Downloader – pobieranie danych

Wykorzystywana biblioteka

Do realizacji modułu wykorzystano bibliotekę yt_dlp, która umożliwia pobieranie materiałów wideo oraz audio z różnych platform, takich jak YouTube, TikTok czy Instagram. Do zmiany formatu na mp3 zostało wykorzystane narzędzie ffmpeg które wydłuża czas wykonywania się programu o 3 sekundy ale zmniejsza jego rozmiar z 47.6 MB na 5.10 MB. Ta biblioteka została wybrana z kilku kluczowych powodów:

- Aktywne wsparcie społeczności oraz częste aktualizacje, co zapewnia wysoką stabilność działania i kompatybilność z najnowszymi zmianami po stronie serwisów.
- Bogate możliwości konfiguracyjne, m.in. wybór formatu i jakości pobieranych plików.
- Wysoka wydajność, umożliwiająca szybkie pobieranie nawet dużych plików dzięki elastycznemu dostosowaniu parametrów.
- Obsługa wielu serwisów, nie tylko YouTube, ale również innych popularnych platform.

Inne technologie

Pytube:

- Ograniczona funkcjonalność – brak możliwości konfiguracji pobierania wyłącznie audio.
- Rzadsze aktualizacje, co może wpływać na jej niezawodność.
- Obsługuje wyłącznie platformę YouTube.
- W dniu 25 października 2025 roku, pytube zwraca informację Error 400: Bad Request która jest powiązana z brakiem kompatybilności wersji pytube z najnowszą wersją Youtube API

Analiza i porównanie dostępnych modeli Speech-to-Text

Modele poddane testom:

- OpenAI Whisper (przez API)
- Google Speech-to-Text
- AWS Transcribe
- Transformers (model OpenAI Whisper)

Kryteria oceniania

1. Dokładności

Test dokładności zostanie wykonany na 10 krótkich plikach dźwiękowych a kryteriom oceniania będzie średnia wskaźników błędu słów WER

$$WER = \frac{S + D + I}{N}$$

gdzie:

S = substitutions, ilość słów zamienionych na inne
D = deletions, ilość usuniętych słów
I = insertions, ilość dodanych słów
N – ilość słów w oryginalnym tekście

Model	OpenAI	Google	AWS	Transformers
WER	0.264008	0.335317	0.422007	1.992381

Tabela 1. Wyniki testu dokładności

2. Szybkość

Test szybkości polega na zmierzenie czasu w jakim każdy z modeli jest w stanie zwrócić plik z tekstem.

Testy były wykonywane na 5 minutowym pliku .wav

Model	OpenAI	Google	AWS	Transformers
Czas (s)	62.28	67.78	71.26	407.3

Tabela 2. Wyniki testu szybkości

3. Koszt

Wszystkie rozwiązania wykorzystujące przetwarzanie w chmurze bądź calle API nalicza koszt za korzystanie z usług i każdą transkrybowaną minutę dźwięku

Model	OpenAI	Google	AWS	Transformers
Koszt (za min)	$0.006	$0.024	$0.024	$0.00

Tabela 3. Koszty naliczane przez usługi

Mimo, że te usługi są płatne, zarówno Google jak i AWS zapewniają darmowe środki wystraczające dla naszych potrzeb.

4. Trudność implementacji

Ostatnim kryterium jest subiektywna ocena trudności implementacji każdego z modeli

- OpenAI – najłatwiejsza implementacja zarówno w kodzie jak i usługi
- Google STT – średnia trudność skonfigurowania usługi, oraz średnia trudność implementacji
- AWS – najtrudniejsza Do skonfigurowania usługa, oraz średnia trudność implementacji
- Transformers – łatwa implementacja w kodzie, dla prostej konfiguracji

Otrzymane wyniki:

- Model OpenAI Whisper (API) konsekwentnie osiągał najwyższą dokładność (najniższy współczynnik błędu WER) spośród wszystkich testowanych rozwiązań, wykazując najlepszą odporność na różne akcenty i jakość nagrań.
- Wszystkie modele wykazały się podobną szybkością, różnią się nieznacznie w porównaniu do długości oryginalnego pliku.
- Testy biblioteki transformers potwierdziły, że jej szybkość jest silnie uzależniona od dostępnych podzespołów oraz komplikuje wdrożenie.

Ostatecznie wybranym rozwiązaniem jest OpenAI Whisper

PORÓWNANIE TECHNOLOGII DLA ANALIZY SENTYMENTU

Technologia	Plusy	Wady	Najlepszy scenariusz użycia
Transformers (np. BERT, RoBERTa)	Najwyższa dokładność oraz jakość. Idealne do klasyfikacji sentymentu dla danego aspektu, dzięki możliwości fine-tuningu. Wsparcie dla języka polskiego. Elastyczność – możliwość adaptacji do specyficznej domeny.	Średni do wysokiego czas przetwarzania. Przy dłuższych tekstach i dużych modelach konieczna potrzeba użycia GPU. Przez limit tokenów, potrzeba dzielenia dłuższych tekstów na mniejsze części. Wymagają wiedzy z zakresu uczenia maszynowego do fine-tuningu.	Najlepszy do: systemów gdzie priorytetem jest najwyższa możliwa dokładność, a koszt i złożoność wdrożenia są akceptowalne.
SpaCy	Bardzo szybki czas przetwarzania. Łatwość użycia i gotowe wytrenowane modele. Przychodzi z wytrenowanymi modelami. Niskie wymagania sprzętowe. Doskonałe do ekstrakcji kandydatów.	Nie wykonuje klasyfikacji sentymentu samodzielnie (wymaga integracji z innymi narzędziami). Ograniczona dokładność reguł – podejście bardziej podatne na błędy przy złożonych konstrukcjach zdaniowych.	Najlepszy do: Szybkiej ekstrakcji kandydatów na aspekty w podejściu hybrydowym.
Metody leksykalne (np. NTLK VADER)	Ekstremalnie szybkie i lekkie Nie wymagają uczenia Przydatne dla ogólnej analizy sentymentu. Przydatne do ogólnej analizy sentymentu	Niska dokładność i precyzja Opierają się na regułach, co nie pozwala na wiązanie sentymentu z konkretną cechą.	Najlepszy do: Sytuacji gdzie potrzebna jest jedynie bardzo ogólna, przybliżona ocena sentymentu całego dokumentu.
LLMs (np. GPT-4/5 lub Llama-3)	Doskonałe rozumienie kontekstu i niuansów Bardzo wysoka dokładność bez potrzeby fine-tuningu, jedynie na podstawie dobrze sformułowanego polecenia (promptu).	Wyższe opóźnienia Wysokie koszta Lokalnie wymaga dużej ilości RAM/GPU. Niedeterministyczny – wyniki mogą się różnić pomiędzy wywołaniami. Mniejsze wyspecjalizowane modele po fine-tuningu często przewyższają dokładnością.	Najlepszy do: Szybkiego prototypowania bez danych treningowych oraz analizy bardzo złożonych tekstów.

PyABSA	Łatwość implementacji Wysoka dokładność Zbudowanie specjalnie dla ABSA (Analizy sentymentu na podstawie cech)	Mniejsza kontrola nad architekturą Zależność od ciężkich modeli – wykorzystuje duże modele Transformer, co wiąże się z ich wymaganiami sprzętowymi i czasem przetwarzania.	Najlepszy do: Szybkiego wdrożenia wysokiej jakości modelu ABSA. Idealny przy ograniczonym czasie lub doświadczeniu w MLOPs.

Test czasu przetwarzania tekstu na podstawie 11 minutowego filmu z YouTube (https://www.youtube.com/watch?v=rng_yUSwrgU) zawierającego 2043 wyrazów:

- PyABSA na CPU
  - 27.70s
- PyABSA na GPU
  - 8.08s
- SpaCy + BERT-Base-Multilingual na CPU
  - 3.41s
- SpaCy + BERT-Base-Multilingual na CUDA
  - 1.47s

Porównanie jakości PyABSA z SpaCy + BERT-Base-Multilingual (ocenia sentyment w skali 1-5):

L.p.	Model	Zdanie (Fragment)	Aspekt	Oczekiwany Sentyment	Otrzymany Sentyment	Wniosek
1	PyABSA	So, not only can we finally say goodbye to the $ 800 60 Hz iPhone, but it also now enables things like the always on display if you ' re into that.	display	Positive	Negative	Błąd
	SpaCy + BERT-Base-Multilingual				4 stars	Poprawnie

L.p.	Model	Zdanie (Fragment)	Aspekt	Oczekiwany Sentyment	Otrzymany Sentyment	Wniosek
2	PyABSA	This screen is also protected by the new ceramic shield 2.	display	Positive	Neutral	Błąd
	SpaCy + BERT-Base-Multilingual				5 stars	Poprawnie
3	PyABSA	It’s the new selfie camera.	camera	Neutral	Positive	Błąd
	SpaCy + BERT-Base-Multilingual				3 stars	Poprawnie
4	PyABSA	So, better display, better selfie camera, and then to round out the trio of obvious improvements that everyone will like on a new phone, better battery.	display camera battery	positive positive positive	positive positive positive	Poprawnie
	SpaCy + BERT-Base-Multilingual				3 stars 3 stars 3 stars	Błąd przy baterii

Wnioski:

PyABSA:

- Ma w miarę wysoką dokładność na poziomie 70-80%.
- Z prostymi zdaniami (nr 4) radzi sobie bardzo dobrze, jednak ma problem jak jest potrzebny większy kontekst (nr 2).

- Częściej popełnia błędy jak w zdaniu nr 1.
- Dużo dłuższy czas przetwarzania (nawet 4-5 krotnie).

SpaCy + Transformers (nlptown/bert-base-multilingual-uncased-sentiment):

- Bardzo wysoka dokładność na poziomie 80-90%.
- W większości zdań, nawet takich gdzie jest potrzebny większy kontekst (nr 1, 2) daje rezultat pozytywne.
- Jest możliwość polepszenia wyników za pomocą fine-tuningu oraz dodatkową konfiguracją, w tym większą listą z listą cech telefonów.
- Krótszy czas przetwarzania.

Najlepsze rozwiązanie: Podejście hybrydowe

1. Ekstrakcja kandydatów na aspekty: SpaCy
2. Klasyfikacja sentymentu: Transformers (nlptown/bert-base-multilingual-uncased-sentiment)

Łączą one szybkość SpaCy z dokładnością i rozumieniem kontekstu przez Transformers.