalnie w aplikacji. Po wpisaniu tekstu system przeliczy embedding dla podanego opisu i wyszuka w bazie najbardziej podobne pliki audio, zwracając posortowaną listę wyników wraz z możliwością odsłuchu.
   Funkcja umożliwi odnajdywanie w lokalnej bazie nagrań, które „znaczeniowo" lub brzmieniowo przypominają podany opis.

Planowany zakres eksperymentów

- Subiektywna analiza podobieństwa dla par audio-audio, tekst-tekst i audio-tekst. Sprawdzenie czy podobne gatunki, instrumenty itd. grupują się razem w przestrzeni ukrytej; analogicznie dla par tekstów. W przypadku multimodalności audio-tekst, sprawdzenie wrażliwość modeli na różne aspekty opisu audio, np. instrumenty, nastrój, gatunek.
- Analiza zużycia zasobów komputera w różnych scenariuszach, np. obliczania embeddingów, redukcji wymiarowości embeddingów, rosnącej liczby przechowywanych danych.
- Analiza wpływu parametrów przetwarzania audio na działanie modeli (np. długość audio, częstotliwość próbkowania, obecność i brak normalizacji embeddingów itp.)
- Porównanie modeli embeddingowych pod kątem aspektów wspomnianych w powyższych punktach.

- Porównanie działania modeli dla różnych typów sygnałów audio, np. muzyka instrumentalna, mowa, dźwięki spoza zakresu słyszalności człowieka, dźwięki natury, dźwięki syntetyczne itp.
- Analiza działania modeli pod kątem następujących zastosowań:
    - Propozycje podobnych utworów w odtwarzaczach
    - Generowanie mowy na podstawie nagrania

Stack technologiczny

- **Python**
- **Streamlit** – framework działania aplikacji (serwer + interfejs użytkownika)
- **CLAP** (HuggingFace), **SLAP –** modele embeddingowe
- **SQLite** – baza danych
- **scikit-learn** – redukcja wymiarów w wizualizacji przestrzeni ukrytych
- **librosa** – audio processing
- **Docker** – konteneryzacja aplikacji
- **Plotly** - wizualizacje

Bibliografia

Elizalde, B., Deshmukh, S., Al Ismail, M., & Wang, H. (2022). *CLAP: Learning audio concepts from natural language supervision*. Microsoft Research. Pobrano z CLAP: Learning Audio Concepts From Natural Language Supervision

Hugging Face Transformers. *CLAP model documentation*. CLAP Hugging Face Audio Models

Guinot, J., Riou, A., Quinton, E., & Fazekas, G. (2025). *SLAP: Siamese Language–Audio Pretraining without Negative Samples for Music Understanding*. W *Proceedings of the 26th International Society for Music Information Retrieval Conference (ISMIR 2025), Daejeon, South Korea*. Pobrano z SLAP: Siamese Language-Audio Pretraining Without Negative Samples for Music Understanding

Pliploop. (b.d.). *SLAP: Siamese Language–Audio Pretraining*. Repozytorium GitHub https://github.com/Pliploop/SLAP

Tovstogan, P., Serra, X., & Bogdanov, D. (2022). *Visualization of deep audio embeddings for music exploration and rediscovery*. W *Proceedings of the 19th Sound and Music Computing Conference (SMC 2022)* (ss. 108–115). Music Technology Group, Universitat Pompeu Fabra. Pobrano z https://repositori-api.upf.edu/api/core/bitstreams/9eed935a-aa7f-4cf6-a803-103389d66369/content

Harmonogram

Tydzień	Okres	Planowane prace
1	9.10 - 15.10 (design proposal deadline)	Przygotowanie design proposal - analiza tematu projektu - wstępny przegląd literatury - wybór technologii
2	16.10 - 22.10	Zapoznanie z przedstawioną w ramach design proposal literaturą, analiza, wnioski. Konfiguracja środowiska, przygotowanie repozytorium. Rozpoczęcie prac nad prototypem.
3	23.10 - 29.10 (zgłoszenie gotowości prototypu)	Przygotowanie prototypu - integracja pojedynczego modelu - umożliwienie wprowadzania tekstu do modelu - obliczanie similarity score dla wprowadzonego tekstu i z góry założonego pliku audio
4	30.10 - 05.11 (prototyp - deadline na spotkania)	Zrealizowanie spotkania przedstawiającego przygotowany prototyp.
5	06.11 - 12.11 (w tym okresie 4 dni wolne)	Przegląd aktualnego postępu prac, refaktoryzacja, przemyślenie dalszego kierunku prac.
6	13.11 - 19.11	Umożliwienie wprowadzania audio do modelu
7	20.11 - 26.11	Umożliwienie korzystania z drugiego modelu
8	27.11 - 03.12	Integracja z bazą danych – ogólny model danych
9	04.12 - 10.12	Przechowywanie plików audio oraz embeddingów
10	11.12 - 17.12	Próba wizualizacji przestrzeni ukrytej
11	18.12 - 24.12	Przeprowadzenie opisanych eksperymentów
12	25.12 - 31.12 (w tym okresie 4 dni wolne)	Duże zespołowe spotkanie podsumowujące rezultaty projektu. Ustalenie pozostałych do wykonania prac.
13	01.01 - 07.01	Skompletowanie projektu, optymalizacja, testy.
14	08.01 - 14.01	Nagranie demo przygotowanego rozwiązania.
15	15.01 - 19.01 (termin zwolnienia)	Skompletowanie dokumentacji, przekazanie projektu prowadzącemu.
16	20.01 - 26.01 (termin ostateczny)	Potencjalne przekazanie poprawek prowadzącemu.