



POLSKO-JAPOŃSKA AKADEMIA  
TECHNIK KOMPUTEROWYCH

**Wydział Informatyki**

**Katedra Inżynierii Oprogramowania**

Inżynieria Oprogramowania i Baz Danych

**Jędrzej Kosakowski**

Nr albumu s34593

**Analiza sentymentu wypowiedzi z wykorzystaniem  
elementów sztucznej inteligencji**

Praca magisterska, promotor

dr inż. Mariusz Trzaska, prof. PJATK

Warszawa, wrzesień 2026

## **Streszczenie**

Praca dotyczy istotnego problemu, jakim jest zróżnicowanie wyników analiz sentymentu generowanych przez różne modele językowe. Obecnie na rynku istnieje wiele rozwiązań, które oferują różne poziomy rozumienia tekstu naturalnego, w zależności od ich architektury oraz danych treningowych. Postawiono pytanie badawcze: w jakim stopniu wyniki analizy sentymentu są zbieżne między modelami, a w jakich obszarach i z jakich powodów modele te różnią się między sobą. Odpowiedzią na to jest zaproponowany system wielomodelowej analizy sentymentu integrujący pięć modeli językowych w jednolitym procesie przetwarzania, uzupełnionym o oryginalny mechanizm wykrywania wielu perspektyw i ważonego obliczania wyniku końcowego. Architektura systemu opiera się na punktach rozszerzeń umożliwiających dodawanie kolejnych źródeł oraz modeli.

Przeprowadzone zostały badania empiryczne na 510 zagadnieniach z siedemnastu różnych kategorii tematycznych przy wykorzystaniu utrwalonych danych wejściowych, co przełożyło się na 2550 wyników. Wykazano wysoką zgodność modeli w zakresie stosowania oceny liczbowej oraz umiarkowaną w klasyfikacji wydźwięku.

## **Słowa kluczowe**

analiza sentymentu, aplikacja webowa, modele językowe, przetwarzanie tekstu, sztuczna inteligencja

## **Abstract**

The thesis addresses the significant problem of divergence in sentiment analysis results generated by different language models. The market currently offers numerous solutions providing varying levels of natural language understanding, depending on their architecture and training data. The following research question was posed: to what extent are sentiment analysis results consistent across models, and in what areas and for what reasons do these models differ from one another? The answer is the proposed multi-model sentiment analysis system, integrating five language models in a unified processing pipeline, supplemented by an original mechanism for detecting multiple perspectives and weighted final score calculation. The system architecture is based on extension points allowing further sources and models to be added.

Empirical research was conducted on 510 topics from seventeen different thematic categories using frozen input data, yielding 2550 results. High agreement between the models was demonstrated in terms of numerical scoring, and moderate agreement in sentiment classification.

# Spis treści

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>Streszczenie .....</b>                                           | <b>2</b>  |
| <b>Słowa kluczowe .....</b>                                         | <b>2</b>  |
| <b>Abstract .....</b>                                               | <b>3</b>  |
| <b>1. Wstęp .....</b>                                               | <b>7</b>  |
| 1.1 Motywacja .....                                                 | 7         |
| 1.2 Cel pracy .....                                                 | 8         |
| 1.3 Zakres pracy .....                                              | 9         |
| 1.4 Rezultat pracy .....                                            | 10        |
| <b>2. Istniejące rozwiązania analizy sentymentu .....</b>           | <b>11</b> |
| 2.1 Narzędzia komercyjne .....                                      | 11        |
| 2.1.1 Brandwatch .....                                              | 11        |
| 2.1.2 Sprout Social .....                                           | 12        |
| 2.1.3 Brand24 .....                                                 | 13        |
| 2.2 Narzędzia open source i biblioteki .....                        | 14        |
| 2.2.1 VADER (Valence Aware Dictionary and sEntiment Reasoner) ..... | 14        |
| 2.2.2 Hugging Face Transformers .....                               | 15        |
| 2.3 Porównanie z proponowanym rozwiązaniem .....                    | 16        |
| <b>3. Analiza i projekt rozwiązania .....</b>                       | <b>17</b> |
| 3.1 Założenia projektowe .....                                      | 17        |
| 3.1.1 Wymagania funkcjonalne .....                                  | 17        |
| 3.1.2 Wymagania нефункционалне .....                                | 18        |
| 3.2 Architektura koncepcyjna systemu .....                          | 18        |
| 3.2.1 Ogólny schemat działania .....                                | 18        |
| 3.2.2 Przepływ danych w systemie .....                              | 19        |
| 3.2.3 Punkty rozszerzeń architektury .....                          | 21        |
| 3.3 Funkcjonalności aplikacji .....                                 | 21        |
| 3.3.1 Moduł analizy sentymentu .....                                | 21        |
| 3.3.2 Moduł źródeł danych .....                                     | 22        |
| 3.3.3 Panele użytkownika i administratora .....                     | 23        |
| 3.4 Model analizy sentymentu .....                                  | 24        |
| 3.4.1 Struktura odpowiedzi modeli językowych .....                  | 24        |
| 3.4.2 System perspektyw i wag .....                                 | 24        |
| 3.4.3 Algorytm obliczania wyniku końcowego .....                    | 25        |
| 3.5 Założenia metodyczne badania porównawczego .....                | 26        |
| 3.6 Wybór technologii i uzasadnienie .....                          | 26        |
| 3.6.1 Technologie warstwy frontend .....                            | 26        |
| 3.6.2 Technologie warstwy backend .....                             | 27        |
| 3.6.3 Technologie bazodanowe .....                                  | 28        |
| 3.6.4 Uwierzytelnianie .....                                        | 28        |
| 3.6.5 Interfejsy modeli językowych .....                            | 28        |
| <b>4. Technologie wykorzystane w pracy .....</b>                    | <b>30</b> |
| 4.1 Technologie warstwy frontend .....                              | 30        |
| 4.1.1 React .....                                                   | 30        |
| 4.1.2 TypeScript .....                                              | 31        |
| 4.1.3 Tailwind CSS .....                                            | 31        |
| 4.1.4 Vite .....                                                    | 31        |
| 4.2 Technologie warstwy backend .....                               | 32        |
| 4.2.1 Node.js .....                                                 | 32        |
| 4.2.2 Express .....                                                 | 32        |
| 4.2.3 Biblioteki uzupełniające .....                                | 33        |
| 4.3 Baza danych .....                                               | 33        |
| 4.3.1 MongoDB .....                                                 | 33        |

|                                                             |           |
|-------------------------------------------------------------|-----------|
| 4.3.2 Mongoose .....                                        | 34        |
| 4.4 Modele językowe .....                                   | 34        |
| 4.4.1 OpenAI GPT.....                                       | 34        |
| 4.4.2 Anthropic Claude.....                                 | 35        |
| 4.4.3 Google Gemini .....                                   | 35        |
| 4.4.4 Perplexity.....                                       | 35        |
| 4.4.5 Grok.....                                             | 35        |
| 4.5 Źródła danych.....                                      | 35        |
| 4.5.1 Reddit API.....                                       | 36        |
| 4.5.2 YouTube Data API .....                                | 36        |
| 4.5.3 NewsAPI .....                                         | 36        |
| <b>5. Implementacja i projekt techniczny .....</b>          | <b>37</b> |
| 5.1 Organizacja kodu.....                                   | 37        |
| 5.1.1 Podział na warstwy i typy współdzielone.....          | 37        |
| 5.1.2 Izolacja warstwy domenowej .....                      | 40        |
| 5.2 Punkty rozszerzeń systemu.....                          | 41        |
| 5.2.1 Kontrakty dostawcy analizy i źródła danych.....       | 41        |
| 5.2.2 Rejestry i ich konsumenci.....                        | 42        |
| 5.2.3 Koszt rozszerzenia.....                               | 43        |
| 5.3 Warstwa domenowa analizy sentymentu.....                | 44        |
| 5.3.1 Konstrukcja instrukcji przekazywanej modelom .....    | 44        |
| 5.3.2 Odczyt odpowiedzi i sygnalizowanie braków .....       | 46        |
| 5.3.3 Wyznaczanie oceny i klasyfikacja wydźwięku.....       | 47        |
| 5.3.4 Deterministyczny dobór danych wejściowych .....       | 49        |
| 5.4 Przebieg analizy i utrwalanie wyników.....              | 50        |
| 5.4.1 Koordynacja wywołań i izolacja niepowodzeń .....      | 50        |
| 5.4.2 Utrwalanie wyników.....                               | 51        |
| 5.4.3 Statystyki liczone po stronie bazy danych.....        | 52        |
| 5.5 Bezpieczeństwo i odporność .....                        | 53        |
| 5.5.1 Granica żądania .....                                 | 53        |
| 5.5.2 Rozpoznawanie niedostępności usług zewnętrznych ..... | 55        |
| 5.6 Moduł badawczy .....                                    | 56        |
| 5.6.1 Rozdzielenie pozyskiwania i analizy .....             | 56        |
| 5.6.2 Przyrostowość przebiegu i postać wyników .....        | 57        |
| <b>6. Badanie porównawcze modeli językowych .....</b>       | <b>59</b> |
| 6.1 Metodyka badania .....                                  | 59        |
| 6.1.1 Schemat eksperymentu i warunki pomiaru.....           | 59        |
| 6.1.2 Zbiór zagadnień .....                                 | 61        |
| 6.1.3 Miary zgodności .....                                 | 62        |
| 6.2 Charakterystyka wyników poszczególnych modeli.....      | 63        |
| 6.3 Zgodność ocen między modelami .....                     | 65        |
| 6.3.1 Zgodność ocen liczbowych .....                        | 66        |
| 6.3.2 Zgodność klasyfikacji wydźwięku .....                 | 67        |
| 6.3.3 Przypadki największej rozbieżności .....              | 69        |
| 6.4 Wykrywanie perspektyw .....                             | 70        |
| 6.5 Ograniczenia badania .....                              | 72        |
| 6.6 Wnioski .....                                           | 72        |
| <b>7. Podsumowanie .....</b>                                | <b>74</b> |
| 7.1 Zalety rozwiązania .....                                | 74        |
| 7.2 Wady i ograniczenia.....                                | 75        |
| 7.3 Plany na przyszłość .....                               | 76        |
| <b>Bibliografia .....</b>                                   | <b>78</b> |
| <b>Spis ilustracji.....</b>                                 | <b>81</b> |
| <b>Spis tabel .....</b>                                     | <b>82</b> |

|                                                                |           |
|----------------------------------------------------------------|-----------|
| <b>Spis listingów kodu .....</b>                               | <b>83</b> |
| <b>Dodatek A. Zbiór zagadnień wykorzystany w badaniu .....</b> | <b>84</b> |

# 1. Wstęp

W dobie dynamicznego rozwoju Internetu oraz mediów społecznościowych, gdzie codziennie generowane są miliardy treści tekstowych oraz ekspresywnych wypowiedzi, coraz większego znaczenia nabiera analiza opinii zawartej w treści i związanych z nią emocji wyrażanych przez użytkowników w sieci. Zrozumienie sentymentu towarzyszącego dyskusjom na tematy generujące zainteresowanie wśród społeczeństwa, takie jak polityka, sport czy gospodarka, jest dziś wartościową informacją. Praca koncentruje się na analizie sentymentu zawartego w wypowiedziach użytkowników Internetu poprzez aplikację webową, która integruje wiele modeli językowych oraz źródeł danych.

Rozdział składa się z motywacji podjęcia tematu, celu pracy wzbogaconego o cele pomniejsze, zakresu oraz osiągniętego rezultatu. Kolejne części pracy przedstawiają przegląd istniejących już rozwiązań, założenia projektowe systemu, wykorzystane technologie, opis realizacji technicznej, a także wyniki przeprowadzonego badania.

## 1.1 Motywacja

Początek publicznego Internetu datuje się na lata 80. XX wieku [1]. Szybko stał się on przestrzenią do dzielenia się treściami pomiędzy użytkownikami. Według danych pochodzących z raportu Digital 2025 [2], corocznego opracowania o stanie digitalizacji na świecie, w 2025 roku ponad 5,56 miliarda ludzi aktywnie korzysta z Internetu, a 2 na 3 osoby korzystają z mediów społecznościowych. Średni czas spędzany na tych platformach przez użytkownika to około 141 minut dziennie, w wyniku czego przyswaja on treści dostępne tam przez ponad jedną dwunastą doby. W tym okresie użytkownicy wielu platform internetowych, takich jak Reddit czy Twitter, nieustannie wyrażają opinie na tematy takie jak polityka, sport czy gospodarka. Doprowadziło to do powstania terminu Web 2.0 [3], gdzie to aktywność użytkowników jest odpowiedzialna za jego rozwój. Już od lat korporacje zdają sobie sprawę, że informacje o sentymencie oraz analiza wypowiedzi stanowią dużą wartość marketingową. Rynek tego rodzaju usług osiągnął w 2024 roku poziom 4,68 miliarda USD, a jego wzrost prognozowany jest na 14,4% rocznie do 2034 roku według Polaris Market Research [4].

Pomimo szybkiego rozwoju technologicznego istnieją dzisiaj jednak pewne ograniczenia sprzętowe i systemowe, takie jak wyzwania związane z przetwarzaniem bardzo dużych zbiorów danych z wielu źródeł. Wymaga to dostępu do odpowiedniej infrastruktury, jednak nawet wtedy generuje duże koszty zarówno czasowe, jak i obliczeniowe. Prowadzi to do sytuacji, w której mniejsze firmy nie są w stanie pozwolić sobie na analizę tego typu danych. Obecnie dostępne narzędzia komercyjne wiążą się więc z bardzo drogimi subskrypcjami i ogromem funkcjonalności, która często nie interesuje docelowego klienta.

Dodatkowo przetwarzanie języka naturalnego jeszcze do niedawna stanowiło dla wielu systemów duże wyzwanie ze względu na sarkazm, ironię czy wieloznaczność wypowiedzi [5]. Należy również pamiętać o uwzględnieniu kontekstu kulturowego czy sytuacyjnego. Proste metody zawodzą podczas poprawnej klasyfikacji tak wielu zawiłych elementów, przez co, aby poprawnie dokonywać badań w tym segmencie, potrzebne są bardziej zaawansowane narzędzia bądź integracja wielu z nich.

Odrębnym ograniczeniem jest sprowadzanie wydźwięku całej dyskusji do pojedynczej wartości. Opinia publiczna wobec danego zagadnienia rzadko bywa jednolita, bardzo często wypowiadają się grupy o odmiennym stosunku do przedmiotu, a ich stanowiska wzajemnie się znoszą przy uśrednianiu. Grupy takie określane są w niniejszej pracy mianem perspektyw,

przez które rozumie się spójne punkty widzenia wyróżniające się w analizowanych treściach, posiadające własne nacechowanie emocjonalne.

Dopiero niedawny rozwój dużych modeli językowych (*Large Language Models*, LLM), będących przełomowymi narzędziami sztucznej inteligencji do przetwarzania i generowania tekstu na poziomie zbliżonym do ludzkiego [6], umożliwił użytkownikom komercyjnym powszechny dostęp do chmurowych i lokalnych rozwiązań, takich jak chatboty. Narzędzia te są dziś ogólnodostępne i umożliwiają podstawową analizę sentymentu wypowiedzi oraz integrację z innymi usługami, takimi jak platformy mediów społecznościowych, w modelu *on the go*. Istnieją jednak pewne ograniczenia, takie jak zestaw danych, na bazie których model był uczony, czy ograniczony dostęp modelu do Internetu.

Obserwacje te stały się inspiracją do stworzenia prostego w obsłudze narzędzia łączącego różne modele językowe i źródła danych, umożliwiając kompleksowe porównywanie i analizę sentymentu wypowiedzi.

## 1.2 Cel pracy

Głównym celem pracy magisterskiej było zbadanie możliwości analizy sentymentu wypowiedzi poprzez aplikację webową, w oparciu o automatyczne wykrywanie i klasyfikację sentymentu na podstawie słów kluczowych, z wykorzystaniem różnych źródeł danych oraz popularnych modeli językowych, takich jak:

- OpenAI GPT;
- Anthropic Claude;
- Google Gemini;
- Perplexity;
- Grok.

Dane wejściowe pozyskiwane są ze znanych platform internetowych:

- YouTube;
- Reddit;
- serwisy informacyjne (NewsAPI).

Drugim równorzędnym celem było zebranie i porównanie odpowiedzi różnych modeli na takie same dane wejściowe. Badane były takie właściwości jak podobieństwo uzyskiwanych ocen wraz ze wskazaniem obszarów, w których różnice te były systematyczne, a nie przypadkowe. Dzięki temu aplikacja stanowi narzędzie badawcze, a nie wyłącznie użytkowe.

Główny cel pracy został zrealizowany bazując na osiągnięciu pomniejszych celów takich jak:

- zaprojektowanie architektury systemu z podziałem na warstwę frontendową oraz backendową;
- integracja pięciu modeli językowych, umożliwiającą równoległe przeprowadzanie analizy sentymentu;
- implementacja mechanizmu pozyskiwania danych z trzech źródeł internetowych: platformy Reddit, serwisu YouTube oraz serwisów informacyjnych za pośrednictwem NewsAPI;
- zaprojektowanie architektury opartej na punktach rozszerzeń, umożliwiającej dołączenie kolejnego modelu językowego lub źródła danych bez ingerencji w pozostałe elementy systemu;

- opracowanie modelu analizy sentymentu z wykrywaniem wielu perspektyw i punktów widzenia wraz z przypisanymi wagami oraz algorytmem obliczania wyniku końcowego;
- implementacja systemu rejestracji i uwierzytelniania użytkowników stworzonego w oparciu o tokeny JWT;
- stworzenie panelu historii zapytań umożliwiającego przeglądanie i porównywanie wyników poprzednich analiz;
- stworzenie panelu administracyjnego z możliwością podglądu statystyk i aktywności użytkowników;
- opracowanie narzędzia badawczego generującego pulę wyników na zamrożonym korpusie danych wejściowych, co umożliwia porównanie modeli w identycznych warunkach;
- przeprowadzenie testów porównawczych pięciu modeli językowych w celu oceny różnic w klasyfikacji sentymentu tych samych treści.

### 1.3 Zakres pracy

W pracy zawarte są zagadnienia z zakresu przetwarzania i automatycznej analizy sentymentu z wykorzystaniem popularnych modeli językowych. Segment teoretyczny pracy zawiera:

- tematykę przetwarzania języka naturalnego;
- metody analizy sentymentu;
- architektury nowoczesnych aplikacji webowych.

Zakres techniczny pracy to projekt oraz pełna implementacja aplikacji webowej składającej się z warstwy frontendu zbudowanej w oparciu o React oraz TypeScript, warstwy backendu stworzonej przy wykorzystaniu Node.js oraz jego frameworka Express połączonych z dokumentową bazą danych MongoDB. Całość została dodatkowo zintegrowana z zewnętrznymi usługami API zarówno modeli LLM, jak i źródeł danych, z których pochodzą dane wejściowe dla przeprowadzania faktycznej analizy. Ponadto utworzone zostało narzędzie badawcze uruchamiane z poziomu wiersza poleceń, umożliwiające wygenerowanie danych analizowanych w rozdziale 6. Poza podstawową funkcjonalnością aplikacja udostępnia:

- rejestrację i logowanie dla odbiorców;
- zakładkę panelu użytkownika;
- zakładkę historii zapytań;
- panel administracyjny z wbudowaną analizą danych.

Zakres badawczy pracy polegał na przeprowadzeniu analizy wyników badania sentymentu, które zostały uzyskane w wyniku przeprowadzonych testów z wykorzystaniem narzędzia badawczego. Stworzono pulę 510 jednakowych zagadnień w 17 kategoriach, na bazie których uzyskano 2550 odpowiedzi z modeli, które następnie poddano porównaniu z dodatkowym uwzględnieniem klasyfikacji sentymentu, wartości przypisywanych perspektywom pochodzącym z różnych grup oraz poziomu pewności odpowiedzi zwracanej przez model.

Praca nie obejmuje takich zagadnień jak:

- wdrożenia produkcyjnego aplikacji;
- kwestii związanych ze skalowalnością infrastruktury;

- trenowania własnych modeli uczenia maszynowego;
- analizy materiałów audio oraz video.

## 1.4 Rezultat pracy

Rezultatem pracy jest model analizy sentymentu oparty na równoległym wykorzystywaniu wielu modeli językowych, które analizują ten sam materiał źródłowy. Dodatkowo umożliwiają one wyodrębnianie osobnych perspektyw wraz z przypisanymi im ocenami i wagami. Dzięki zastosowanym rozwiązaniom rozstrzygane są dwie kwestie nieobejmowane przez pozycje omówione w rozdziale 2. Pierwszą jest brak punktu odniesienia dla pojedynczej oceny, niepozwalający stwierdzić, czy jest ona miarodajna. Druga to problematyczność samej skali liczbowej, która przy wartościach bliskich zeru zaciera różnicę pomiędzy tekstem neutralnym a takim, gdzie skrajne opinie wzajemnie się równoważą. Wraz z modelem opracowano algorytm wyznaczania oceny ważonej oraz regułę klasyfikacji rozróżniającą oba te przypadki.

Drugim rezultatem jest sposób organizacji systemu oparty na punktach rozszerzeń, umożliwiający dołączanie kolejnych modeli językowych oraz źródeł poprzez utworzenie pojedynczego modułu i wpisu w rejestrze. Dzięki temu nie są konieczne ingerencje w warstwę koordynującą analizę, interfejs użytkownika ani moduł statystyk. Właściwość tę zweryfikowano doświadczalnie poprzez dołączenie i usunięcie testowego dostawcy, w sposób opisany w podrozdziale 5.2.3. Ma ona znaczenie wykraczające poza wygodę utrzymania, ponieważ pozwala powtórzyć badanie porównawcze po pojawieniu się nowych modeli, przy zachowaniu identycznych warunków pomiaru.

Trzecim rezultatem jest metodyka badania porównawczego wraz z jego wynikami. Rozdzielenie przebiegu na dwie fazy, czyli utrwalenie korpusu zbudowanego z treści pobranych ze źródeł oraz właściwą analizę przy ujednoliconej instrukcji i parametrach wywołania, pozwoliło oddzielić zmienność danych dostępnych w Internecie od różnic pomiędzy modelami. Badanie przeprowadzone na 510 zagadnieniach wykazało, że zbieżność ocen zależy od przyjętej miary: dla ocen liczbowych była ona wysoka, natomiast dla etykiet umiarkowana.

Realizowalność powyższych założeń została wykazana poprzez skonstruowaną aplikację webową, udostępniającą opisane mechanizmy użytkownikowi końcowemu. Po rejestracji i zalogowaniu może on z poziomu panelu wpisać szukaną frazę oraz wybrać analizowane źródła i modele. Uzyskane oceny prezentowane są w tym samym miejscu wraz z wykrytymi perspektywami oraz uzasadnieniem. Wyniki są utrwalane w bazie danych i dostępne w panelu historii oraz administracyjnym. Odrębnym elementem jest moduł badawczy uruchamiany z wiersza poleceń, wykorzystujący tę samą logikę analizy, co pozwoliło przeprowadzić badanie na tym samym systemie, który jest przedmiotem pracy.

Opracowane rozwiązanie może znaleźć zastosowanie w monitorowaniu wizerunku marek i firm oraz analizie opinii publicznej na temat wydarzeń sportowych i politycznych. Ograniczenie zbioru badawczego do zagadnień produktowych, opisane w podrozdziale 6.1.2, wynika z wymogów pomiaru różnic między modelami, nie zaś z ograniczeń samego narzędzia. Dodatkowo może być ono wykorzystywane w celu badania nastrojów konsumenckich oraz w celach akademickich jako platforma do porównywania zachowania różnych modeli językowych w zadaniach analizy sentymentu.

## 2. Istniejące rozwiązania analizy sentymentu

Pierwszym krokiem przed przystąpieniem do projektowania i implementacji systemu jest przeprowadzenie analizy istniejących rozwiązań związanych z tematyką pracy. Analiza sentymentu, rozumiana jako automatyczne rozpoznawanie opinii, ocen i postaw wyrażanych w tekście [5], doczekała się zarówno rozwiązań komercyjnych, jak i narzędzi rozwijanych przez społeczność. W tym przypadku analiza polegała na zbadaniu tego segmentu wśród zarówno produktów komercyjnych end-to-end, jak i narzędzi open source. Jej celem była identyfikacja dostępnych rozwiązań, ocena ich możliwości oraz wskazanie ograniczeń czy problemów, które uzasadniałyby zastosowanie nowego podejścia. Następnie zostały one ze sobą zestawione celem porównania możliwości oraz podejścia wszystkich wziętych pod uwagę metod, tak aby wskazać różnice oraz przewagi i wady.

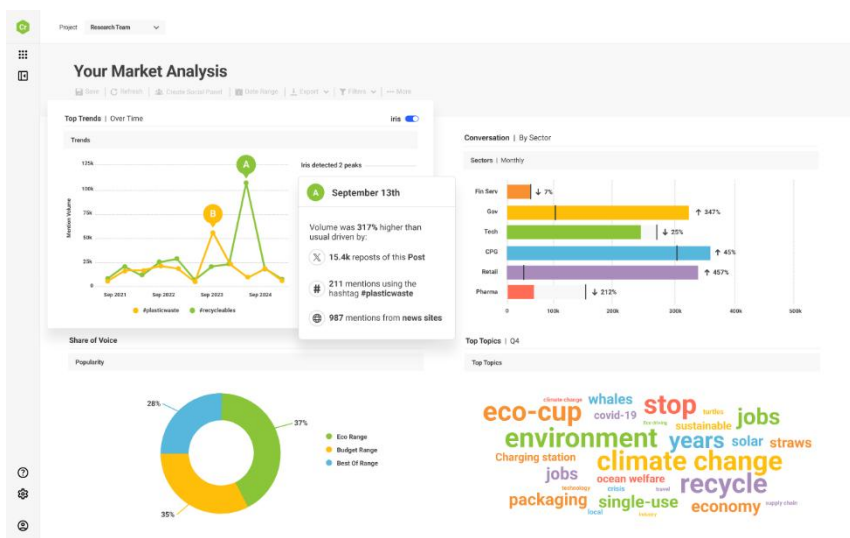
### 2.1 Narzędzia komercyjne

Podrozdział przedstawia analizę rozwiązań komercyjnych, które służą do analizy sentymentu wypowiedzi, a ich funkcjonalność zbliżona jest do stworzonej aplikacji.

#### 2.1.1 Brandwatch

Brandwatch to jedna z największych platform służących do *social listeningu* oraz monitorowania wzmianek o firmie w przestrzeni internetowej [7]. Jej flagową funkcjonalnością jest możliwość śledzenia wspomnień o produktach i markach w czasie rzeczywistym. Dodatkowo aplikacja umożliwia monitorowanie szerokiego wachlarza źródeł, takich jak:

- sieci społecznościowe;
- blogi;
- fora;
- serwisy informacyjne.



Rysunek 1. Pulpit aplikacji Brandwatch. Źródło:[7]

Platforma kładzie przy tym nacisk na globalny zasięg i wielojęzyczność. Obejmuje obszerne, ale skomplikowane pulpity informacyjne umożliwiające prezentację wielorakich danych i statystyk; przykładowy interfejs zaprezentowany jest na rysunku 1. Bogata oferta

aplikacji odnajduje swoje odzwierciedlenie w wysokiej cenie produktu, idącej w nawet setki tysięcy dolarów rocznie. Brandwatch oferuje integrację z wieloma znanymi mediami społecznościowymi i posiada bazę milionów pozycji opartych na analizie m.in. Google Reviews, Amazon i Trustpilot.

#### **Zalety:**

- rozbudowana baza danych historycznych;
- analiza w czasie rzeczywistym;
- wsparcie dla ponad 44 języków;
- integracja z systemami CRM;
- możliwości konfiguracji pulpitów.

#### **Wady:**

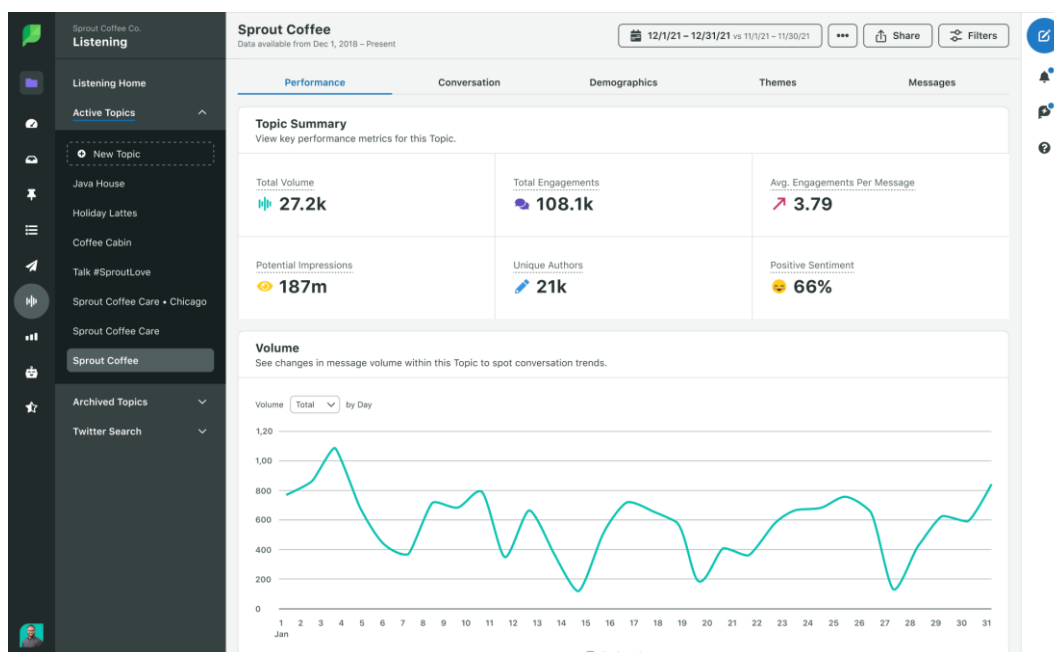
- wysoki koszt subskrypcji oparty na indywidualnej wycenie bez publicznie dostępnego cennika;
- znacząca krzywa uczenia się;
- platforma nie umożliwia porównania wyników z różnych modeli językowych jednocześnie, analiza oparta jest wyłącznie na własnym modelu Brandwatch.

#### **2.1.2 Sprout Social**

Sprout Social to platforma służąca do zarządzania mediami społecznościowymi, która została rozbudowana między innymi o narzędzia umożliwiające analizę sentymentu opartą na sztucznej inteligencji [8]. Produkt ten oferuje przede wszystkim możliwość monitorowania odbioru marki oraz jej prezencję w różnych aplikacjach, a także na forach. Dodatkowo oferta rozbudowana jest o analizę sentymentu odbiorców, a także możliwość dostosowywania swoich publikacji, tak aby trafiały one do jak najszerszego grona. Sprout Social wykorzystuje technikę uczenia maszynowego opartą na sieciach neuronowych, które obliczają wynik prawdopodobieństwa dla każdej wiadomości, klasyfikując ją jako:

- pozytywną;
- negatywną;
- neutralną.

Rysunek 2 prezentuje przykładowy pulpit informacyjny dostępny w aplikacji, na którym dostrzec można oferowane przez produkt funkcjonalności, takie jak wykrywanie pozytywnych i negatywnych trendów, tagowanie i klasyfikację wiadomości odbiorców przez AI Assistanta zgodnie z wykrytymi emocjami oraz sugerowanie kampanii i wykorzystania słów kluczowych.



Rysunek 2. Pulpit aplikacji Sprout Social. Źródło:[8]

### Zalety:

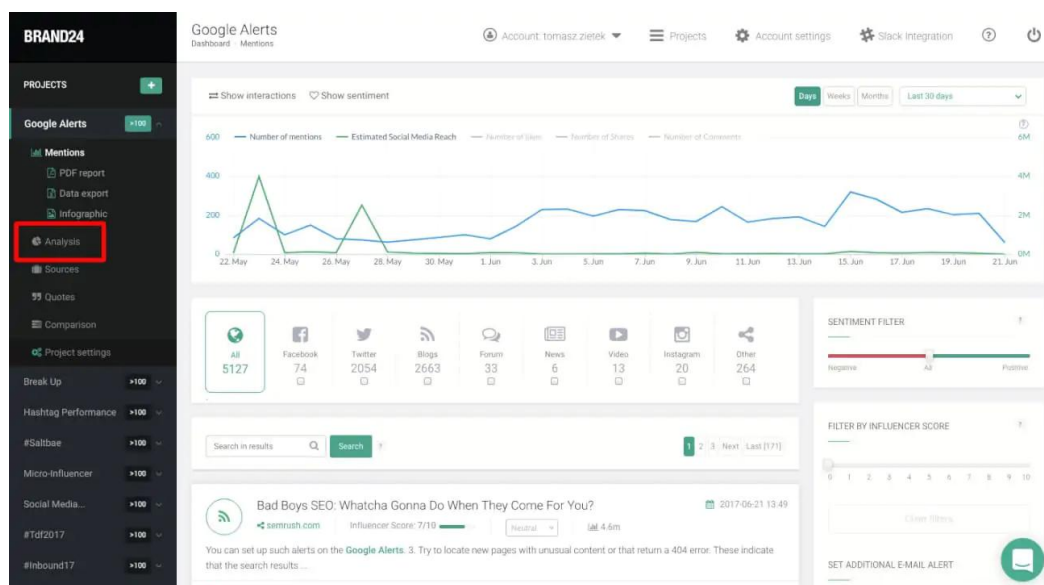
- połączenie analizy sentymentu z publikacją treści i obsługą klienta w jednym narzędziu;
- alerty w czasie rzeczywistym;
- możliwość ręcznego doskonalenia modelu.

### Wady:

- analiza sentymentu ograniczona do klasyfikacji pozytywny/negatywny/neutralny bez wykrywania perspektyw ani szczegółowych emocji;
- wiadomości zawierające wyłącznie materiały wideo lub graficzne bez tekstu są klasyfikowane jako nierozpoznane;
- model oparty wyłącznie na własnej sieci neuronowej Sprout, brak możliwości wyboru lub porównania różnych modeli językowych.

### 2.1.3 Brand24

Brand24 jest polską platformą do monitorowania mediów społecznościowych oraz tradycyjnych [9]. Posiada ona ponad 500 tysięcy klientów korzystających z jej funkcji śledzenia obecności marki w Internecie, co czyni ją popularnym rozwiązaniem na rodzimym rynku. Produkt oferuje możliwość analizy sentymentu, klasyfikując go na trzy rodzaje analogicznie do Sprout Social. Do realizacji tego wykorzystywane są modele językowe mające na celu rozpoznawanie tonu wypowiedzi na podstawie jej zawartości i tematu. Główną funkcjonalnością platformy pozostaje możliwość śledzenia wzmianek o marce bądź produkcie oraz weryfikacji trendów i efektywności kampanii reklamowych. Wyniki tych operacji widoczne są na łatwych do sparаметryzowania pulpitych informacyjnych, takich jak ten zaprezentowany na rysunku 3. Platforma zbiera wzmianki z ponad 20 różnych źródeł komunikacji, w tym z mediów społecznościowych. Istnieje również możliwość parametryzacji alertów informujących użytkowników o nagłych wydarzeniach związanych z ich marką w Internecie.



Rysunek 3. Pulpit aplikacji Brand24. Źródło: [9]

### Zalety:

- przystępna cena;
- analiza emocji wykraczająca poza podstawową klasyfikację sentymentu;
- szeroki zakres monitorowanych źródeł.

### Wady:

- wymaga ręcznych korekt, a samo dostrojenie słów kluczowych dla popularnych fraz bywa trudne ze względu na dużą liczbę nieistotnych wyników;
- brak możliwości wyboru lub porównania różnych modeli językowych, analiza oparta wyłącznie na własnym modelu platformy.

## 2.2 Narzędzia open source i biblioteki

Alternatywą dla rozwiązań komercyjnych opisanych w poprzednim podrozdziale stanowią narzędzia i biblioteki *open source*. Są to pozycje ogólnodostępne i darmowe, a także często rozwijane przez ich społeczność. Charakteryzują się one pełną dostępnością do kodu źródłowego, możliwością wdrożenia lokalnie oraz brakiem konieczności opłacania licencji.

### 2.2.1 VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER to rozbudowany algorytm analizy sentymentu oparty na leksykonie będącym słownikiem emocjonalnym oraz zestawie reguł językowych. Został on opracowany przez C.J. Hutto oraz E.E. Gilberta w 2014 roku w celu analizy krótkich, nieformalnych wypowiedzi pochodzących głównie z mediów społecznościowych [12]. Narzędzie to jest zaopatrzone we wcześniej wspomniany leksykon będący zbiorem około 7500 słów, skrótów czy emotikon. Każdy taki element posiada przypisaną wartość sentymentu od  $-4$  do  $+4$ , oceniającą intensywność towarzyszących mu emocji. Badania wykazały, że VADER osiąga wynik F1 równy 0,96 przy klasyfikacji tweetów, przewyższając tym samym indywidualnych oceniających, którzy osiągają wynik 0,84 [12]. Wynik ten dotyczy jednak wyłącznie krótkich wypowiedzi z mediów społecznościowych, przykładowo dla recenzji filmowych i tekstów prasowych spada on do poziomu 0,55–0,61. Narzędzie udostępniane jest na licencji MIT jako biblioteka języka Python wraz z dokumentacją opisującą sposób jego wykorzystania [10], [11].

### Zalety:

- całkowicie darmowe i *open source*;
- nie wymaga trenowania modelu ani danych uczących, bardzo wysoka szybkość przetwarzania umożliwiającą analizę w czasie rzeczywistym;
- możliwość wdrożenia w wielu projektach o różnej charakterystyce.

### Wady:

- trudności z rozpoznawaniem sarkazmu i ironii oraz złożonych negacji;
- ograniczony do słów zawartych w leksykonie, nowe wyrażenia slangowe i neologizmy nie są uwzględniane;
- leksykon zaprojektowany wyłącznie dla języka angielskiego, co skutkuje błędną interpretacją sentymentu w innych językach i kontekstach kulturowych;
- narzędzie wymaga znajomości technicznej i wdrożenia, nie jest to rozwiązanie *end-to-end*.

## 2.2.2 Hugging Face Transformers

Hugging Face Transformers jest biblioteką *open source* umożliwiającą wykorzystanie jednego z tysięcy nowoczesnych modeli językowych opartych na architekturze Transformer [13]. Dzięki jej zastosowaniu uzyskuje się dostęp do modeli uczenia maszynowego i *deep learningu*, takich jak BERT, RoBERTa czy GPT.

Jednym z kluczowych i szerzej wykorzystywanych modeli jest należący do Google BERT, uwzględniający przy analizie każdego słowa kontekst zarówno poprzedzający, jak i następujący po nim. RoBERTa jest rozwinięciem modelu BERT, stworzonym przez Facebook AI [14]. Została zbudowana na większej i bogatszej bazie danych treningowych, co czyni ją bardziej efektywną w rozumieniu tekstu i analizie zawartego w nim sentymentu. Modele oparte na RoBERTa przygotowane do analizy sentymentu osiągają dokładność przekraczającą 90%, zależnie od dziedziny i wykorzystanego zbioru danych [14].

### Zalety:

- bardzo wysoka dokładność klasyfikacji dzięki głębokiemu zrozumieniu kontekstu;
- ogromna biblioteka gotowych modeli dostrojonych do konkretnych dziedzin, takich jak finanse, medycyna czy social media;
- możliwość dalszego dostrajania modelu na własnych danych w celu zwiększenia dokładności dla konkretnego zastosowania.

### Wady:

- modele transformerowe przetwarzają maksymalnie 512 tokenów jednorazowo, co oznacza, że dłuższe teksty są przycinane i część informacji może zostać utracona;
- wysokie wymagania sprzętowe, efektywne działanie wymaga GPU, co znacząco utrudnia wdrożenie bez odpowiedniej infrastruktury;
- znaczna złożoność wdrożenia i konfiguracji wymagająca zaawansowanej wiedzy z zakresu uczenia maszynowego i NLP.

## 2.3 Porównanie z proponowanym rozwiązaniem

Analiza rozwiązań zarówno komercyjnych, jak i *open source* przyniosła ciekawe spostrzeżenia oraz wnioski. W tabeli 1 zamieszczone zostały wszystkie analizowane wcześniej pozycje wraz ze wskazaniem kluczowych zbieżności oraz różnic pomiędzy nimi. Dodatkowo zostały one zestawione z proponowanym rozwiązaniem. Pod uwagę brane są takie aspekty jak wielomodelowość, analizowane źródła danych, możliwość wykrywania perspektyw, koszty wynikające z korzystania z narzędzia oraz potrzeba wiedzy technicznej. Żadna z analizowanych pozycji poza proponowaną nie spełnia wszystkich oczekiwanych założeń.

Tabela 1. Zestawienie porównywanych rozwiązań

| Narzędzie                      | Wiele modeli          | Źródła danych                                            | Perspektywy | Interfejs  | Wiedza techniczna |
|--------------------------------|-----------------------|----------------------------------------------------------|-------------|------------|-------------------|
| Brandwatch                     | nie                   | media społecznościowe, serwisy informacyjne, blogi, fora | częściowo   | tak        | średnia           |
| Sprout Social                  | nie                   | media społecznościowe                                    | nie         | tak        | niska             |
| Brand24                        | nie                   | ponad 20 rodzajów źródeł                                 | nie         | tak        | niska             |
| VADER                          | nie                   | brak (dane własne)                                       | nie         | nie        | wysoka            |
| Hugging Face                   | częściowo             | brak (dane własne)                                       | nie         | nie        | wysoka            |
| <b>Proponowane rozwiązanie</b> | <b>tak (5 modeli)</b> | <b>Reddit, YouTube, serwisy informacyjne</b>             | <b>tak</b>  | <b>tak</b> | <b>niska</b>      |

Narzędzia komercyjne oferują rozbudowane interfejsy użytkownika i szerokie obsługiwanie źródeł danych. Z drugiej jednak strony ich analiza sentymentu opiera się wyłącznie na własnych, zamkniętych modelach z ograniczonymi możliwościami, często skupionymi tylko na ogólnym określeniu sentymentu w wypowiedzi. Użytkownik nie ma możliwości wyboru modelu językowego ani porównania wyników uzyskanych od różnych systemów AI z tego samego źródła danych. Dodatkowo narzędzia te wiążą się z wysokimi kosztami subskrypcji, co znacząco ogranicza ich dostępność dla indywidualnych użytkowników i mniejszych organizacji.

Rozwiązania *open source*, takie jak VADER i Hugging Face Transformers, są bezpłatne i bardziej elastyczne. Wymagają one jednak zaawansowanej wiedzy technicznej, bez której niemożliwe jest wdrożenie i korzystanie z narzędzi. Nie są to rozwiązania typu *end-to-end*, więc aby były one użyteczne, muszą stanowić część większego systemu. Ogranicza to ich użyteczność dla pojedynczych przedsiębiorstw.

Proponowane rozwiązanie wypełnia zaprezentowane luki poprzez połączenie różnych modeli językowych dostępnych równolegle, agregację danych z wielu niezależnych źródeł internetowych oraz wykrywanie różnych perspektyw z przypisanymi wagami. Ponadto posiada ono przejrzysty interfejs webowy, dostępny bez wiedzy programistycznej i bez dodatkowych kosztów.

### 3. Analiza i projekt rozwiązania

Na podstawie przeprowadzonej analizy rozwiązań komercyjnych oraz narzędzi *open source*, a także bibliotek, zidentyfikowano ich możliwości i ograniczenia. Pozwoliło to opracować:

- zestaw wymagań funkcjonalnych i нефункциональных systemu;
- architekturę koncepcyjną wzbogaconą o przepływ danych;
- szczegółowy opis funkcjonalności tworzonej aplikacji;
- informacje o stworzonym modelu analizy sentymentu oraz uzasadnienie wyboru poszczególnych technologii;
- koncepcję rozszerzalności systemu opartą na punktach rozszerzeń;
- założenia metodyczne badania porównawczego modeli.

#### 3.1 Założenia projektowe

Podstawowym etapem projektowania systemu jest zebranie i analiza wymagań, które opisują oczekiwane zachowanie aplikacji oraz jej właściwości. Wymagania dzieli się na funkcjonalne i нефункциональные. Stanowią one fundament, na którym opiera się projektowanie architektury systemu, modelu danych oraz interfejsu. Wymagania funkcjonalne określają, jakie funkcje i możliwości powinien udostępniać system, natomiast wymagania нефункциональные definiują sposób jego działania oraz cechy jakościowe, takie jak:

- wydajność;
- bezpieczeństwo;
- użyteczność;
- niezawodność.

##### 3.1.1 Wymagania funkcjonalne

System powinien spełniać następujące wymagania funkcjonalne:

- użytkownik powinien mieć możliwość rejestracji i logowania się do systemu z wykorzystaniem nazwy użytkownika i hasła;
- użytkownik powinien mieć możliwość wprowadzenia dowolnego słowa kluczowego lub frazy jako pozycji do analizy;
- użytkownik powinien mieć możliwość wyboru jednego lub wielu modeli językowych spośród pięciu dostępnych;
- użytkownik powinien mieć możliwość wyboru jednego lub wielu źródeł danych spośród trzech dostępnych: platformy Reddit, serwisu YouTube oraz serwisów informacyjnych za pośrednictwem NewsAPI;
- system powinien automatycznie pobierać dane ze wskazanych źródeł na podstawie wprowadzonego zapytania i przekazywać je do wybranych modeli językowych;
- system powinien prezentować wyniki analizy w postaci klasyfikacji sentymentu, oceny punktowej w skali od  $-10$  do  $+10$  wraz z oceną ważoną, poziomu pewności odpowiedzi, uzasadnienia oraz wykrytych perspektyw z przypisanymi wagami;
- system powinien umożliwiać równoległe uruchomienie analizy z wykorzystaniem wszystkich pięciu modeli jednocześnie w celu porównania wyników;

- użytkownik powinien mieć dostęp do historii wszystkich przeprowadzonych analiz wraz z możliwością ich przeglądania i porównywania;
- użytkownik powinien mieć możliwość zarządzania swoim profilem, w tym edycji danych osobowych;
- administrator powinien mieć dostęp do panelu analitycznego umożliwiającego podgląd aktywności i statystyk użytkowników systemu.

### 3.1.2 Wymagania niefunkcjonalne

System powinien spełniać następujące wymagania niefunkcjonalne:

- wydajność – zapytania do wielu modeli językowych powinny być wykonywane równolegle, co minimalizuje czas oczekiwania na wyniki;
- bezpieczeństwo — system powinien uwierzytelniać użytkowników za pomocą mechanizmu bezstanowego oraz przechowywać hasła wyłącznie w postaci skrótów kryptograficznych;
- użyteczność – interfejs użytkownika powinien być intuicyjny i dostępny bez wiedzy programistycznej, umożliwiając przeprowadzenie analizy w kilku prostych krokach;
- niezawodność – system powinien obsługiwać błędy związane z niedostępnością zewnętrznych źródeł danych lub modeli językowych, zachowując odporność na awarie tych usług zewnętrznych oraz możliwość prześledzenia przebiegu zapytań;
- rozdzielność warstw – architektura systemu powinna wyraźnie rozdzielać warstwę frontendową od backendowej, co umożliwia niezależny rozwój i utrzymanie każdej z nich.

## 3.2 Architektura koncepcyjna systemu

System aplikacji oparty został na klasycznej architekturze trójwarstwowej, wyróżniając warstwę:

- prezentacji;
- logiki biznesowej;
- danych.

Podejście takie zapewnia możliwość rozwoju poszczególnych warstw niezależnie od siebie, co ułatwia ich utrzymanie oraz wprowadzanie zmian. Umożliwia również niezależne skalowanie każdej z warstw stosownie do jej obciążenia.

### 3.2.1 Ogólny schemat działania

Warstwa prezentacji odpowiada za interakcję z użytkownikiem końcowym. Przyjmuje dane wejściowe, prezentuje wyniki analizy, obsługuje historię zapytań oraz umożliwia zarządzanie kontem. Komunikacja z warstwą logiki biznesowej odbywa się za pośrednictwem protokołu HTTP, przy czym warstwa ta udostępnia interfejs programistyczny w stylu REST, pośrednicząc między prezentacją a logiką.

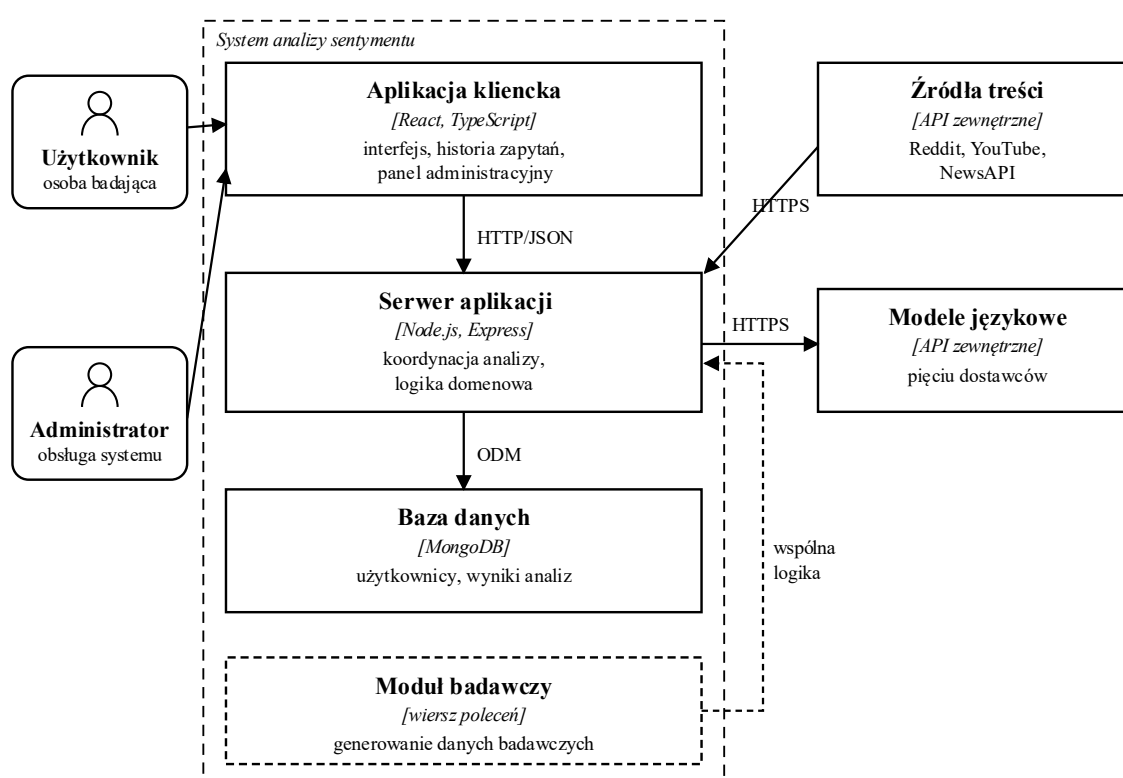
Warstwa logiki biznesowej stanowi centralny element systemu i realizuje jego założenia funkcjonalne, takie jak:

- uwierzytelnianie i rejestrowanie użytkowników;
- komunikacja z modelami językowymi;
- zapisywanie i odczytywanie danych zawartych w bazie.

Zastosowana architektura umożliwia prostą rozbudowę o nowe modele oraz źródła danych obsługiwane przez aplikację.

Warstwa danych została oparta na dokumentowej bazie danych o elastycznym schemacie, w której przechowywane są dane użytkowników oraz wyniki przeprowadzonych analiz. Wybór modelu nierelacyjnego podyktowany był zmienną strukturą wyników, wynikającą z różnej liczby perspektyw wykrywanych przy poszczególnych analizach. Dostęp do bazy realizowany jest przez warstwę odwzorowania obiektowo-dokumentowego, oddzielającą kod aplikacji od sposobu składowania danych.

System integruje się z pięcioma zewnętrznymi modelami językowymi oraz trzema źródłami danych, omówionymi szerzej w podrozdziałach 4.4 i 4.5. Połączenie realizowane jest przez udostępniane przez nie interfejsy programistyczne. Rysunek 4 przedstawia diagram architektury systemu z uwzględnieniem wszystkich warstw.



Rysunek 4. Diagram architektury systemu. Źródło: opracowanie własne

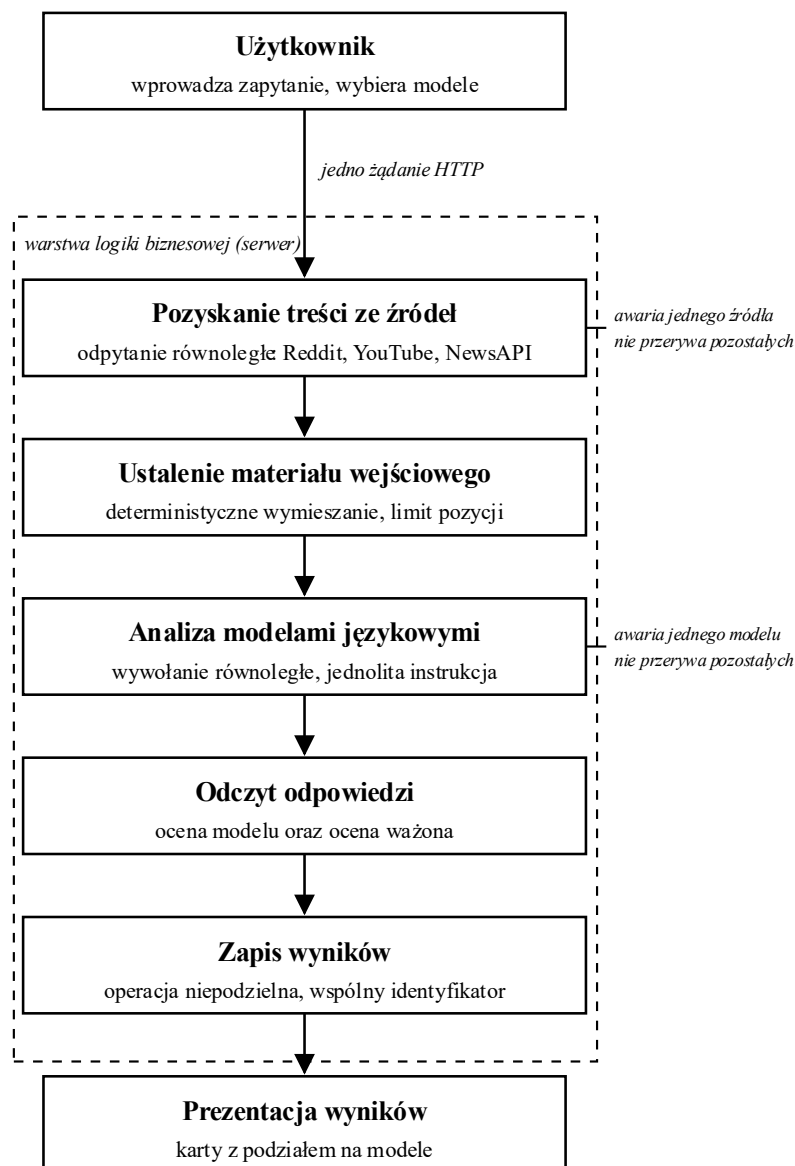
### 3.2.2 Przepływ danych w systemie

Przepływ w systemie związany z analizą danych przebiega zgodnie z zaprojektowanymi dla niego krokami zaprezentowanymi na rysunku 5. Przepływ przedstawiono w konwencji uproszczonej, ograniczonej do bloków oznaczających kolejne etapy przetwarzania oraz strzałek wskazujących kierunek, z wyróżnieniem granicy warstwy serwerowej. Rozpoczyna się on od wprowadzenia przez użytkownika zapytania, a kończy wyświetleniem wyników analizy na ekranie klienta.

Pierwszym krokiem jest wprowadzenie przez użytkownika słowa lub frazy klucz w interfejsie aplikacji w dedykowanym dla tego miejscu oraz wskazanie modeli językowych oraz

źródeł danych, z których mają zostać pobrane treści do analizy. Reszta procesu odbywa się po stronie serwera, gdzie odpytywane równoległe są wybrane wcześniej źródła danych. Awaria jednego z nich nie przerywa przetwarzania pozostałych, natomiast informacja o niej zostaje zapisana do odpowiedzi obok wyników. Odczytane treści zostają połączone, poddane deterministycznemu wymieszaniu i ograniczone do narzuconej wcześniej liczby pozycji, dzięki czemu powstaje gotowy materiał wejściowy do przeprowadzenia analizy.

Następnie dochodzi do złączenia wcześniej uzyskanych danych z instrukcją systemową, która zostaje przekazana równoległe modelom językowym. Podobnie jak w przypadku źródeł awaria pojedynczego modelu nie przerywa analizy pozostałych. Po wykonaniu analizy komplet wyników zapisywany jest w bazie jako jedna niepodzielna operacja, opatrzona wspólnym identyfikatorem zapytania. Częściowy zapis prowadziłby bowiem do powstania grupy wyników niepełnej, nieodróżnialne od analizy wykonanej mniejszą liczbą modeli, co zaburzyłoby późniejsze porównania.



Rysunek 5. Diagram przepływu w systemie. Źródło: opracowanie własne

Ostatnim krokiem jest przekazanie wyników do warstwy prezentacji, gdzie zostają wyświetlone użytkownikowi.

### **3.2.3 Punkty rozszerzeń architektury**

Podstawowym założeniem projektowym podczas tworzenia aplikacji było takie jej zorganizowanie, aby ewentualne rozszerzenia systemu o nowe modele językowe lub źródła danych były jak najmniej kosztowne. Aby to osiągnąć, zastosowano rozwiązanie opierające się na dwóch punktach rozszerzeń. W obu przypadkach metodyka podłączenia opisana jest w formie kontraktu, który określa, jak ma zostać ono zrealizowane. Pozostałe elementy specyficzne dla konkretnego dostawcy, takie jak sposób uwierzytelniania, adresy usług czy format odpowiedzi, zamknięte są wewnątrz jego modułu.

Dzięki temu wszelkie informacje na temat istniejących w systemie pozycji przechowywane są w jednym miejscu. Umożliwia to ograniczenie zakresu zmian, jakie są konieczne do dodania lub edycji nowego modelu czy źródła, co zweryfikowano doświadczalnie w sposób opisany w podrozdziale 5.2.3. Reszta systemu wykorzystuje tę jedną przestrzeń, gdzie są one zdefiniowane, zamiast je powielać.

Rozwiązanie to wykracza poza samo utrzymanie systemu i umożliwia powtórzenie badania dla modeli obecnie nieistniejących, które pojawiają się w przyszłości.

## **3.3 Funkcjonalności aplikacji**

Opracowanie odpowiedniej architektury systemu oraz organizacja przepływu danych umożliwiły przejście do kolejnego etapu projektowego, jakim jest pełne określenie funkcjonalności wszystkich elementów w projekcie. W podrozdziale zawarte są informacje o logicznym podziale zastosowanym w aplikacji wraz z szerszym opisem każdego z nich. Funkcjonalności zostały podzielone na cztery główne moduły: analizy sentymentu, źródeł danych odpowiedzialny za pozyskiwanie treści, panel użytkownika obejmujący zarządzanie kontem i historię zapytań oraz panel administracyjny służący monitorowaniu działania systemu.

### **3.3.1 Moduł analizy sentymentu**

Moduł analizy sentymentu stanowi główną funkcjonalność aplikacji i jest dostępny dopiero po rejestracji i zalogowaniu przez użytkownika. Proces ten zostaje rozpoczęty poprzez wprowadzenie dowolnego słowa kluczowego lub frazy w dedykowanym do tego polu wejściowym. Może to być nazwa drużyny sportowej, temat związany z polityką, nazwa produktu lub firmy, a także dowolne inne zagadnienie będące przedmiotem zainteresowania użytkownika.

Kolejnym krokiem jest wybranie modelu językowego, za pomocą którego przeprowadzona zostanie analiza. Dostępnych jest pięć modeli i można je uruchomić w dowolnej konfiguracji, od wyboru pojedynczego modelu po zastosowanie wszystkich jednocześnie. Analogicznie użytkownik wskazuje źródło bądź źródła danych, z których mają zostać pobrane treści do analizy, lub wybiera opcję agregacji wszystkich dostępnych jednocześnie. Wybór większej liczby pozycji wydłuża czas oczekiwania na wynik, przy czym wywołania wykonywane są równolegle, wobec czego czas ten odpowiada najwolniejszemu z nich, nie zaś ich sumie.

Zatwierdzenie zapytania przez użytkownika wyzwala w systemie automatyczne pobranie danych z wyselekcjonowanych wcześniej źródeł, przekazanie ich do wybranego bądź

wybranych modeli językowych, a następnie prezentację ustrukturyzowanego wyniku analizy. Każdy wynik składa się z następujących elementów:

- ogólnej klasyfikacji sentymentu w jednej z czterech kategorii: pozytywny, negatywny, neutralny lub mieszany;
- oceny punktowej w skali od -10 do +10 określającej intensywność i kierunek sentymentu;
- poziomu pewności odpowiedzi modelu wyrażonego w procentach;
- szczegółowej analizy tekstowej opisującej wydźwięk emocjonalny analizowanych treści;
- uzasadnienia klasyfikacji w postaci listy konkretnych argumentów wydobytych z analizowanych danych;
- wykrytych perspektyw, czyli różnych grup interesariuszy obecnych w analizowanych treściach wraz z przypisanymi im wagami, ocenami punktowymi i opisami;
- odnośników do źródeł stanowiących podstawę wystawionej oceny.

Jeżeli wykryte zostaną różne stanowiska, system wyznacza na ich podstawie dodatkową ocenę w postaci średniej ważonej ocen poszczególnych perspektyw, prezentowaną obok oceny wystawionej przez model. Umożliwia to zobrazowanie zróżnicowania opinii często występującego w wypowiedziach publikowanych w mediach społecznościowych. Dodatkowo ukazuje to, że w określonym temacie mogą występować odmienne punkty widzenia, dzięki czemu użytkownik nie zostanie zmylony pozornie umiarkowaną oceną ogólną.

### 3.3.2 Moduł źródeł danych

Moduł źródeł danych jest odpowiedzialny za pozyskiwanie treści internetowych stanowiących podstawę do wykonania analizy sentymentu danego tematu. Aplikacja została zintegrowana z trzema niezależnymi źródłami danych o odmiennej charakterystyce, które użytkownik może wybierać indywidualnie lub łączyć w dowolnej konfiguracji.

Pierwszym źródłem jest platforma Reddit, z której pobierane są posty i komentarze użytkowników odszukane w związku z wprowadzoną frazą. Dostęp do danych na platformie realizowany jest za pośrednictwem oficjalnego Reddit OAuth API, w którym uzyskiwanie poświadczeń działa na zasadzie mechanizmu OAuth, gdzie użytkownik, aby uzyskać dostęp, musi posiadać utworzoną aplikację wraz z *client id* oraz *client secret* [15]. Reddit charakteryzuje się brakiem formalności oraz większą swobodą wypowiedzi użytkowników [16]. Jest to wartościowe źródło w pozyskiwaniu danych i opinii, które nie są nadmiernie redagowane.

Drugie źródło stanowi serwis YouTube będący platformą do publikowania filmów wideo. W przypadku tej platformy źródłem danych są reakcje użytkowników pod filmami związanymi z wyszukiwaną frazą. Dane pozyskiwane są za pośrednictwem YouTube Data API [17] będącego częścią Google Cloud Platform. Stanowią one bezpośrednią reakcję odbiorców na prezentowane treści, często wyrażaną w sposób nacechowany emocjonalnie. Należy jednak pamiętać, że podlegają one moderacji zarówno ze strony platformy, jak i autorów filmów, wobec czego pozyskany materiał może nie obejmować pełnego zakresu wyrażanych opinii. Istotny jest również fakt, że dane pobierane z tego źródła to komentarze i opis filmu, a nie sama zawartość multimedialna.

Trzecim i zarazem ostatnim źródłem wykorzystywanym w aplikacji jest API serwisów informacyjnych występujące pod nazwą NewsAPI [18]. Zgromadzone są tutaj tysiące artykułów z różnych źródeł prasowych z całego świata. Rozwiązanie to umożliwia analizę sentymentu medialnego związanego z wyszukiwanym tematem. Należy zwrócić uwagę, że

źródło to jest bardzo zróżnicowane: z jednej strony występuje tu zawartość czysto informacyjna, z drugiej natomiast artykuły opiniotwórcze zawierające liczne subiektywne oceny, komentarze oraz interpretacje autorów. Umożliwia to uchwycenie wielu aspektów i obrazów współczesnego dziennikarstwa.

Po pobraniu danych z wybranych źródeł system łączy je, miesza i ogranicza do maksymalnie pięciu pozycji, które są następnie przekazywane do modeli. Proces ten zapobiega dominacji jednego ze źródeł nad pozostałymi. Całość ma charakter deterministyczny, dzieje się tak w wyniku wygenerowania liczby pseudolosowej dla konkretnego zapytania. Zapewnia to pozorną losowość przy zachowaniu powtarzalności, jako że to samo zapytanie skierowane do tego samego zestawu źródeł zawsze da dokładnie tę samą kolejność. Wartość sterująca doborem zostaje zapisana przy każdej analizie, dzięki czemu możliwe jest odtworzenie jej nawet w przyszłości.

### **3.3.3 Panele użytkownika i administratora**

W skład aplikacji poza jej podstawową funkcjonalnością wchodzi system obsługi kont użytkowników obejmujący rejestrację, logowanie oraz zarządzanie profilem. Do zarejestrowania się w systemie wymagane jest podanie nazwy użytkownika, imienia, nazwiska oraz hasła.

Hasła przechowywane są w bazie danych wyłącznie w postaci skrótów kryptograficznych, wyznaczanych funkcją zaprojektowaną do tego celu, wyposażoną w parametr kosztu utrudniający odgadywanie. Uwierzytelnianie użytkownika oparte jest na mechanizmie bezstanowych tokenów, które generowane są podczas logowania i dołączane do każdego kolejnego żądania wymagającego potwierdzenia tożsamości; bez nich użytkownik nie posiada dostępu do aplikacji.

Panel profilu umożliwia użytkownikowi przeglądanie i edycję danych osobowych, takich jak imię i nazwisko. Ma on również wgląd w podstawowe statystyki dotyczące swojej aktywności podczas korzystania z aplikacji, należą do nich takie pozycje jak liczba przeprowadzonych analiz czy najczęściej używane modele i źródła danych.

Kolejnym elementem jest moduł historii zapytań, w którym możliwe jest przeglądanie wszystkich analiz wykonanych przez obecnie zalogowanego użytkownika. Wyniki agregowane są zgodnie z zapytaniami, dzięki czemu przechowywane są względem operacji wykonanych przez użytkownika, a nie pojedynczych pozycji w bazie. Dzięki temu podczas prezentacji użytkownik widzi wyszukiwane przez siebie frazy wraz z przypisanymi im analizami z różnych źródeł i modeli. Każdy wpis historii zawiera datę przeprowadzenia analizy, użyte źródła danych oraz pełne wyniki wszystkich modeli, które zostały uruchomione dla danego zapytania.

Aplikacja posiada również panel administracyjny dostępny wyłącznie dla użytkowników z uprawnieniami administratora. Panel ten umożliwia podgląd listy zarejestrowanych użytkowników, statystyk ich aktywności oraz szczegółów przeprowadzonych analiz. Funkcjonalność ta pozwala na monitorowanie wykorzystania systemu oraz identyfikację ewentualnych nieprawidłowości w jego działaniu. Dodatkowo panel ten zawiera podstawową analizę danych użytkowników. Jest on szczególnie przydatny w przypadku potrzeby dokonania analizy działania modeli oraz ich ewentualnej kalibracji bądź dokonywania prac utrzymaniowych w aplikacji.

### 3.4 Model analizy sentymentu

Najważniejszym elementem projektowanego systemu jest model analizy sentymentu, który określa, w jaki sposób aplikacja przetwarza odpowiedzi modeli językowych. Podrozdział obejmuje ponadto informacje o przekształcaniu odpowiedzi w ustrukturyzowane wyniki, następnie prezentowane użytkownikowi. Omówiono zaprojektowany format odpowiedzi modeli językowych, mechanizm wykrywania i reprezentacji perspektyw oraz algorytm obliczania ważonego wyniku końcowego.

#### 3.4.1 Struktura odpowiedzi modeli językowych

W celu zapewnienia spójności i możliwości porównywania wyników uzyskiwanych od różnych modeli językowych zaprojektowany został jednolity format odpowiedzi, obowiązujący dla wszystkich pięciu modeli. Umożliwia to odczytanie odpowiedzi, dzięki czemu wyniki przekazywane są w odpowiednim formacie do bazy danych oraz do warstwy prezentacji. Format ten zdefiniowany jest przez instrukcję systemową załączaną do każdego modelu wraz z danymi wejściowymi. Instrukcja w precyzyjny sposób określa, jak przygotowana ma być zarówno struktura, jak i zawartość odpowiedzi. Szczegółowa implementacja instrukcji systemowej wraz z listingiem kodu omówiona została w podrozdziale 5.3.1.

W celu zaprojektowania formatu odpowiedzi musiały zostać przyjęte pewne założenia, których część opisano poniżej. Ocena sentymentu opiera się na tonie i emocjach zawartych w wypowiedzi, a nie na samym jej temacie. Punktem odniesienia dla oceny punktowej jest wartość zero, zarezerwowana dla tekstów wyłącznie faktycznych, pozbawionych zabarwienia emocjonalnego. W związku z tym każde rozpoznanie emocji, preferencji lub opinii skutkuje przypisaniem wartości co najmniej +1 lub -1, co ma na celu wykrycie jakiegokolwiek zabarwienia w tekście. Zastosowanie wartości dodatnich i ujemnych umożliwia jednoznaczne wskazanie kierunku sentymentu. Istotny jest tutaj również poziom pewności odpowiedzi, określający stopień, w jakim model jest przekonany o jej trafności. Jeżeli wartość ta wynosi powyżej 80 punktów, sentyment jest jednoznaczny, natomiast wartość między 40 a 60 punktów sugeruje niejednoznaczność wypowiedzi bądź znaczne zróżnicowanie emocjonalne. Model ma za zadanie aktywne poszukiwanie różnych perspektyw, a w przypadku, gdy w tekście wyróżniają się skrajne emocje, powinien wykryć towarzyszące im punkty widzenia. Ma to na celu ukazanie, że pojedyncza ocena ogólna nie zawsze w pełni oddaje charakter wypowiedzi i analizowanego tematu.

Każda odpowiedź modelu musi zawierać następujące pola w ściśle określonej kolejności:

- klasyfikacja wydźwięku w jednej z czterech kategorii: pozytywny, negatywny, neutralny lub mieszany;
- ocena punktowa w skali od -10 do +10, określająca intensywność i kierunek sentymentu;
- poziom pewności odpowiedzi modelu wyrażony jako wartość całkowita od 0 do 100;
- opis tekstowy wydźwięku emocjonalnego analizowanych treści;
- lista konkretnych argumentów uzasadniających przyznaną klasyfikację;
- wykaz wykrytych perspektyw, z których każda opisana jest nazwą, własną klasyfikacją, oceną punktową, wagą oraz krótkim opisem stanowiska.

#### 3.4.2 System perspektyw i wag

Moduł wykrywania i prezentacji perspektyw stanowi element wyróżniający projektowane rozwiązanie na tle omawianych wcześniej. Perspektywa w kontekście opisywanego systemu to spójny punkt widzenia dającej się wyodrębnić grupy osób

wyróżniającej się w analizowanych treściach, posiadający własne nacechowanie emocjonalne i dający się wskazać jako jedna ze stron dyskusji.

Mechanizm ten powstał w odpowiedzi na obserwację, że opinia publiczna na dany temat rzadko jest jednolita, przez co sumaryczna ocena analizy wyrażona jedną wartością liczbową bywa niemiarodajna i myląca. Wypowiedzi zawierające dwa wyraziste, przeciwstawne perspektywy dają w uśrednieniu wynik bliski zeru, a więc nieodróżnialny od materiału pozbawionego zabarwienia emocjonalnego. Instrukcja systemowa zawiera odrębny fragment, w którym model jest instruowany do aktywnego poszukiwania różnych stron dyskusji oraz rozbicia analizy na mniejsze, odnalezione grupy. Przykładowo w sporcie wykrywani są jednocześnie kibice drużyny zwycięskiej i przegranej, w polityce zarówno zwolennicy, jak i krytycy, a w przypadku produktów zarówno entuzjaści, jak i osoby krytyczne. Celem jest wykrycie od dwóch do czterech perspektyw dla tematów wielostronnych, dzięki czemu użytkownik ma wgląd w to, co realnie dzieje się w analizowanej konwersacji.

Waga każdej perspektywy obliczana jest jako iloczyn pewności jej wykrycia przez model oraz częstości występowania danego punktu widzenia w analizowanych danych.

### **3.4.3 Algorytm obliczania wyniku końcowego**

Poza oceną wystawioną bezpośrednio przez model system wyznacza dodatkowo ocenę ważoną, która powstaje poprzez zestawienie wykrytych perspektyw za pomocą średniej ważonej. Ma to na celu uwzględnienie stron, które biorą udział w dyskusji. Zastosowana skala ocen podzielona została na podgrupy odpowiadające intensywności sentymentu. Ocena 0 zarezerwowana jest tylko dla tekstów faktycznych pozbawionych jakiegokolwiek zabarwienia emocjonalnego. Instrukcja dodatkowo wyraźnie wskazuje, jakie nacechowanie powinno towarzyszyć każdej kategorii. Każde wykrycie emocji, preferencji lub opinii skutkuje przypisaniem wartości co najmniej +1 lub -1.

System został zaprojektowany tak, aby przechowywać zarówno ocenę ważoną złożoną z odnalezionych perspektyw, jak i ocenę wyliczoną przez model. Pierwsza z nich przedstawia wypadkową perspektyw poszczególnych grup, druga natomiast opisuje ogólny sentyment określony przez model. Rozdzielenie to ma również uzasadnienie metodyczne, ponieważ do porównywania odpowiedzi modeli można wykorzystać obie wartości, co rozszerza możliwości badawcze.

Na podstawie wyznaczonej oceny określana jest klasyfikacja sentymentu zgodnie z następującymi progami:

- wynik większy lub równy +2 przy braku perspektyw przeciwstawnych oznacza wydźwięk pozytywny;
- wynik mniejszy lub równy -2 przy braku perspektyw przeciwstawnych oznacza wydźwięk negatywny;
- wynik w przedziale od -2 do +2 przy obecności zarówno perspektyw pozytywnych, jak i negatywnych oznacza wydźwięk mieszany;
- wynik w przedziale od -2 do +2 przy braku perspektyw przeciwstawnych oznacza wydźwięk neutralny.

Przyjęte progi klasyfikacji zostały dobrane doświadczalnie na podstawie odpowiedzi zwracanych przez modele w trakcie projektowania systemu, tak aby zminimalizować liczbę błędnych klasyfikacji przy jednoczesnym zachowaniu wrażliwości na subtelne różnice w sentymencie analizowanych treści.

### 3.5 Założenia metodyczne badania porównawczego

Praca badawcza skupia się na weryfikacji stopnia zbieżności pomiędzy odpowiedziami generowanymi przez modele językowe od różnych dostawców przy zachowaniu tych samych danych wejściowych.

Jednym z wyzwań, z którym należało się zmierzyć, było ograniczenie ilościowe i czasowe dotyczące liczby zapytań kierowanych do API źródeł. W wyniku tego zastosowano rozwiązanie polegające na przyrostowym zbieraniu danych i zamrożeniu ich w bazie. Ma to na celu zapewnienie niezmienności danych dla różnych uruchomień modeli rozrzuconych w czasie. Następnie każdy z modeli analizował ten sam, niezmienny materiał.

Rozwiązanie takie niesie ze sobą dwie dodatkowe korzyści. Pierwszą jest ograniczenie kosztów uzyskania danych, ponoszonych jednokrotnie niezależnie od liczby porównywanych modeli. Drugą jest możliwość późniejszego wykorzystania zgromadzonego materiału do zbadania kolejnych, nowo dodanych modeli.

Dodatkowo stworzona została jednolita instrukcja systemowa oraz zbiór parametrów, tak aby każdy z modeli odpowiadał w możliwie identycznych warunkach. Zróżnicowanie tych ustawień prowadziłoby do powstania rozbieżności wynikających z odmiennej konfiguracji. Wraz z każdym wynikiem zapisywany jest ponadto identyfikator wersji modelu nadany przez dostawcę oraz skrót instrukcji systemowej. Dostawcy podmieniają bowiem wersje modeli udostępniane pod niezmienną nazwą, wobec czego bez tej informacji wyniki uzyskane w różnych okresach nie byłyby porównywalne.

Opisana rozdzielność na etapy wymaga, aby realizacja narzędzia odbyła się poza aplikacją webową. W tym celu stworzono skrypt wykorzystujący logikę przygotowaną na potrzeby aplikacji. Umożliwiło to nie powielanie tej samej funkcjonalności.

### 3.6 Wybór technologii i uzasadnienie

Odpowiedni dobór technologii stanowił istotny etap projektowania systemu, mający na celu wyselekcjonowanie rozwiązań najlepiej wpisujących się w obecne realia oraz oczekiwane rezultaty. W kolejnych podrozdziałach przedstawiono wybrane technologie wraz z uzasadnieniem ich zastosowania i omówieniem potencjalnych alternatyw. Technologie zostały wybrane na podstawie kryteriów takich jak:

- renoma technologii;
- dostępność dokumentacji;
- wsparcie społeczności;
- wydajność;
- spójność z pozostałymi elementami stosu technologicznego.

Miało to na celu zbudowanie odpowiedniej palety narzędzi, która umożliwi sprawny i efektywny proces twórczy oraz zapewni długoterminowe wsparcie i możliwość dalszego rozwoju aplikacji. Na podstawie charakterystyki projektu zdecydowano się na MERN stack, będący sprawdzonym i popularnym zestawem rozwiązań. Opisy wszystkich wymienionych dalej technologii wraz ze wskazaniem źródeł zamieszczono w rozdziale 4.

#### 3.6.1 Technologie warstwy frontend

Warstwa prezentacji wymagała doboru biblioteki do budowy interfejsu, języka programowania oraz narzędzi wspierających stylowanie i proces budowania aplikacji. Poniżej omówiono przyjęte rozwiązania wraz z uzasadnieniem oraz rozważanymi alternatywami.

Wybór biblioteki React jako technologii warstwy frontendowej podyktowany był jej dominującą pozycją na rynku, ogromną społecznością oraz bogatym ekosystemem gotowych bibliotek i komponentów, przydatnym przy interfejsie z wieloma widokami i dynamicznie aktualizowanymi wynikami analiz. Głównymi rozważanymi alternatywami były Angular i Svelte. Angular odrzucono ze względu na znacznie większą złożoność i stromą krzywą uczenia się, która nie byłaby uzasadniona dla projektu tej skali. Svelte, choć prostszy w nauce, posiada mniejszy ekosystem i słabszą popularność na rynku pracy, co przekłada się na ograniczoną dostępność zasobów i wsparcia.

Ekosystem React uzupełniono o dwie biblioteki realizujące funkcje niezbędne w projektowanej aplikacji. Obsługę nawigacji między widokami powierzono bibliotece React Router. Prezentacja zestawień statystycznych w panelu analitycznym zrealizowana została za pomocą biblioteki Recharts. Wybór obu podyktowany był ich ugruntowaną pozycją w ekosystemie React oraz brakiem konieczności wprowadzania dodatkowych zależności spoza niego.

Językiem implementacji uczyniono TypeScript. Zastosowanie tego rozwiązania umożliwiło wykrywanie błędów na etapie kompilacji oraz poprawiło czytelność kodu, a szczególnie typów danych wykorzystywanych w aplikacji. Jest to przydatne zwłaszcza w przypadku realizowania integracji z wieloma zewnętrznymi API, gdzie statyczne typowanie znacząco ułatwia pracę z różnorodnymi strukturami danych zwracanymi przez modele językowe i źródła danych. Alternatywą było wykorzystanie samego JavaScriptu, jednak ze względu na złożoność projektu i liczbę integracji zdecydowano się na TypeScript jako rozwiązanie bardziej niezawodne i czytelne.

Zastosowanie frameworka Tailwind CSS stanowi zdecydowane ułatwienie realizacji projektu aplikacji webowej oraz umożliwia ograniczenie liczby plików potrzebnych do utrzymywania takiego rozwiązania. Przyspiesza to prace nad wyglądem, niebędące istotnym elementem projektu. Alternatywą były Bootstrap oraz Material UI, jednak pierwszy z nich opiera się na gotowych komponentach o narzuconym wyglądzie, co ogranicza swobodę dostosowania. Material UI z drugiej strony narzuca gotową paletę rozwiązań, co wiąże się z mniejszymi możliwościami w dostosowywaniu stylów do własnych potrzeb.

Proces budowania powierzono narzędziu Vite. Wybór Vite zamiast Create React App podyktowany był jego wydajnością, która wynika ze znacząco krótszego czasu uruchamiania środowiska programistycznego. Skraca to czas iteracji podczas rozwoju aplikacji.

### **3.6.2 Technologie warstwy backend**

Warstwa serwerowa wymagała wyboru środowiska uruchomieniowego, frameworka obsługującego żądania oraz bibliotek realizujących zadania przekrojowe. Poniżej przedstawiono podjęte rozstrzygnięcia.

Wybór Node.js jako technologii backendowej był celowym zabiegiem mającym na celu unifikację wykorzystywanego języka programowania, jakim jest TypeScript, zarówno po stronie frontendowej, jak i backendowej. Dzięki temu projekt prowadzony jest w jednym języku, co upraszcza proces tworzenia i utrzymania aplikacji. Znaczenie miała również przydatność tego środowiska przy równoległych wywołaniach wielu usług zewnętrznych, stanowiących podstawę działania systemu. Alternatywą były inne frameworki do tworzenia serwerów backendowych, takie jak Spring Boot należący do ekosystemu języka Java oraz Django wywodzący się ze środowiska języka Python. Pomimo, że oba stanowią cenione i sprawdzone rozwiązania, ich zastosowanie wymagałoby wprowadzenia do projektu drugiego języka programowania, nie zapewniając w zamian znaczącej przewagi.

Wybór frameworka Express podyktowany był jego prostotą, elastycznością i nieodzowną rolą w ekosystemie Node.js jako efektywnego i szeroko wykorzystywanego rozwiązania. Alternatywą dla Express były Fastify i NestJS. Pomimo że Fastify projektowany był z myślą o wyższej przepustowości, różnice te ujawniają się dopiero przy znacznym obciążeniu, co nie stanowi przewagi w kontekście rozważanego projektu. NestJS oferuje bardziej ustrukturyzowaną architekturę wzorowaną na Angular, jednak jego złożoność oraz skomplikowana konfiguracja nie były uzasadnione dla projektu tej skali.

Warstwa serwerowa wzbogacona została o biblioteki realizujące zadania wspólne dla wszystkich obsługiwanych żądań. Sprawdzanie poprawności danych przychodzących powierzono bibliotece zod, ponieważ typowanie statyczne języka TypeScript obowiązuje wyłącznie na etapie kompilacji i nie chroni przed danymi niezgodnymi z deklaracją, nadchodzącymi z zewnątrz. Za ustawianie nagłówków bezpieczeństwa odpowiada biblioteka helmet, natomiast ograniczanie liczby żądań kierowanych do punktów końcowych uwierzytelniania realizuje express-rate-limit, chroniąc system przed próbami odgadywania haseł. Kontrolę pochodzenia żądań zapewnia cors. Komunikacja z usługami zewnętrznymi odbywa się za pośrednictwem biblioteki axios, wykorzystywanej również po stronie warstwy prezentacji, co ujednoliciło sposób obsługi żądań w całym systemie. Wartości konfiguracyjne, w tym klucze dostępowe do usług zewnętrznych, wczytywane są ze zmiennych środowiskowych z wykorzystaniem biblioteki dotenv, dzięki czemu nie są zapisywane w kodzie źródłowym.

### 3.6.3 Technologie bazodanowe

Charakterystyka przechowywanych danych zdecydowała o wyborze modelu bazy oraz sposobu dostępu do niej z poziomu kodu aplikacji.

Wybór bazy MongoDB był podyktowany jej elastycznością w kontekście schematu danych. Każda analiza może zawierać różną liczbę perspektyw i różne pola w zależności od wyniku zwracanego z modelu językowego, co sprawia, że jej struktura za każdym razem może ulec zmianie. Relacyjna baza danych wymagałaby z góry predefiniowanego schematu, co znacząco utrudniłoby obsługę zmiennej struktury wyników. Dodatkowo jako warstwę ODM zastosowano Mongoose umożliwiając modelowanie danych, walidację schematów i wygodny interfejs do komunikacji z bazą. Rozważaną alternatywą były relacyjne bazy danych, jak na przykład PostgreSQL. Ze względu na chęć zachowania prostoty rozwiązania oraz prostą konfigurację zdecydowano się na pozostanie przy nierelacyjnej bazie danych.

### 3.6.4 Uwierzytelnianie

Zabezpieczenie kont użytkowników obejmuje dwa odrębne zagadnienia: sposób przechowywania haseł oraz mechanizm potwierdzania tożsamości przy kolejnych żądaniach.

System uwierzytelniania oparty został na tokenach JWT oraz bibliotece bcrypt służącej do wyznaczania skrótów haseł. JWT wybrano ze względu na jego bezstanowość, co oznacza, że serwer nie przechowuje informacji o sesjach użytkowników. Prowadzi to do uproszczenia stosowanej architektury i ułatwia potencjalne skalowanie systemu. Bibliotekę bcrypt zastosowano ze względu na jej odporność na ataki słownikowe oraz metodą pełnego przeglądu (*brute force*). Alternatywą było zastosowanie gotowych rozwiązań uwierzytelniania, takich jak Firebase Authentication, jednak zdecydowano się na własną implementację w celu zachowania pełnej kontroli nad procesem uwierzytelniania i uniknięcia zależności od zewnętrznych usług.

### 3.6.5 Interfejsy modeli językowych

Jedną z kluczowych decyzji było zastosowanie zewnętrznych API modeli językowych zamiast trenowania własnego modelu uczenia maszynowego. Jest to uzasadnione różnymi czynnikami. Pierwszym z nich jest fakt, że trenowanie własnego modelu to proces długotrwały

i wymagający mocy obliczeniowej oraz dużego zestawu danych uczących, co wykracza poza zakres pracy. Dostępne modele językowe, takie jak GPT, Claude czy Gemini oferują znacznie wyższy poziom rozumienia kontekstu i niuansów językowych niż modele trenowane od podstaw na potrzeby jednego projektu, ze względu na skalę danych i mocy obliczeniowej wykorzystanych przy ich trenowaniu. Były one trenowane na bardzo dużych zestawach danych, a wyniki uzyskane z samodzielnie trenowanego modelu nie mogłyby dorównać tym, które uzyskują modele komercyjne. Dostawcy modeli różnią się sposobem udostępniania swoich usług. Część z nich, jak OpenAI czy Google, oferuje dedykowane biblioteki klienckie upraszczające integrację, podczas gdy pozostali wymagają bezpośredniego kierowania żądań protokołem HTTP.

Rozbieżności te stanowiły motywację do zaprojektowania wspólnego kontraktu opisanego w podrozdziale 3.2.3. Jego zadaniem jest ukrycie ich przed pozostałymi elementami systemu. Zastosowanie pięciu odrębnych interfejsów umożliwia ponadto porównanie wyników i ocenę różnic między modelami, co stanowi dodatkową wartość badawczą pracy.

## 4. Technologie wykorzystane w pracy

Dobór technologii wykorzystanych do realizacji aplikacji stanowi jeden z kluczowych etapów wytwarzania oprogramowania. Rozdział zawiera przegląd technologii zastosowanych w projektowaniu i implementacji systemu, przedstawionych w oderwaniu od samego projektu. Uzasadnienie ich wyboru zostało omówione w podrozdziale 3.6. Opis każdej z nich obejmuje ogólną charakterystykę, sposób działania oraz najważniejsze cechy decydujące o jej zastosowaniach. Technologie zostały pogrupowane według warstwy systemu, której dotyczą, począwszy od technologii frontendowych, przez backendowe i bazodanowe, po zewnętrzne modele językowe i źródła danych.

### 4.1 Technologie warstwy frontend

Warstwa frontendowa aplikacji została zrealizowana w oparciu o zestaw nowoczesnych technologii webowych. Mają one na celu zapewnienie wydajnego renderowania interfejsu użytkownika, statycznego typowania kodu oraz elastycznego stylowania komponentów. Podrozdział omawia kolejno bibliotekę React, język TypeScript, framework CSS Tailwind oraz narzędzie budujące Vite.

#### 4.1.1 React

React to biblioteka JavaScript stworzona przez Facebooka w 2013 roku i rozwijana do dziś [20]. Jest to jedno z najpopularniejszych rozwiązań, oferujących tysiące zrealizowanych projektów oraz duże wsparcie społeczności. Umożliwia ono realizację interfejsów webowych na podstawie koncepcji komponentów będących niezależnymi od siebie elementami wielokrotnego użytku. Za pomocą stworzonych komponentów można następnie budować zróżnicowane i skomplikowane interfejsy webowe.

Jedną z kluczowych koncepcji zastosowanych w bibliotece React jest wirtualny DOM, będący reprezentacją rzeczywistego drzewa DOM przechowywaną w pamięci. Umożliwia to wydajne renderowanie aplikacji webowej w przypadku zachodzących w niej zmian. Jest to możliwe dzięki mechanizmowi, jakim jest odszukanie zmian w nowym drzewie i aktualizacja wyłącznie tych elementów, które uległy modyfikacji. Rozwiązanie to jest wydajniejsze od manipulowania całym drzewem DOM.

Dodatkowo React cechuje się jednokierunkowym przepływem danych w aplikacji, gdzie dane przekazywane są od komponentu nadrzędnego do podrzędnego. Dzieje się to za pomocą właściwości (*props*), czyli danych przekazywanych komponentowi podrzêdnemu przez nadrzêdny, których komponent otrzymujący nie modyfikuje. Do zarządzania stanem wewnętrznym komponentu wykorzystywane są natomiast mechanizmy zwane hookami.

React na przestrzeni swojego istnienia został znacząco wzbogacony o dodatkowe biblioteki, takie jak React Query czy React Router. Umożliwiają one wykonywanie szeregu operacji, takich jak komunikacja z serwerem czy budowanie aplikacji wielostronicowych dzięki zastosowaniu routingu. Dzięki takim rozwiązaniom stał się on dzisiaj jednym z najczęściej wybieranych rozwiązań do budowania interfejsów aplikacji webowych, gdzie według Statista [21] z rozwiązania tego korzysta 44,7% programistów.

Ekosystem React zbudowany jest w oparciu o zbiory bibliotek rozszerzające jego podstawową funkcjonalność. Jedną z nich jest React Router umożliwiający obsługę nawigacji dla aplikacji jednostronicowych odwzorowując poszczególne adresy na komponenty widoków bez przeładowywania strony [22]. Kolejną wykorzystaną biblioteką jest Recharts, służąca do

tworzenia wykresów, zbudowana na komponentach React oraz standardzie SVG, co pozwala osadzać zestawienia statystyczne bezpośrednio w drzewie komponentów [23].

#### 4.1.2 TypeScript

TypeScript to język programowania stworzony przez firmę Microsoft, będący nadzbiorem popularnego języka JavaScript [24]. Jest to jednoznaczne z faktem, że każdy poprawny kod JavaScript jest jednocześnie poprawnym kodem TypeScript, natomiast TypeScript rozszerza JavaScript o dodatkowe funkcjonalności, takie jak system statycznego typowania umożliwiające zdefiniowanie typów zmiennych.

Typy deklarowane mogą być dla zmiennych, parametrów funkcji oraz wartości zwracanych. Umożliwia to kompilatorowi wykrywanie błędów już na etapie pisania kodu, zanim aplikacja zostanie uruchomiona. Dzięki temu błędy, takie jak przekazanie wartości niewłaściwego typu, wywołanie nieistniejącej metody czy odwołanie do niezdefiniowanej właściwości obiektu, są natychmiast sygnalizowane przez kompilator oraz edytor kodu.

Dzięki kompilowaniu do JavaScript kod napisany w TypeScript może być uruchamiany w każdym środowisku obsługującym język JavaScript. Umożliwia to wykorzystanie go zarówno w środowisku uruchomieniowym Node.js, jak i w przeglądarce. Zastosowanie tego narzędzia daje programiście większe możliwości w zakresie zarządzania danymi w realizowanej aplikacji.

#### 4.1.3 Tailwind CSS

Tailwind CSS to framework CSS oparty na koncepcji klas użytkowych, stworzony przez Adama Wathana [25]. W przeciwieństwie do tradycyjnych frameworków CSS, takich jak Bootstrap, Tailwind nie korzysta z predefiniowanych komponentów wizualnych. Oferuje za to zbiór wcześniej stworzonych klas wielokrotnego użytku, które umożliwiają efektywne stylowanie aplikacji webowych.

Jest to tożsame z faktem, że w przeciwieństwie do czystego CSS lub innych frameworków stylowanie odbywa się bezpośrednio w kodzie HTML czy JSX/TSX. Nie ma konieczności tworzenia dedykowanych plików arkuszy stylów do ich przechowywania. Znacząco usprawnia to proces wytwarzania aplikacji i pozwala na trzymanie całego kodu w jednym miejscu. Ponadto framework ten wyposażony jest w mechanizm usuwania nieużywanych klas w procesie budowania aplikacji produkcyjnej, co sprawia, że końcowy plik CSS zawiera wyłącznie te klasy, które zostały użyte w projekcie.

Kolejną zaletą frameworka Tailwind jest wbudowana obsługa trybu ciemnego, stanów interakcji oraz responsywności za pomocą mechanizmu prefiksów, takich jak *md:* czy *lg:*, co pozwala na łatwe dostosowywanie stylów do różnych rozmiarów ekranu i stanów elementu bez konieczności definiowania osobnych reguł.

#### 4.1.4 Vite

Vite to narzędzie służące do budowania aplikacji frontendowych, którego nazwa pochodzi od francuskiego słowa oznaczającego szybki [26]. Jest to nawiązanie do jednej z najważniejszych cech tego rozwiązania, jaką jest relatywnie szybki start serwera programistycznego oraz przebudowa modułów w porównaniu do tradycyjnych narzędzi budujących (bundlerów), takich jak Webpack czy Create React App.

Narzędzie opiera swoje działanie na dwóch kluczowych mechanizmach. W trybie programistycznym wykorzystywane są natywne moduły ECMAScript obsługiwane bezpośrednio przez przeglądarkę, co eliminuje konieczność scalania całej aplikacji przed uruchomieniem, dzięki czemu serwer uruchamia się niemal natychmiast. Dodatkową zaletą jest

wykorzystanie mechanizmu *hot module replacement*, umożliwiającego aktualizację zmienionych modułów w przeglądarce bez konieczności przeładowania całej strony, co sprawia, że zachowany zostaje stan aplikacji. Jest to szczególnie przydatne w przypadku wykorzystywania biblioteki React.

Vite obsługuje TypeScript, JSX, CSS i inne formaty plików bez konieczności dodatkowej konfiguracji, a jego architektura wtyczek oparta na narzędziu Rollup umożliwia łatwe rozszerzanie funkcjonalności.

## 4.2 Technologie warstwy backend

Warstwa backendowa aplikacji pełni rolę serwera i odpowiada za obsługę:

- żądań przychodzących z frontendu;
- komunikacji z zewnętrznymi API modeli językowych i źródeł danych;
- uwierzytelniania użytkowników;
- zapisu i odczytu danych z bazy.

Została ona zrealizowana przy wykorzystaniu języka TypeScript oraz środowiska uruchomieniowego Node.js wraz z frameworkiem Express, uzupełnionych o biblioteki realizujące zadania przekrojowe, takie jak sprawdzanie poprawności danych i zabezpieczenia warstwy sieciowej.

### 4.2.1 Node.js

Node.js to środowisko uruchomieniowe JavaScript zbudowane na silniku V8 przeglądarki Chrome [27]. Jego głównym zadaniem jest umożliwienie wykonywania kodu JavaScript poza przeglądarką, po stronie serwera, co pozwala wykorzystać go nie tylko do budowy interfejsu aplikacji, ale również rozwiązań backendowych.

Jedną z kluczowych cech Node.js jest asynchroniczny model wejścia i wyjścia oparty na pętli zdarzeń. Pozwala to na obsługę wielu żądań jednocześnie bez blokowania wykonania, mimo że kod aplikacji wykonywany jest w pojedynczym wątku. Operacje wejścia i wyjścia realizowane są bowiem poza nim, a ich wyniki trafiają do pętli zdarzeń po zakończeniu. Podejście to sprawia, że Node.js szczególnie dobrze radzi sobie z aplikacjami wykonującymi wiele równoległych operacji sieciowych, takimi jak serwery API komunikujące się z wieloma zewnętrznymi usługami jednocześnie.

Dodatkowo Node.js dysponuje menedżerem pakietów npm posiadającym największe repozytorium bibliotek *open source* na świecie. Jest to duża przewaga, która znacząco przyspiesza proces tworzenia aplikacji poprzez dostęp do gotowych rozwiązań dla niemal każdego problemu programistycznego.

### 4.2.2 Express

Express jest minimalistycznym frameworkiem webowym dla środowiska uruchomieniowego Node.js, który zapewnia zestaw niezbędnych narzędzi do szybkiego i prostego budowania serwerów HTTP oraz interfejsów REST API [28]. Powstał on w 2010 roku z inicjatywy T.J. Holowaychuka i rozwijany jest przez społeczność OpenJS Foundation.

Jednym z najważniejszych elementów frameworka jest routing służący do definiowania odpowiedzi serwera na żądania HTTP dla konkretnych ścieżek i metod. Kolejną istotną funkcjonalnością są warstwy pośrednie (*middleware*), będące mechanizmem łańcuchowego przetwarzania żądań przez kolejne funkcje przed dotarciem do docelowego kontrolera. Dzięki

nim możliwa jest implementacja funkcjonalności przekrojowych, takich jak uwierzytelnianie, rejestrowanie zdarzeń, sprawdzanie poprawności danych czy obsługa mechanizmu CORS.

#### 4.2.3 Biblioteki uzupełniające

Ekosystem środowiska Node.js obejmuje zbiór bibliotek realizujących zadania wspólne dla wielu aplikacji serwerowych, niezwiązane bezpośrednio z ich logiką.

Biblioteka `zod` służy do deklaratywnego opisywania oczekiwanej postaci danych oraz weryfikowania jej w czasie działania programu [29]. Zdefiniowany schemat pełni podwójną rolę: sprawdza dane przychodzące i jednocześnie stanowi źródło typu wykorzystywanego przez kompilator języka TypeScript, dzięki czemu deklaracja typu i reguła sprawdzająca nie mogą się rozjeść.

Helmet to zestaw warstw pośrednich ustawiających nagłówki protokołu HTTP związane z bezpieczeństwem [30]. Nagłówki te ograniczają możliwość osadzania strony w ramce, wymuszają połączenie szyfrowane oraz usuwają informacje o zastosowanej technologii serwerowej. Biblioteka `express-rate-limit` realizuje natomiast ograniczanie liczby żądań kierowanych do wskazanych punktów końcowych w zadanym oknie czasowym [31].

Za komunikację z usługami zewnętrznymi odpowiada `axios`, klient HTTP dostępny zarówno w środowisku przeglądarki, jak i Node.js [32]. Udostępnia on mechanizm przechwytywania żądań i odpowiedzi, umożliwiający realizację zadań wspólnych dla wszystkich wywołań, takich jak dołączanie nagłówków czy jednolita obsługa niepowodzeń. Kontrolę pochodzenia żądań zapewnia biblioteka `cors`, implementująca standard *Cross-Origin Resource Sharing* [33].

Biblioteka `dotenv` wczytuje wartości konfiguracyjne z pliku zewnętrznego do zmiennych środowiskowych procesu [34]. Pozwala to przechowywać dane wrażliwe, w tym klucze dostępowe do usług zewnętrznych, poza kodem źródłowym.

Za bezpieczne przechowywanie haseł odpowiada biblioteka `bcrypt`, implementująca funkcję skrótu wyposażoną w parametr kosztu oraz mechanizm soli [47]. Potwierdzanie tożsamości przy kolejnych żądaniach realizowane jest natomiast za pomocą tokenów w standardzie JSON Web Token [19], których podpis pozwala wykryć modyfikację bez przechowywania stanu sesji po stronie serwera.

Token w standardzie JSON Web Token składa się z trzech części: nagłówka określającego algorytm podpisu, ładunku zawierającego dane o użytkowniku oraz podpisu wyznaczonego kluczem znanym wyłącznie serwerowi. Konstrukcja ta pozwala zweryfikować autentyczność tokenu bez przechowywania stanu sesji, ponieważ próba modyfikacji ładunku unieważnia podpis.

### 4.3 Baza danych

Warstwa danych aplikacji została zrealizowana w oparciu o nierelacyjną, dokumentową bazę danych MongoDB oraz bibliotekę Mongoose. Biblioteka ta, pełniąca rolę warstwy ODM (*Object-Document Mapping*), została wykorzystana w celu stworzenia warstwy pośredniej między kodem aplikacji a bazą danych. W kolejnych podrozdziałach opisano charakterystykę obu rozwiązań.

#### 4.3.1 MongoDB

MongoDB to nierelacyjna baza danych dokumentowa należąca do rodziny baz danych NoSQL [35]. Dane przechowywane są w niej w formacie BSON, będącym binarną

reprezentacją formatu JSON. Jest to jednoznaczne z faktem, że każdy rekord w bazie ma postać dokumentu złożonego z par klucz-wartość.

Podstawową cechą MongoDB jest elastyczny schemat danych. Oznacza to, że dokumenty w ramach tej samej kolekcji mogą cechować się zupełnie inną strukturą i innymi polami. Dzięki temu rozwiązanie to jest bardziej adaptacyjne i znacząco różni się od relacyjnych baz danych, które wymagają odgórnie zdefiniowanego i sztywnego schematu tabel. Podejście to sprawdza się szczególnie dobrze w aplikacjach przechowujących dane o zmiennej strukturze.

MongoDB jest sprawdzonym rozwiązaniem komercyjnym, które poza samą bazą danych oferuje obsługę różnorodnych zapytań, indeksowanie, agregację oraz replikację danych. Dostęp do bazy uzyskać można na wiele sposobów, zarówno jako oprogramowanie instalowane lokalnie, jak i jako usługa chmurowa MongoDB Atlas, która eliminuje konieczność zarządzania infrastrukturą bazodanową. Usługa ta udostępnia podstawowy klaster w wersji bezpłatnej.

### 4.3.2 Mongoose

Mongoose to biblioteka typu ODM dla środowiska Node.js, pozwalająca na definiowanie schematów i modeli danych dla bazy MongoDB [46]. Stanowi ona warstwę pośrednią pomiędzy serwerami napisanymi w Node.js a bazą danych, realizując odwzorowanie obiektów języka programowania na dokumenty przechowywane w bazie.

Mimo że MongoDB nie wymaga zdefiniowanego schematu, Mongoose wprowadza opcjonalną warstwę schematów pozwalającą na określenie struktury dokumentów, typów pól i reguł walidacji. Umożliwia to połączenie pożądanых cech rozwiązania, jakimi są elastyczność bazy MongoDB oraz możliwość zachowania podstawowej struktury danych. Dodatkowo Mongoose dostarcza wygodne funkcje wykonywania operacji na bazie danych w postaci metod, takich jak *find*, *findById*, *save* czy *deleteOne*. Istotnym elementem rozwiązania jest także mechanizm warstw pośrednich (*middleware*), pozwalający na wykonywanie operacji przed zapisem dokumentu lub po nim, co umożliwia na przykład wyznaczanie skrótów haseł przed ich utwaleniem w bazie.

## 4.4 Modele językowe

W celu realizacji swojej podstawowej funkcjonalności aplikacja integruje pięć zewnętrznych modeli językowych dostępnych za pośrednictwem interfejsów API. Pomimo wspólnych podstaw technologicznych, każdy z nich udostępniany jest jako komercyjna usługa oparta na dużym modelu językowym. Wszystkie trenowane były na indywidualnych zbiorach danych oraz przy wykorzystaniu różnych metod optymalizacyjnych. W rezultacie zarówno podejście do analizy tekstu, jak i sam wynik mogą znacząco się różnić. Sposób udostępniania usług różni się przy tym między dostawcami część z nich publikuje dedykowane biblioteki klienckie dla popularnych języków programowania, pozostali natomiast udostępniają wyłącznie interfejs REST wymagający bezpośredniego kierowania żądań.

### 4.4.1 OpenAI GPT

GPT (*Generative Pre-trained Transformer*) to rodzina zaawansowanych i dużych modeli językowych opracowana przez firmę OpenAI [36]. Modele GPT oparte są na architekturze transformerowej i trenowane na ogromnych zestawach danych tekstowych metodą uczenia bez nadzoru, a następnie dostrajane z wykorzystaniem informacji zwrotnej od człowieka. Najnowsze modele, zaczynające się od GPT-4, zyskały ogromną popularność i uznanie wśród użytkowników na przestrzeni ostatnich kilku lat od ich debiutu na ogólnodostępnym rynku w

listopadzie 2022 roku. Dostęp do modeli OpenAI realizowany jest za pośrednictwem interfejsu REST API, który wymaga uwierzytelnienia przy użyciu klucza API.

#### **4.4.2 Anthropic Claude**

Claude to rodzina dużych modeli językowych opracowana przez firmę Anthropic, założoną w 2021 roku przez byłych pracowników OpenAI [37]. Są to modele służące do zaawansowanego przetwarzania języka naturalnego. Tym, co wyróżnia to rozwiązanie, jest nie tylko nacisk na bezpieczeństwo oraz zgodność z zasadami etycznymi, ale również szeroka gama produktów oferowanych przez Anthropic. Model posiada wiele iteracji oraz rozwiązań towarzyszących, takich jak narzędzie do pisania kodu Claude Code czy środowisko pracy Claude Cowork. Claude charakteryzuje się dużym oknem kontekstowym, zdolnością do złożonego wnioskowania oraz szczególną dbałością o dokładność i rzetelność generowanych odpowiedzi.

#### **4.4.3 Google Gemini**

Gemini to rodzina multimodalnych modeli językowych opracowana przez Google DeepMind. Zostały one zaprojektowane w taki sposób, aby służyć nie tylko do przetwarzania tekstu, ale także innych typów danych, takich jak obrazy, dźwięk czy wideo [38]. Gemini dostępny jest w kilku wariantach różniących się rozmiarem i możliwościami, od lekkiego Gemini Nano po zaawansowany Gemini Ultra. Architektura zastosowana w modelu umożliwia dodatkowo integrację informacji pochodzących z różnych rodzajów danych, co pozwala na usprawnienie analizy oraz generowania odpowiedzi. Dostęp do modeli Gemini realizowany jest przez Google AI Studio API.

#### **4.4.4 Perplexity**

Perplexity to model językowy opracowany przez firmę Perplexity AI, charakteryzujący się możliwością przeszukiwania Internetu w czasie rzeczywistym, co czyni go przydatnym w zadaniach wymagających bieżących informacji [39]. Pozwala to na uzupełnianie jego wiedzy o informacje niedostępne w danych treningowych, co jest wartościowe w kontekście analizy sentymentu dotyczącej aktualnych wydarzeń. Perplexity dostępny jest przez dedykowane API i oferuje modele zoptymalizowane zarówno pod kątem szybkości, jak i jakości odpowiedzi.

#### **4.4.5 Grok**

Grok to rodzina dużych modeli językowych stworzonych przez firmę xAI [40]. Modele te charakteryzują się dostępem do platformy X, dawniej Twitter, będącej jedną z najpopularniejszych platform społecznościowych na świecie. Skutkuje to możliwością korzystania z treści publikowanych na bieżąco, związanych z aktualnymi wydarzeniami. Dostęp do modeli realizowany jest za pośrednictwem dedykowanego interfejsu API zgodnego ze standardem stosowanym przez OpenAI.

### **4.5 Źródła danych**

Aplikacja integruje się z trzema zewnętrznymi źródłami danych internetowych, z których pozyskiwane są treści wejściowe dla modeli, stanowiące podstawę analizy sentymentu. Podrozdział przedstawia charakterystykę każdego z zastosowanych źródeł oraz udostępnianych przez nie interfejsów API. Źródła dobrano tak, by obejmowały treści o odmiennym charakterze: swobodne wypowiedzi na forum dyskusyjnym, komentarze pod materiałami wideo oraz publikacje prasowe.

#### **4.5.1 Reddit API**

Reddit to jedna z największych platform społecznościowych na świecie, mająca charakter forum dyskusyjnego. Umożliwia ona publikowanie, ocenianie i komentowanie treści w ramach grup tematycznych, nazywanych subredditami. Platforma, ze względu na nieformalny charakter wypowiedzi i autentyczność opinii niepoddanych zbyt intensywnej moderacji, jest wartościowym źródłem danych w kontekście analizy sentymentu. Jeden z głównych mechanizmów platformy polega na wyrażaniu aprobaty lub dezaprobaty wobec poszczególnych wpisów, dzięki czemu komentarze o najwyższej ocenie wyświetlane są jako pierwsze w danym wątku.

Reddit udostępnia oficjalne REST API umożliwiające dostęp do postów, komentarzy oraz metadanych [15]. Dostęp do API wymaga rejestracji aplikacji w panelu programisty oraz uzyskania danych uwierzytelniających w standardzie OAuth 2.0.

#### **4.5.2 YouTube Data API**

YouTube to największy serwis wideo na świecie należący do Google, służący do publikowania filmów i umożliwiający prowadzenie dyskusji w sekcji komentarzy. Komentarze te stanowią wartościową informację na temat reakcji i emocji wywołanych przez poruszany w materiale temat.

YouTube Data API v3 to oficjalny interfejs programistyczny udostępniany przez Google, umożliwiający wyszukiwanie filmów oraz pobieranie metadanych i komentarzy [17]. Dostęp realizowany jest za pośrednictwem klucza API generowanego w konsoli Google Cloud. Interfejs udostępnia punkt końcowy wyszukiwania, zwracający listę filmów powiązanych z podanym zapytaniem, oraz punkt końcowy komentarzy, umożliwiający pobranie komentarzy do konkretnego filmu.

#### **4.5.3 NewsAPI**

NewsAPI to usługa udostępniająca dostęp do artykułów pochodzących z tysięcy źródeł prasowych z całego świata [18]. Platforma indeksuje materiały z popularnych serwisów informacyjnych, blogów oraz czasopism internetowych, umożliwiając ich przeszukiwanie według słów kluczowych, daty publikacji, języka oraz źródła. NewsAPI udostępnia REST API zwracające tytuły, opisy oraz treści artykułów wraz z odnośnikami do materiałów źródłowych.

## 5. Implementacja i projekt techniczny

Rozdział opisuje szczegółową techniczną realizację rozwiązania, którego założenia zostały opisane szerzej w rozdziale 3. Celem postawionym tutaj jest wykazanie, że przyjęte założenia zostały zrealizowane, oraz wskazanie konkretnych rozwiązań, które pozwoliły to osiągnąć. Jego układ został zorganizowany tak, aby odpowiadał kolejności decyzji projektowych, a nie strukturze projektu. Rozdział rozpoczyna omówienie organizacji samego kodu oraz odseparowania logiki dziedzinowej od infrastruktury. Następnie opisane zostały punkty rozszerzeń, za pomocą których dodawane są modele językowe i źródła danych. Kolejno omówiona została logika analizy sentymentu, stanowiąca kluczowy element pracy, wraz z opisem przebiegu pojedynczej analizy oraz zastosowanych zabezpieczeń. Rozdział zamyka opis narzędzia badawczego, które jest podstawą do realizacji rozdziału 6.

Każde omawiane zagadnienie przedstawiono zgodnie z jednolitym schematem, w którym opisano:

- problem, który jest rozstrzygany;
- rozwiązanie, które zostało przyjęte, wraz z odrzuconą alternatywą;
- sposób działania;
- konsekwencje opisanej wcześniej realizacji.

### 5.1 Organizacja kodu

Zaproponowana struktura projektu obrazuje nie tylko podział techniczny na warstwy, ale przede wszystkim odpowiada rozdzieleniu odpowiedzialności pomiędzy nimi. Podrozdział przedstawia sposoby organizacji kodu, ze szczególnym naciskiem na odpowiednie wydzielanie definicji współdzielonych zarówno przez warstwę frontendową, jak i backendową, a także na zasady ograniczające zależności logiki analizy od pozostałych elementów systemu.

#### 5.1.1 Podział na warstwy i typy współdzielone

System został złożony z dwóch niezależnych od siebie aplikacji, które nie współdzieliły pamięci ani kompilacji. Pomiedzy nimi występuje tylko jeden łącznik, zrealizowany przy wykorzystaniu protokołu sieciowego HTTP oraz formatu danych JSON. Wynikiem tego jest problem niewystępujący w programach jednolitych, jakim jest brak mechanizmu weryfikującego zgodność deklaracji danych wśród aplikacji. Rozwiązaniem tego jest zastosowanie kontraktu współdzielonego przez obie warstwy. Równolegle do frontendu i backendu powstał katalog Shared, zawierający definicje typów opisujących dane wymieniane pośród nich. Katalog ten nie jest oddzielną warstwą, a elementem wspólnym dołączanym do obu projektów za pomocą aliasu ścieżki `@shared/*`. Dzięki temu ewentualne rozbieżności pomiędzy częścią kliencką a serwerem zmieniają się z błędu wykonania w błąd kompilacji.

Warstwa frontend została zaprezentowana na rysunku 6. Składa się ona z następujących katalogów:

- `api` — moduły wywołujące punkty końcowe warstwy serwerowej, po jednym na obszar funkcjonalny;
- `components` — komponenty wielokrotnego użytku, w tym elementy prezentacyjne wydzielone do podkatalogu `ui` oraz komponenty rozstrzygające o dostępie do tras;
- `context` — stan współdzielony przez całą aplikację, w szczególności dane zalogowanego użytkownika;

- `hooks` — funkcje realizujące pobieranie danych wraz z obsługą stanów pośrednich, odseparowane od układu widoków;
- `lib` — wspólny klient komunikacji sieciowej oraz funkcje formatujące wyniki analizy;
- `pages` — komponenty odpowiadające poszczególnym widokom aplikacji;
- `services` — logika widoku wykraczająca poza pojedynczy komponent;
- `types` — definicje typów wykorzystywane wyłącznie po stronie klienta.



Rysunek 6. Struktura warstwy frontend aplikacji. Źródło: opracowanie własne

Strukturę warstwy serwerowej przedstawia rysunek 7. Zastosowany podział odzwierciedla rozdzielenie logiki dziedziny, punktów rozszerzeń oraz infrastruktury:

- `domain` — logika analizy sentymentu i obliczeń statystycznych, pozbawiona zależności od infrastruktury, omówiona w podrozdziale 5.1.2;
- `providers` — kontrakt dostawcy analizy wraz z implementacjami dla poszczególnych modeli językowych oraz rejestrem, opisany w podrozdziale 5.2;
- `sources` — kontrakt źródła danych wraz z implementacjami i rejestrem;

- *services* — koordynacja kolejności i współbieżności wywołań oraz utrwalanie wyników;
- *routes* — definicje punktów końcowych interfejsu programistycznego;
- *validation* — schematy sprawdzające poprawność danych przychodzących;
- *middleware* — warstwy pośrednie realizujące uwierzytelnianie, ograniczanie liczby żądań i obsługę błędów;
- *models* — schematy dokumentów przechowywanych w bazie danych;
- *config* — walidacja zmiennych środowiskowych wykonywana przy uruchomieniu;
- *lib* — narzędzia wspólne: klasyfikacja awarii usług zewnętrznych, dziennik zdarzeń, generator liczb pseudolosowych;
- *research* — narzędzie badawcze uruchamiane niezależnie od aplikacji, opisane w podrozdziale 5.6.



Rysunek 7. Struktura warstwy backend. Źródło: opracowanie własne

Ponadto opisana wcześniej zasada jednego źródła została rozszerzona na wartości liczbowe służące do określania skali ocen w analizie. Zasady opisujące, w jaki sposób modele wyrażają analizowane treści, wykorzystywane są w wielu miejscach na przestrzeni całej

aplikacji, takich jak schemat bazy danych czy schemat walidacji żądań. Umieszczenie ich definicji w jednym miejscu, przedstawione na listingu 1, pozwoliło uniknąć wielokrotnego definiowania tych samych wartości, zachowując spójność oraz ułatwiając późniejsze zmiany.

Listing 1. Definicja skali ocen

```
export const SCORE_MIN = -10;
export const SCORE_MAX = 10;
export const SCORE_RANGE = SCORE_MAX - SCORE_MIN;
export const CONFIDENCE_MIN = 0;
export const CONFIDENCE_MAX = 100;
```

Wartość score range zapisano jako wypadkową dwóch zmiennych, dzięki czemu jest ona zabezpieczona przed sytuacją, w której zmiana jednej z granic wymagałaby dodatkowej zmiany. Konsekwencje wprowadzonego podziału spowodowały, że skala znajduje się w jednym miejscu i jest ona współdzielona wraz z pozostałymi zawartościami folderu shared na przestrzeni całego projektu.

### 5.1.2 Izolacja warstwy domenowej

Logika związana z analizą sentymentu składa się z ustrukturyzowania instrukcji, która jest przekazywana dalej modelom językowym, odczytania zwróconej przez nie odpowiedzi, wyznaczenia oceny oraz klasyfikacji uzyskanego wyrażenia zgodnie z wytycznymi. Reguły te wynikają bezpośrednio z założeń, które zostały uformowane w rozdziale 3. Całość została umieszczona w katalogu domain, który jest oddzielony od pozostałej warstwy serwerowej. Wynika to z konieczności utworzenia uniwersalnego modułu, który może być wykorzystany zarówno przez aplikację serwerową, jak i moduł badawczy będący kluczowym elementem realizacji rozdziału 6.

Rozdzielenie to polega na konkretnym uformowaniu kierunku zależności tak, aby elementy serwerowe, takie jak baza danych czy framework obsługujący żądania HTTP, mogły z niego korzystać, natomiast aby on nie wiedział o ich istnieniu i był całkowicie autonomiczny. Rozwiązanie to zostało tak zaprojektowane, aby nie było konieczne odtworzenie logiki analizy w module badawczym. Niesie to ze sobą dwie bezpośrednie korzyści: mniejszą objętość kodu wraz z bardziej spójnym projektem oraz wykluczenie jakichkolwiek różnic pomiędzy dwoma potencjalnymi modułami.

Alternatywą dla takiego rozwiązania byłoby umieszczenie tej logiki bezpośrednio w funkcjach obsługujących poszczególne punkty końcowe. Kod byłby wtedy w jednym miejscu, ale jego ponowne wykorzystanie w module badawczym byłoby niemożliwe do zrealizowania.

Rozwiązanie to ma jednak swoją cenę, jaką jest konieczność przekazywania dodatkowych parametrów w sygnaturach. Przykładem jest funkcja porządkująca nazwy źródeł danych przed ich zapisem w bazie, zaprezentowana na listingu 2. Nie posiada ona informacji o dostępnych źródłach, ponieważ zapisane są one w rejestrze należącym do warstwy infrastruktury. Import rejestru wprowadziłby natomiast do warstwy domenowej zależności od bibliotek komunikacji sieciowej i modułu konfiguracji, czyli dokładnie te, których odseparowanie miało uniknąć.

Listing 2. Sygnatura funkcji porządkującej nazwy źródeł danych

```
function normalizeSources(
  sources: readonly string[],
  knownSources: readonly string[]
): string[] {
  ...
}
```

```
}
```

Ewentualne zapisanie listy źródeł bezpośrednio w module domenowym sprawiłoby, że w przypadku dodania nowego źródła wymagana byłaby modyfikacja logiki analizy, co zniosłoby korzyść szerzej opisaną w podrozdziale 5.2. Przyjęte rozwiązanie polega na przekazaniu dodatkowego parametru wywołania, dostarczanego przez warstwę wywołującą, dla której odwołanie do rejestru nie stanowi naruszenia kierunku zależności. Kosztem jest więc dodatkowy parametr, niewynikający z samego zadania funkcji, lecz z przyjętego ograniczenia.

## 5.2 Punkty rozszerzeń systemu

Założenie o rozszerzalności systemu, przedstawione w podrozdziale 3.2.3, wymagało zdecydowania, w którym miejscu kodu ma zostać zapisana wiedza o tym, jakie modele językowe i jakie źródła danych są w systemie dostępne. Podrozdział omawia przyjęte w tym celu rozwiązania oraz jego wpływ na pozostałe elementy systemu, a także koszt dołączenia kolejnej pozycji.

### 5.2.1 Kontrakty dostawcy analizy i źródła danych

Charakterystyka systemu wymaga od niego integracji z wieloma usługami zewnętrznymi. W obecnej konfiguracji jest to osiem pozycji: pięć modeli językowych oraz trzy źródła treści. Każde z nich posiada swoje charakterystyczne cechy, takie jak sposób uwierzytelniania, własne adresy usług i format odpowiedzi. Dodatkowo niektóre rozwiązania są wyposażone w dedykowane biblioteki klienckie, podczas gdy inne wymagają wykorzystania bezpośrednio protokołu HTTP.

Po raz kolejny przyjęte rozwiązanie zostało zrealizowane w oparciu o kontrakt programistyczny. Wszelkie unikatowe elementy dla danej pozycji zamknięto w dedykowanym module, podczas gdy resztę informacji wyciągnięto do kontraktu będącego podstawą realizacji integracji z usługą zewnętrzną. Dzięki temu w momencie dodania, usunięcia bądź edycji konkretnego dostawcy wystarczy dokonać zmian w jednym miejscu, zamiast w wielu.

Interfejs przedstawiony na listingu 3 określa jedną operację, jaką jest przyjęcie zestawu tekstów ze źródeł i zwrócenie odpowiedzi modelu w postaci nieprzetworzonej. Celowo pomijane są tutaj dalsze kroki, takie jak budowa instrukcji systemowej czy odczyt zwróconej odpowiedzi, jako że realizowane są one przez warstwę domenową wspólnie dla wszystkich dostawców, co omówiono w podrozdziale 5.3.

Listing 3. Kontrakt dostawcy analizy sentymentu

```
export interface SentimentProvider {  
  readonly name: string;  
  readonly modelId: string;  
  readonly maxOutputTokens: number | null;  
  readonly supportsTemperature: boolean;  
  analyze(texts: string[]): Promise<string>;  
}
```

Reszta pól kontraktu służy do opisu dostawcy potrzebnego do dokumentowania otrzymywanych wyników:

- `name` — identyfikator używany wewnątrz systemu, zapisywany wraz z każdym wynikiem analizy;
- `modelId` — oznaczenie wersji modelu nadane przez dostawcę;

- `maxOutputTokens` — górna granica długości generowanej odpowiedzi, przy czym wartość pusta oznacza pozostawienie ustawienia domyślnego dostawcy;
- `supportsTemperature` — informacja o tym, czy dostawca przyjmuje parametr określający stopień losowości generowania.

Dwa pierwsze pola są wymagane dla uzyskania porównywalności pomiędzy rezultatami opisywanymi szerzej w rozdziale 6. Pozwala to na przechowanie informacji o wykorzystanym modelu, co umożliwi rozróżnienie ich podczas badania wyników. Ostatnie pole natomiast przechowuje informację o potencjalnej rozbieżności pomiędzy modelami, wynikającej z faktu, że nie każdy model przyjmuje parametr związany z temperaturą. Pozwala to również odnotować, czy analiza powstała z jego wykorzystaniem.

Interfejs źródeł zobrazony na listingu 4 został zawężony w związku z brakiem wymogu metadanych opisujących wersję źródła. Kluczowym elementem jest tutaj ujednolicenie postaci jaką przyjmuje wynik.

Listing 4. Kontrakt źródła danych

```
export interface SourceItem {
  content: string;
  link: string;
}
export interface SourceProvider {
  readonly name: string;
  fetch(query: string): Promise<SourceItem[]>;
}
```

Jak widać na listingu, każde źródło zwraca treść wraz z odsyłaczem do materiału źródłowego, niezależnie od jego pochodzenia. Umożliwia to realizację wszystkich źródeł jednakowo oraz sprawia, że wszystkie, pomimo różnej charakterystyki, mogą być traktowane tak samo. Dzięki temu dalsze etapy przetwarzania, takie jak ustalenie kolejności, ograniczenie liczby pozycji czy złożenie instrukcji dla modeli, traktują pozyskane materiały jednakowo.

### 5.2.2 Rejestry i ich konsumenci

Rejestry przechowują wiedzę o dostawcach modeli oraz źródeł dostępnych w aplikacji. Każdy z nich posiada własny plik, w którym zawarte są dotyczące go dane.

Listing 5 prezentuje realizację rejestru dla dostawców analizy sentymentu. Lista `PROVIDERS` zawiera w sobie moduły konkretnych utworzonych już rozwiązań. Ich nazwy nie są zapisywane odrębnie, a jedynie odczytywane z pola `name` samego dostawcy, dzięki czemu nie może dojść do rozbieżności między nazwą zadeklarowaną a rzeczywistą.

Listing 5. Rejestr dostawców analizy

```
const PROVIDERS: readonly SentimentProvider[] = [
  openaiProvider,
  claudeProvider,
  geminiProvider,
  grokProvider,
  perplexityProvider,
];
const registry: ReadonlyMap<string, SentimentProvider> = new Map(
  PROVIDERS.map(provider => [provider.name, provider])
);
export function getProvider(name: string): SentimentProvider |
undefined {
```

```

    return registry.get(name);
  }
  export function listProviders(): readonly SentimentProvider[] {
    return PROVIDERS;
  }

```

Udostępnione są tutaj dwie operacje: odszukanie dostawcy po nazwie oraz zwrócenie pełnego zestawienia. Na nich opierają się wszystkie miejsca systemu odwołujące się do dostawców:

- warstwa koordynująca przebieg analizy — odczytuje zestawienie i kieruje zapytanie do każdej wskazanej pozycji, wobec czego nie zawiera rozgałęzienia sterowania ani żadnej nazwy modelu;
- warstwa punktów końcowych — nazwa dostawcy stanowi parametr ścieżki adresu, sprawdzany względem rejestru, wobec czego każda pozycja uzyskuje własny punkt końcowy bez definiowania go osobno;
- warstwa prezentacji — zestawienie dostępnych nazw pobierane jest z osobnego punktu końcowego, przedstawionego na listingu 6;
- moduł statystyk — pełny wykaz wyznacza kształt zestawienia porównawczego, dzięki czemu obejmuje ono również pozycje, dla których nie zgromadzono jeszcze żadnych danych.

Listing 6. Punkt końcowy udostępniający zestawienie dostawców

```

analyzeRouter.get('/providers', authenticateToken, (_req, res) => {
  res.json({ providers: listProviderNames() });
});

```

Dzięki takiemu zastosowaniu informację o dostępnych dostawcach można przechowywać wyłącznie w warstwie serwerowej, bez powielania jej po stronie klienta. Pozwala to na proste i mało kosztowne dodawanie kolejnych pozycji bez zmian w aplikacji webowej.

Rejestr źródeł jest skonstruowany w oparciu o takie same zasady i korzysta z analogicznych rozwiązań, udostępniając takie same operacje. Dzięki temu wiedza o źródłach przechowywana jest w jednym miejscu, przez co wszelkie elementy, takie jak warstwa koordynująca analizę czy punkty końcowe, odwołują się zawsze do tego samego miejsca, stanowiącego jedyne źródło wiedzy o dostępnych w systemie pozycjach.

### 5.2.3 Koszt rozszerzenia

Celem zastosowanych rozwiązań było zminimalizowanie zakresu wymaganych zmian przy dodawaniu nowych pozycji oraz edycji tych istniejących. Ponadto ważnym elementem było zachowanie uniwersalności i generyczności rozwiązania względem użycia w aplikacji webowej i module badawczym. Weryfikacja takiego założenia wymaga jednak wskazania liczby plików, które trzeba zmodyfikować podczas dokonywania operacji zmian na dostawcach.

Zostało to zrealizowane doświadczalnie poprzez dołączenie tymczasowego testowego dostawcy analizy oraz źródła danych. Następnie sprawdzono w jakich miejscach w aplikacji pojawiły się nowe pozycje, a które wymagały dodatkowych zmian. Wyniki zostały zestawione w tabeli 2.

Tabela 2. Zakres zmian wymaganych przy rozszerzeniu systemu

| Rodzaj zmiany                   | Pliki nowe | Pliki modyfikowane |
|---------------------------------|------------|--------------------|
| Dołączenie modelu językowego    | 1          | 1                  |
| Dołączenie źródła danych        | 1          | 1                  |
| Zastąpienie dostawcy innym      | 1          | 1                  |
| Zmiana wersji modelu u dostawcy | 0          | 1                  |

Dołączenie lub zastąpienie modelu językowego oraz źródła danych wymaga utworzenia modułu realizującego kontrakt dla dostawcy analizy bądź źródła danych oraz dodania nowej pozycji w rejestrze. W przypadku podmiany modelu w istniejącej pozycji wystarczy zmiana w jednym miejscu. Dzięki temu nienaruszone pozostają następujące elementy systemu:

- warstwa koordynująca przebieg analizy, która iteruje po zestawieniu z rejestru;
- definicje punktów końcowych, ponieważ nazwa dostawcy stanowi parametr ścieżki adresu;
- interfejs użytkownika, który zestawienie dostępnych pozycji pobiera z warstwy serwerowej;
- moduł statystyk, wyznaczający kształt zestawienia porównawczego na podstawie rejestru;
- schemat bazy danych, przechowujący nazwę dostawcy jako wartość tekstową bez ograniczenia do zbioru dopuszczalnych nazw.

Należy jednak wspomnieć o pewnym ograniczeniu, jakie stanowi ostatnia z wymienionych pozycji. Dane historyczne zapisują się wraz z nazwą dostawcy, dlatego w przypadku jej zmiany dane zgromadzone wcześniej przestaną być powiązane z nowymi. Nie stanowi to jednak przeszkody dla systemu oraz badań w obecnym kształcie. Może to jednak stanowić problem w kontekście przyszłego rozwoju aplikacji i ewentualnych zmian w dostawcach.

## 5.3 Warstwa domenowa analizy sentymentu

W podrozdziale 5.1.2 wskazano elementy tworzące warstwę domenową, stanowiącą najważniejszą część systemu. Obecny podrozdział przybliży jej zawartość oraz to, w jaki sposób zarządza ona logiką związaną z analizą sentymentu. Omówiono w nim sposób formułowania zadań przekazywanych modelom, odczytu zwracanych przez nie odpowiedzi, wyznaczania oceny wraz z klasyfikacją wydźwięku oraz ustalania materiału poddawanego analizie. Elementy te są wspólne dla wszystkich dostawców, wobec czego moduł konkretnego modelu nie zawiera własnej logiki analizy.

### 5.3.1 Konstrukcja instrukcji przekazywanej modelom

Rolą modelu językowego jest przyjęcie tekstu oraz jego zwrócenie. Wykorzystanie go jako elementu systemu wymaga zweryfikowania postaci, w jakiej zwraca on swoją odpowiedź, oraz rozłożenia jej na konkretne pola. W tym celu stworzona została instrukcja, w której znaleźć można opis zasad oceniania sentymentu oraz format odpowiedzi, jakiego oczekuje aplikacja. Celem instrukcji jest określenie roli, jaką ma pełnić model, oraz wytycznych obowiązujących podczas przeprowadzania analizy. Dodatkowo zawarty jest tam dokładny format odpowiedzi,

jaka ma być zwracana wraz z treściami źródłowymi. Jest to jednakowe dla wszystkich dostawców, aczkolwiek sposób jej przekazania może się różnić w zależności od rozwiązań programistycznych oferowanych przez dostawcę.

Instrukcja została zdefiniowana w jednym miejscu w całym projekcie, konkretnie w warstwie domenowej. Rozwiązanie alternatywne, polegające na zdefiniowaniu jej odrębnie w module każdego dostawcy, byłoby prostsze w realizacji, jednak ewentualna rozbieżność pomiędzy dostawcami zakłóciłaby badanie porównawcze przeprowadzone w rozdziale 6. Dodatkowo utrudniałoby to utrzymanie i rozwój systemu.

Jednym z jej elementów są progi liczbowe opisujące skalę, jaką model ma wykorzystać podczas oceny sentymentu w tekście. Wartości te nie są zdefiniowane wprost, lecz wyznaczane na podstawie skali opisanej szerzej w podrozdziale 5.1.1, co przedstawia listing 7. Celowo uniknięto zapisywania ich na stałe, jako że progi liczbowe mogą ulec zmianie, a skala przestałaby być na nią odporna i zwracane przez nią wartości byłyby nie tyle mylące, co błędne.

Listing 7. Wyznaczanie progów skali w treści instrukcji

```
const EXTREME_POSITIVE = `+${SCORE_MAX - 2} to +${SCORE_MAX}`;  
const VERY_POSITIVE   = `+${SCORE_MAX - 5} to +${SCORE_MAX - 3}`;  
const EXTREME_NEGATIVE = `${SCORE_MIN + 2} to ${SCORE_MIN}`;  
const VERY_NEGATIVE   = `${SCORE_MIN + 5} to ${SCORE_MIN + 3}`;
```

Instrukcja określa wymagany format odpowiedzi, przedstawiony na listingu 8. Narzucenie sztywnej postaci pozwala odczytać poszczególne pola niezależnie od dostawcy, u którego wywołanie zostało zrealizowane.

Listing 8. Fragment instrukcji systemowej definiującej wymagany format odpowiedzi modeli językowych

```
Provide your analysis in EXACTLY this format:  
Sentiment: [positive/negative/neutral/mixed]  
Score: [number from -10 to +10, where -10 is extremely negative, 0 is  
neutral, +10 is extremely positive]  
Confidence: [0-100, how confident you are in this assessment]  
Analysis: [2-4 sentences explaining the sentiment and key emotional  
indicators you found]  
Justification:  
- [YOUR first key reason why you gave this specific score and  
sentiment assessment]  
- [YOUR second reasoning point explaining the sentiment]  
- [YOUR third reasoning point, or leave empty if not applicable]  
Perspectives:  
[If multiple distinct perspectives detected:]  
Label / Sentiment / Score / Weight / Description  
---  
[Repeat for each perspective, separated by "---"]  
[If no distinct perspectives, write "None"]
```

Odrębny fragment instrukcji, widoczny na listingu 9, opisuje zasady wykrywania perspektyw oraz sposób przypisywania im wag. Zasady przypisywania wartości liczbowej, określające opisy słowne dla poszczególnych przedziałów skali, zapisano w instrukcji zgodnie z założeniami przedstawionymi w podrozdziale 3.4.3.

Listing 9. Fragment instrukcji systemowej definiującej zasady wykrywania perspektyw

```
PERSPECTIVE DETECTION (CRITICAL):  
- Look for ALL sides of the issue, not just the dominant one
```

- In sports: detect BOTH winners' fans (positive) AND losers' fans(negative)
- In politics: detect BOTH supporters AND critics
- In products: detect BOTH enthusiasts AND critics
- In debates: detect ALL competing viewpoints
- Weight = (confidence in detection) × (prevalence in data %)
- Aim for 2-4 perspectives when viewpoints exist
- If you only find ONE perspective but the topic is inherently two-sided(sports match, political figure, competitive products), actively search for the OTHER side too
- If no distinct perspectives detected, write "None"

Odrębnego omówienia wymaga element służący do przypisywania oceny neutralnej, ze względu na fakt, że może być ona rozpatrywana na dwa różne sposoby. Wynika to z faktu, że modele językowe przypisują wartość neutralną nie tylko podczas braku sentymentu w tekście, ale również w sytuacji, kiedy wypadkowa skrajnych emocji jest zrównoważona. Nie oznacza to, że tekst jest neutralny, ale emocje pozytywne równoważą te negatywne, co czyni tekst trudnym do jednoznacznego sklasyfikowania jedną wartością liczbową. Dlatego instrukcja wyraźnie zaznacza, aby model zawsze przypisał wartość pozytywną lub negatywną, nawet w sytuacji, gdzie trudno odnaleźć emocje albo są one mocno równoważące.

Decyzja ta stanowi pewnego rodzaju ingerencję w sposób oceniania przez modele i ma realny wpływ na uzyskiwane wartości. Należy o tym pamiętać podczas analizy wyników w rozdziale 6, gdzie możliwość neutralnego oceniania jest zaburzona przez instrukcję. Ważny do odnotowania jest również fakt, że każdy model posiada takie same wytyczne, więc nie występuje żadna rozbieżność pomiędzy modelami. Instrukcja podkreśla ponadto metodykę, zgodnie z którą model ma poszukiwać różnych punktów widzenia, nazywanych inaczej perspektywami, opisanymi szerzej w podrozdziale 3.4.2, oraz zasadę wyznaczania ich wag. Stanowi ona iloczyn częstotliwości występowania danego poglądu oraz pewności jego rozpoznania.

Wszystkie treści przekazywane modelom podlegają wcześniej jednakowym przygotowaniu, obejmujące usunięcie nadmiarowych znaków podziału wiersza oraz odrzucenie pozycji pustych. Realizuje je funkcja wspólna dla wszystkich dostawców, dzięki czemu różnice w postaci materiału wejściowego nie stanowią czynnika zakłócającego porównanie.

### 5.3.2 Odczyt odpowiedzi i sygnalizowanie braków

Odpowiedź modelu zawsze zwracana jest w postaci tekstu o z góry narzuconym przez instrukcję formacie. Zadaniem modułu odczytu jest rozłożenie odpowiedzi na składowe pola rekordu. Z pozoru prosta czynność jest narażona na niepowodzenia o niewidocznych na pierwszy rzut oka skutkach. W przypadku błędnego sformułowania odpowiedzi przez model nie każde pole zostanie dobrze odczytane. Nie zwróci to jednak błędu i niekompletny rekord zostanie zapisany w bazie. Dopiero po weryfikacji widoczna będzie wybrakowana odpowiedź. Poszczególne pola oznaczone są w instrukcji etykietami odpowiadającymi klasyfikacji, ocenie, poziomowi pewności, opisowi oraz uzasadnieniu, a wykryte perspektywy rozdzielone są ustalonym separatorem.

Podstawowym mechanizmem rozstrzygającym jest sposób postępowania z polem, które nie zostało odczytane. Ewentualne zastąpienie takiego pola wartością domyślną w postaci zera i neutralną oceną doprowadziłoby do powstania danych, które nie powinny istnieć i są niezgodne ze stanem faktycznym. Przyjęte rozwiązanie jest zatem odwrotne do opisanego i celowo tworzy lukę z brakiem danych, co jest od razu widoczne i odróżnialne od poprawnie odczytanych wartości. Rekord zawiera ponadto dwa pola opisujące przebieg odczytu, przedstawione na listingu 10.

Listing 10. Sygnalizowanie pól nieodczytanych

```
const parseErrorFields: ParsableField[] = [];
if (sentiment === null) parseErrorFields.push('sentiment');
if (score === null) parseErrorFields.push('score');
if (confidence === null) parseErrorFields.push('confidence');
if (analysis === null) parseErrorFields.push('analysis');
return {
  ...
  parseError: parseErrorFields.length > 0,
  parseErrorFields
};
```

Odniesienie dotyczy nie tylko okoliczności błędu, ale również dokładnych pól, których dotyczyło dane niepowodzenie. Szczególnie przydatne jest to podczas przeprowadzania setek analiz jednocześnie, gdzie łatwo jest określić trendy błędów i wykonać konieczne naprawy, aby usprawnić logikę. Rekordy oznaczone jako niepełne są następnie pomijane przy wyznaczaniu statystyk, co omówiono w podrozdziale 5.4.3.

Pomimo wyraźnych i jednoznacznych założeń odnośnie odpowiedzi wprowadzanych przez instrukcję, niektóre modele wyróżniają znacznikami języka Markdown zwracane przez siebie wartości. W tym celu zastosowano wzorce odczytywania danych, które dopuszczają opcjonalne wyróżnienie takiego tekstu. Jest to o tyle istotne, że bez zabezpieczenia doszłoby do sytuacji, w której pomimo że model zwraca wszelkie potrzebne dane, nie są one odczytywane, w wyniku czego zostają utracone przez aplikację.

Format odpowiedzi wykorzystuje te same etykiety w dwóch miejscach: przy ocenie całości materiału oraz przy każdej wykrytej perspektywie. Źle dobrane wyrażenie regularne mogłoby odszukać błędne wartości i nadpisać je poprzez pierwsze dopasowanie. Przed odczytem pól głównych treść odpowiedzi jest więc przycinana do fragmentu poprzedzającego wykaz perspektyw, natomiast same perspektywy odczytywane są z odpowiedzi pełnej.

Jeżeli perspektywa nie posiada jakiegokolwiek z wymaganych pól, zostanie ona odrzucona w całości i nie będzie brana pod uwagę. Uzupełnienie jakiegokolwiek wartości domyślną byłoby przekłamanie i nie stanowiłoby wymiernej wartości. Odrzucenie takiej pozycji nie stanowi jednak błędu i pozwala aplikacji pracować dalej.

### 5.3.3 Wyznaczanie oceny i klasyfikacja wydźwięku

Po odczytaniu danych zwróconych przez modele językowe system dysponuje oceną wystawioną przez model oraz zestawem wykrytych perspektyw z własnymi ocenami. Dzięki temu istnieją dwie wielkości mogące służyć do opisu tego samego materiału, w wyniku czego konieczne jest rozstrzygnięcie, która jest uznawana za właściwy wynik analizy.

W tym celu przyjęte zostało rozwiązanie polegające na zachowaniu obu wartości równolegle. Ocena wystawiona bezpośrednio przez model obrazuje jego całościowe postrzeganie tekstu, natomiast wartość ważona opisuje wypadkową wykrytych perspektyw oraz ich wag wyliczonych przez model. Alternatywnie rozważano traktowanie oceny wynikającej z perspektyw i wag jako głównej, natomiast ocena zwracana bezpośrednio przez model wykorzystywana byłaby tylko w sytuacji, kiedy nie byłoby żadnych perspektyw. Rozwiązanie to niesie jednak ze sobą podstawowe zagrożenie, jakim jest zestawianie dwóch różnych wartości podczas przeprowadzania analizy w rozdziale 6. Odczytując dane, nie sposób ustalić, która wartość została zapisana w bazie.

Sposób liczenia średniej ważonej został zaprezentowany na listingu 11. Brak perspektyw oraz zerowa suma wag prowadzą do wartości pustej, nie zaś do zera, przez co średnia zbioru

pustego pozostaje nieokreślona, a jej zapisanie jako wartości liczbowej wprowadziłoby do danych obserwację, której nie było. Warunek ten stanowi dodatkowo zabezpieczenie z punktu widzenia matematycznego, jako że dzielenie przez zero zakłóciłoby kolejne etapy przetwarzania wyniku.

Listing 11. Wyznaczanie oceny ważonej

```
export function aggregatePerspectives(  
  perspectives: readonly Perspective[]  
) : number | null {  
  if (perspectives.length === 0) {  
    return null;  
  }  
  const totalWeight = perspectives.reduce((sum, p) => sum +  
    p.weight, 0);  
  if (totalWeight === 0) {  
    return null;  
  }  
  const totalWeightedScore = perspectives.reduce(  
    (sum, p) => sum + (p.score * p.weight), 0  
  );  
  return totalWeightedScore / totalWeight;  
}
```

Samo zaokrąglanie wyniku dokonywane jest dopiero w warstwie prezentacji, podczas wyświetlania go użytkownikowi. Zastosowanie go przed zapisem byłoby nieodwracalne i usunęłoby różnicę między wartościami, które przy przyjętych progach klasyfikacji znajdują się po przeciwnych stronach granicy.

Progi opisane w podrozdziale 5.3.1 mają na celu wyznaczenie sposobu, w jaki model dobiera wartość liczbową. Wartość ważona natomiast jest wypadkową wykrytych perspektyw oraz ich wag, co zostało zaprezentowane na listingu 12.

Listing 12. Klasyfikacja wydźwięku

```
export function classifySentiment(  
  score: number,  
  perspectives: readonly Perspective[] = []  
) : Sentiment {  
  const hasPositive = perspectives.some(p => p.score > 0);  
  const hasNegative = perspectives.some(p => p.score < 0);  
  const hasOpposingViews = hasPositive && hasNegative;  
  if (hasOpposingViews && score >= -2 && score <= 2) {  
    return 'mixed';  
  }  
  if (score >= 2) return 'positive';  
  if (score <= -2) return 'negative';  
  return 'neutral';  
}
```

Warunek dotyczący perspektyw przeciwstawnych realizuje założenie przedstawione w podrozdziale 3.4.2. Wartość bliska zero powstaje bowiem w dwóch odmiennych sytuacjach: przy materiale pozbawionym zabarwienia emocjonalnego oraz przy materiale, w którym dwie wyraziste, przeciwstawne perspektywy wzajemnie się znoszą. Sama ocena liczbową nie pozwala ich rozróżnić. Trudność ta jest w literaturze przedmiotu odnotowywana jako jedno z ograniczeń klasyfikacji opartej wyłącznie na wartości wypadkowej [5].

Proces ten jest realizowany po stronie systemu, ponieważ model nie jest w stanie tego dokonać ze względu na brak danych w momencie realizacji oceny. Klasyfikacja ta uzupełnia więc informację, której model w momencie formułowania odpowiedzi nie posiada. Bez niej użytkownik analizujący zagadnienie budzące wyraźny spór otrzymywałby wynik nieodróżnialny od opisu zagadnienia obojętnego.

### 5.3.4 Deterministyczny dobór danych wejściowych

Źródła danych w projekcie mogą zwrócić nawet do kilkunastu pozycji, z których ostatecznie do analizy zostaje wybrane pięć. Istotne jest więc, aby sposób ich wyboru był przejrzysty i powtarzalny, tak aby powtórzone zapytanie opierało się na identycznym materiale.

Rozwiązanie oparte na kolejności pobierania danych doprowadziłoby do wypełnienia limitu tekstów źródłowych elementami pochodzącymi z tego samego miejsca. Wobec tego konieczne jest takie ustalenie kolejności, aby nie faworyzowała ona żadnego ze źródeł. Naturalnym rozwiązaniem wydawałoby się losowe wybieranie elementów, jednak doprowadziłoby to do sytuacji, w której dwa takie same zapytania mogą prowadzić do całkowicie odmiennych wyników. W takim przypadku badanie porównawcze dla dwóch lub więcej modeli byłoby obciążone dużym błędem i brakiem rozróżnialności pomiędzy zmianami spowodowanymi funkcjonowaniem modelu a samym tekstem źródłowym.

Przyjęto więc rozwiązanie zachowujące właściwości statystyczne losowania wartości przy możliwości zachowania powtarzalności wyniku. Kolejność została ustalona poprzez wykorzystanie generatora liczb pseudolosowych, co zapewnia odtwarzalność zapytania. Umożliwia to testowanie odpowiedzi przez różne modele z zachowaniem pełnej przejrzystości danych wejściowych. Sposób wyznaczania kolejności widoczny jest na listingu 13.

Listing 13. Wyznaczanie stanu początkowego generatora

```
export function deriveSeed(
  baseSeed: number,
  query: string,
  sources: readonly string[]
): number {
  const canonicalSources = [...sources]
    .map(source => source.toLowerCase())
    .sort()
    .join(', ');
  return hashSeed(`${baseSeed}:${query}:${canonicalSources}`);
}
```

Nazwy źródeł podlegają wcześniej uporządkowaniu, ponieważ znaczenie ma sam ich zbiór, nie zaś kolejność wskazania. Użytkownik zaznaczający te same źródła w odmiennej kolejności formułuje bowiem identyczne zapytanie i powinien otrzymać identyczny materiał wejściowy. Wartość bazowa pochodzi z konfiguracji systemu, co pozwala na manipulowanie doбором danych i jego testowanie. Konsekwencje podjętych decyzji widoczne są w tabeli 3.

Tabela 3. Zależność doboru pozycji od parametrów zapytania

| Sytuacja                                         | Stan początkowy | Dobór pozycji |
|--------------------------------------------------|-----------------|---------------|
| Powtórzenie tego samego zapytania                | identyczny      | identyczny    |
| Ten sam zbiór źródeł wskazany w innej kolejności | identyczny      | identyczny    |
| Inne zapytanie przy tych samych źródłach         | odmienny        | odmienny      |

|                                           |            |            |
|-------------------------------------------|------------|------------|
| Wszystkie modele w obrębie jednej analizy | identyczny | identyczny |
|-------------------------------------------|------------|------------|

Tabela 3 pokazuje, że stan początkowy oraz dobór pozycji pozostają identyczne zawsze wtedy, gdy zapytanie i zbiór źródeł są takie same, natomiast zmiana zapytania prowadzi do odmiennego doboru. Stan początkowy zależy od zapytania, nie zaś od dostawcy, wobec czego wszystkie modele biorące udział w danej analizie otrzymują dokładnie ten sam materiał. Informacja o dokładnym stanie danych wejściowych zostaje zapisana w bazie z każdym zapytaniem, co umożliwia weryfikowalność uzyskanych wyników, istotną w realizacji badań opisanych w rozdziale 6.

## 5.4 Przebieg analizy i utrwalanie wyników

W poprzednich podrozdziałach omówiono elementy uczestniczące w analizie z osobna: kontrakty opisujące dostawców, reguły domenowe oraz sposób ustalania materiału wejściowego. Niniejszy podrozdział przedstawia sposób ich powiązania, czyli kolejność i współbieżność wywołań, postępowanie w razie niedostępności części usług zewnętrznych oraz warunki utrwalenia uzyskanych wyników.

### 5.4.1 Koordynacja wywołań i izolacja niepowodzeń

Jedno zapytanie wygenerowane przez użytkownika prowadzi do nawet ośmiu wywołań usług zewnętrznych, w tym pięciu modeli językowych oraz trzech źródeł danych. Ze względu na naturę rozwiązań programistycznych każde z nich może zakończyć się niepowodzeniem, zarówno z powodu awarii aplikacji, jak i usług zewnętrznych, na co serwer nie ma żadnego wpływu.

Jednym z rozwiązań może być przeprowadzanie operacji wymiany danych oraz zapytań jedno po drugim. Wykonywanie zapytań sekwencyjnie prowadzi do dwóch następstw:

- czas oczekiwania stanowiłby sumę czasów wszystkich wywołań, podczas gdy odpowiedzi modeli językowych uzyskiwane są w czasie rzędu kilkunastu sekund;
- niepowodzenie któregośkolwiek z nich przerywałoby cały przebieg, przykładowo wyczerpanie limitu zapytań u jednego dostawcy unieważniałoby wyniki uzyskane od pozostałych, mimo że zostały już uzyskane i opłacone.

Realizacja zapytań równolegle wraz z obsługą niepowodzenia umieszczoną wewnątrz mechanizmu komunikacji z API umożliwia rozwiązanie wcześniej opisanych ograniczeń. Rozwiązanie to widoczne jest na listingu 14 i ukazuje istotny element, jakim jest zwracanie wartości zawsze, niezależnie od powodzenia wykonania zapytania. Właśnie to umożliwia realizację wszelkich zadanych zapytań, nawet jeśli nie każde zakończy się powodzeniem.

Listing 14. Koordynacja wywołań źródeł danych

```
const settled = await Promise.all(
  sources.map(async (source) => {
    try {
      const items = await source.fetch(query);
      return [source.name, items, null, null];
    } catch (error) {
      const message = describeExternalFailure(error,
        source.name, timeoutMs);
      return [source.name, [], message,
        classifyExternalError(error)];
    }
  })
);
```

})  
);

Analogiczne rozwiązanie zostało zastosowane przy realizacji wywołań modeli językowych. W obu przypadkach zwracany jest wykaz obejmujący pozycje zwrócone poprawnie oraz niepowodzenia wraz z ich rodzajami, omówionymi szerzej w podrozdziale 5.5.2. Informacja o błędach nie jest pomijana, ale przekazywana do warstwy prezentacji obok poprawnie wygenerowanych danych. Jest to istotne, ponieważ brak wartości wskazującej błąd mógłby prowadzić do błędnego przekonania o poprawności wykonanej analizy.

Rekord zapisywany po zakończeniu analizy odnotowuje źródła, które faktycznie dostarczyły treści, nie zaś te, o które wystąpiono. Rozróżnienie to ma znaczenie dla rzetelności zgromadzonych danych: rekord opisuje materiał, na którym powstała ocena, a nie zakres pierwotnego żądania.

Całość operacji kończy się niepowodzeniem dopiero w momencie, kiedy żadne źródło nie dostarczy jakichkolwiek danych lub ani jeden model nie zwróci odpowiedzi. Nie został przyjęty żaden minimalny próg błędnych wykonania, który oznaczałby analizę jako błędną.

Odrębnego rozstrzygnięcia wymagała odpowiedź uzyskana, lecz niepełna. Sytuacja, w której model odpowiedział, ale części pól nie udało się odczytać, nie jest traktowana jako niepowodzenie wywołania. Rekord trafia do wyników wraz z oznaczeniem opisanym w podrozdziale 5.3.2, ponieważ okoliczność ta stanowi obserwację badawczą cenną w rozdziale 6.

#### 5.4.2 Utrwalanie wyników

Zakończona analiza dostarcza zestaw rekordów, gdzie każdy odpowiada zwrotowi z jednego z modeli. Każdy z nich budowany jest na podstawie tych samych danych wejściowych i ma sens wyłącznie rozpatrywany łącznie z pozostałymi, ponieważ przedmiotem badania jest różnica między nimi. Sama postać rekordu została powielona ze względu na zróżnicowane role pełnione w systemie. Opisany jest on w aplikacji w trzech miejscach:

- w definicji typu współdzielonego przez obie warstwy – działającej na etapie kompilacji;
- w schemacie kolekcji bazy danych – działającym na etapie wykonania;
- w schemacie sprawdzającym poprawność danych przychodzących – działającym na etapie wykonania.

Granice wartości liczbowych pochodzą we wszystkich trzech z definicji skali omówionej w podrozdziale 5.1.1. Rekord obejmuje, poza samą oceną i klasyfikacją wydźwięku, dane pozwalające odtworzyć warunki jego powstania:

- identyfikator grupujący wyniki wszystkich modeli w obrębie jednej analizy;
- treść zapytania oraz wykaz źródeł, które dostarczyły materiał;
- oznaczenie modelu wraz z wersją podaną przez dostawcę;
- wartość wyznaczającą dobór pozycji, opisaną w podrozdziale 5.3.4;
- oznaczenie pól, których nie udało się odczytać z odpowiedzi modelu.

Pusta wartość jest dopuszczalna w polach opisujących ocenę, co wynika bezpośrednio z rozstrzygnięcia przedstawionego w podrozdziale 5.3.2. Bez tego rekord z polem nieodczytanym zostałby odrzucony na etapie zapisu, a rozróżnienie między brakiem odczytu a oceną neutralną przestałoby obowiązywać w warstwie danych.

Rozstrzygnięcie sposobu zapisu jest o tyle istotne, że moment, w którym operacja zostanie przerwana, ma duże znaczenie z perspektywy procesu badawczego. O ile użytkownik

zobaczyłby część z wykonanych analiz i nie stanowiłoby to dla niego dużych konsekwencji, to w procesie badawczym taki skutek jest nie do zaakceptowania. Zestawienia opisane w podrozdziale 5.4.3 obejmują grupy zawierające co najmniej dwa wyniki, przy czym grupa niekompletna spełnia ten warunek i nie różni się niczym od analizy wykonanej mniejszą liczbą modeli. Jednak zgodność wyznaczona na bazie różnej liczby modeli niesie ze sobą niepełne i przekłamane informacje. Zapis wykonywany jest zatem jako operacja niepodzielna, przedstawiona na listingu 15. Utrwalone zostają wszystkie rekordy albo żaden.

Listing 15. Niepodzielny zapis wyników analizy

```
export async function saveAnalyses(
  userId: string,
  records: readonly AnalysisRecord[]
): Promise<string[]> {
  const session = await mongoose.startSession();
  let ids: string[] = [];
  try {
    await session.withTransaction(async () => {
      const saved = await Analysis.insertMany(
        recordsWithUserId, { session, ordered: true }
      );
      ids = saved.map(record => String(record._id));
    });
  } finally {
    await session.endSession();
  }
  return ids;
}
```

Numery identyfikujące rekordy są wyznaczane w trakcie trwania operacji zapisu, nie zaś po jej zakończeniu, ponieważ mechanizm transakcji może powtórzyć wykonanie w razie przejściowego niepowodzenia. Opisane miejsce jest jedynym wymagającym transakcyjności, co stanowi ograniczenie wdrożeniowe. Wymogiem jest, aby baza danych pracowała w konfiguracji umożliwiającej obsługę transakcji.

### 5.4.3 Statystyki liczone po stronie bazy danych

Aby zrealizować zestawienia porównawcze opisane szerzej w rozdziale 6, konieczne jest wyznaczenie wielkości opisujących gromadzone wyniki, takich jak średnie oceny, odchylenia w obrębie pojedynczego zapytania oraz różnice między poszczególnymi parami modeli. Wartości te nie są przechowywane fizycznie w bazie, zamiast tego możliwe jest ich wyznaczenie na żądanie na bazie rekordów zapisanych w sposób opisany w podrozdziale 5.4.2.

Rozwiązanie bazujące na pobraniu wszystkich rekordów, a następnie przetworzeniu ich w pamięci procesu, jest proste w realizacji, ale coraz mniej efektywne wraz ze wzrostem liczby pozycji w bazie. Dodatkowo operacje statystyczne wymagałyby oddzielnej implementacji, podczas gdy są one dostarczane przez samą bazę. Przyjęto zatem podział, w którym wyznaczanie wielkości statystycznych realizuje baza danych, a warstwa domenowa odpowiada za nadanie im znaczenia. Do bazy należy:

- grupowanie rekordów;
- wyznaczanie średnich oraz odchyłeń;
- wyznaczanie wartości skrajnych.

Natomiast system rozstrzyga operacje nieskupiające się na obliczeniach, takie jak:

- przeliczenie odchylenia na miarę zgodności;

- progi rozdzielające zgodność wysoką od niskiej;
- kształt zestawienia porównawczego.

Zanim jednak dojdzie do wyznaczania statystyki, konieczne jest odrzucenie niepełnych rekordów, w których nieodczytana została ocena. Listing 16 obrazuje warunek będący następstwem założeń opisanych w podrozdziale 5.3.2, gdzie oddziela się nieodczytane pozycje od tych, których ocena była neutralna.

Listing 16. Warunek kwalifikujący rekordy do zestawień

```
const AGGREGATION_MATCH: PipelineStage.Match = {
  $match: {
    parseError: { $ne: true },
    score: { $ne: null }
  }
};
```

Pierwsze zestawienie grupuje rekordy zgodnie z identyfikatorem analizy, zwracając średnią uzyskanych ocen oraz ich odchylenie standardowe. Poza wielkościami zbiorczymi przekazywane są również wartości ocen poszczególnych modeli, ponieważ bez nich informacja o wartości odchylenia jest niepełna. Zwracane są tylko zapytania bazujące na więcej niż jednym modelem, co wynika z charakterystyki porównawczej.

Drugie z zestawień wyznacza średnią różnicę ocen dla każdej pary modeli. W obrębie każdej analizy budowane są wszystkie możliwe zestawienia dwóch ocen wraz z ich różnicą bezwzględną, po czym pary modelu z samym sobą zostają odrzucone, a pozostałe zgrupowane i uśrednione. Wraz z wynikiem odnotowywana jest liczba analiz stanowiących jego podstawę, ponieważ zgodność wyznaczona z kilku obserwacji nie ma takiej samej wagi jak ta obliczona z kilkuset, a bez tej informacji obie wyglądałyby jednakowo.

Zestawienie porównawcze tworzone jest na bazie rejestru opisanego szerzej w podrozdziale 5.2.2. W związku z tym modele, dla których nie zgromadzono jeszcze danych, zamiast być pomijane, otrzymują wartości zerowe. Dzięki temu brak danych pozostaje widoczny w zestawieniu, zamiast objawiać się nieobecnością pozycji.

Obliczenia wykonywane po stronie bazy danych mają postać wieloetapowych przekształceń, których wynik trudno ocenić przez samo odczytanie zapisu. W związku z tym zwracane wartości są porównywane z tymi wyznaczonymi przez warstwę domenową na bazie analogicznych obliczeń, uruchamianych na zestawie kontrolnym. Możliwość takiego porównania wynika z własności omówionej w podrozdziale 5.1.2, czyli niezależności warstwy domenowej od infrastruktury.

## 5.5 Bezpieczeństwo i odporność

Poprzednie podrozdziały skupiały się na przebiegu analizy krok po kroku w systemie przy założeniu, że dane wejściowe są poprawne, a usługi zewnętrzne dostępne. Niniejszy podrozdział omawia reakcję systemu na sytuację, w której takie założenia nie zostają spełnione.

### 5.5.1 Granica żądania

System przyjmuje dane pochodzące z zewnątrz, nad którymi nie posiada jakiegokolwiek kontroli. Dotyczy to również danych wysyłanych na serwer przez aplikację kliencką wchodzącą w skład projektu. Żądania takie mogą być modyfikowane w trakcie ich przesyłania, nawet poza przeglądarką. Konieczne jest więc jednoznaczne ustalenie miejsca, w którym

niezweryfikowane dane stają się częścią systemu. Rozwiązanie bazujące na samym typowaniu w języku TypeScript jest jednak niewystarczające i nie zapewnia odpowiedniej ochrony w czasie wykonania. Przyjęcie ciała żądania z adnotacją typu bez faktycznego sprawdzenia oznacza wobec tego, że system opiera się na deklaracji, której nikt nie zweryfikował.

W celu zabezpieczenia aplikacji przed niepożądanymi danymi zaimplementowana została warstwa pośrednia, nazywana inaczej *middleware*, widoczna na listingu 17. Jej celem jest sprawdzenie, czy żądanie ma oczekiwaną przez aplikację strukturę.

Listing 17. Warstwa pośrednia sprawdzająca poprawność danych

```
export function validateBody(schema: ZodTypeAny): RequestHandler {
  return (req, res, next) => {
    const result = schema.safeParse(req.body);
    if (!result.success) {
      const issues = result.error.issues.map(
        issue => formatIssue(issue, '(body)')
      );
      res.status(400).json({
        error: 'Nieprawidłowe dane żądania',
        details: issues
      });
      return;
    }
    req.body = result.data;
    next();
  };
}
```

Należy jednak zaznaczyć, że sprawdzana jest wyłącznie postać danych, a nie ich faktyczna merytoryczna poprawność. Schemat sprawdza, czy dane są przekazywane w odpowiednim formacie, a nie czy pochodzą od właściwego dostawcy. Nazwy dostawców zapisane są w rejestrze omawianym szerzej w podrozdziale 5.2.2, a przechowywanie ich również w schemacie znosiłoby korzyści opisane w podrozdziale 5.2.3.

Schematy odrzucają ponadto żądania zawierające pola nieprzewidziane. Rozwiązanie takie ma znaczenie przy żądaniach wewnętrznie sprzecznych, gdzie wskazanie modelu w adresie punktu końcowego przy jednoczesnym podaniu innego modelu w treści żądania zostaje odrzucone, zamiast zostać po cichu rozstrzygnięte na korzyść jednego z nich.

Aby uzyskać dostęp do funkcji serwera, konieczny jest token przekazywany w nagłówku żądania. Token podlega weryfikacji podpisem wyznaczonym z klucza znanego wyłącznie warstwie serwerowej. Dodatkowo wykorzystywana jest osobna warstwa pośrednia rozstrzygająca o uprawnieniach, stosowana przy punktach końcowych przeznaczonych dla administratora. Odminną rolę pełni rozgałęzienie w module służącym do pobierania historii analiz, gdzie w zależności od posiadanych przez użytkownika uprawnień żądanie zostaje obsłużone w pełni bądź odpowiednio ograniczone.

Punkty końcowe odpowiadające za uwierzytelnianie zostały dodatkowo objęte limitem liczby żądań w oknie czasowym. Każdy z nich posiada dedykowany licznik, który w momencie przekroczenia uniemożliwi odpowiednio proces rejestracji bądź logowania. Liczniki utrzymywane są w pamięci procesu, co przy uruchomieniu wielu instancji systemu proporcjonalnie osłabia skuteczność zabezpieczenia.

Celem rozstrzygania błędów w systemie stworzony został centralny mechanizm określający postać odpowiedzi. Wyróżnić można zgłoszenia o dwóch typach:

- zgłoszenie celowe – niesie ze sobą dedykowany komunikat wyświetlany użytkownikowi, przykładowo nieudane logowanie;
- zgłoszenie ogólne – element systemu obsługujący wszelkie nieprzewidziane zdarzenia, gdzie szczegółowa informacja trafia do dziennika systemu.

### 5.5.2 Rozpoznawanie niedostępności usług zewnętrznych

W podrozdziale 5.4.1 opisany został sposób izolowania niepowodzeń pojedynczego wywołania. Poza opisanym tam postępowaniem konieczne jest skategoryzowanie i rozpoznanie dokładnego powodu błędu. Aplikacja została zintegrowana z ośmioma serwisami zewnętrznymi, korzystającymi z różnych, często dedykowanych bibliotek. Skutkuje to zróżnicowanymi okolicznościami zwracania podobnych błędów w różnej postaci, w zależności od rozwiązania. Obsługa tych elementów w każdej integracji z osobna wiązałaby się z wielokrotnym powielaniem tej samej wiedzy.

W celu rozwiązania tego problemu zastosowano mechanizm polegający na sprowadzeniu wszystkich błędów do skończonego zbioru rodzajów wraz z przypisaniem ich do odpowiednich kategorii, zestawionych w tabeli 4.

Tabela 4. Rodzaje niedostępności usług zewnętrznych

| Rodzaj                | Okoliczność                                 | Ponowienie               |
|-----------------------|---------------------------------------------|--------------------------|
| Przekroczenie czasu   | brak odpowiedzi w wyznaczonym limicie       | zależnie od zastosowania |
| Wyczerpanie limitu    | przekroczona dopuszczalna liczba żądań      | tak, po odczekaniu       |
| Błąd uwierzytelnienia | nieprawidłowy lub wygasły klucz dostępu     | nie                      |
| Awaria usługi         | niepowodzenie po stronie dostawcy           | tak                      |
| Błąd żądania          | żądanie odrzucone jako nieprawidłowe        | nie                      |
| Nieoczekiwana postać  | odpowiedź niezgodna z oczekiwanym kształtem | nie                      |

Ze względu na różnorodne zachowanie serwisów oraz fakt, że niektórzy dostawcy posługują się takimi samymi kodami błędów dla sytuacji, w których system chce podjąć inne działanie, kategoryzacja jest utrudniona. W związku z tym rozstrzygnięcie opiera się na odczytywaniu zawartości błędu, co czyni je nieidealnym. Dodatkowo niektóre zgłoszenia zwracane przez serwisy są zagnieżdżone, co utrudnia ich odczytanie i określenie, co dokładnie spowodowało błąd. Dlatego konieczne jest przeszukanie całego łańcucha celem odnalezienia pierwotnego kodu i rodzaju niepowodzenia. Jeżeli niepowodzenie spełnia określone warunki, takie jak wyczerpanie limitu czy awaria po stronie dostawcy, system podejmie dodatkową próbę uzyskania danych. Jeżeli jednak błąd jest definitywny, a kolejna próba nie miałaby sensu, informacja o nim przekazywana jest dalej.

Jedynym błędem wymagającym odrębnego rozstrzygnięcia jest przekroczenie limitu czasowego na odpowiedź. Dzieje się tak, ponieważ w zależności od okoliczności wywołania oczekiwane jest inne działanie. W przypadku obsługi żądania pochodzącego od aplikacji nie jest ono ponawiane celem oszczędności czasu, ze względu na osobę czekającą na wynik. Jednak w przypadku narzędzia badawczego opisanego w podrozdziale 5.6 przyjęte zostało rozwiązanie przeciwne, ze względu na brak operatora. Ta sama warstwa domenowa obsługiwana jest zatem dwiema odmiennymi zasadami postępowania, dobranymi do sposobu użycia.

## 5.6 Moduł badawczy

Rezultaty przedstawione w rozdziale 6 wymagają zgromadzenia wyników dla kilkuset zagadnień. Następnie przeanalizowanych przez każdy model za pomocą wywołania umożliwiającego ich porównanie. Aplikacja, która została opisana w poprzednich podrozdziałach nie nadaje się do tego celu. Ze względu na sposób realizacji pozyskania treści i ich analizę. Niniejszy podrozdział przedstawia odrębne narzędzie realizujące założenia metodyczne opisane w podrozdziale 3.5.

### 5.6.1 Rozdzielenie pozyskiwania i analizy

Narzędzie do przeprowadzania analizy zostało podzielone na dwa niezależnie uruchamiane etapy. Pierwszy z nich służy do pobierania treści do analizy ze źródeł na bazie skomponowanych kategorii i tematów. Drugi na bazie tych wyników przeprowadza analizę za pomocą wskazanego modelu językowego. Rozwiązanie to zostało przyjęte ze względu na ograniczenia czasowe i ilościowe narzucane przez serwisy zewnętrzne. Dzięki takiemu podziałowi etapy mogą być uruchamiane w szerszej przestrzeni czasowej bez narażania wyników na zmiany w materiale źródłowym. Podejście takie wyklucza różnorodność ocen ze względu na inne dane wejściowe dla analizy, ponieważ ta sama fraza może zwrócić inne teksty źródłowe. Rozwiązanie to niesie ze sobą dwie korzyści praktyczne. Koszt pozyskiwania treści ponoszony jest jednokrotnie, niezależnie od liczby analiz wykonywanych na tym materiale. Ponadto utrwalony materiał pozostaje dostępny po zakończeniu badania, dzięki czemu dołączenie kolejnego modelu i zestawienie go z pozostałymi na identycznych danych możliwe jest również w przyszłości.

Zbiory zagadnień przechowywane są jako pliki tekstowe, a nie rekordy w bazie danych. Rozstrzygnięcie to spowodowane jest koniecznością przytaczania ich w opisie badania oraz porównywalności pomiędzy wersjami. Poprawność zbioru, weryfikująca, czy jest on kompletny i nie posiada powtórzeń, sprawdzana jest przed rozpoczęciem pozyskiwania danych na potrzeby badania. Aby dane zagadnienie było rozpatrywane w trakcie badania, musi być ono pełne, co oznacza, że wszystkie trzy źródła zwróciły odpowiedź. W przeciwnym przypadku nie zostaje ono oznaczone jako pozyskane i zostanie ponownie pobrane podczas kolejnego uruchomienia.

Podczas każdego zapisu wyniku poza treścią utrwalany jest również kontekst, w ramach którego doszło do analizy:

- wersja modelu nadana przez dostawcę;
- wartości parametrów wywołania;
- skrót kryptograficzny instrukcji przekazanej modelowi.

Ten ostatni wyznaczany jest na ustalonym tekście zastępczym w miejsce analizowanych treści, dzięki czemu zależy od struktury instrukcji, nie zaś od materiału konkretnej analizy. Sposób jego wyznaczania przedstawia listing 18. Dzięki takiemu zapisowi możliwe jest zweryfikowanie, czy wyniki powstały w dokładnie takich samych warunkach. Jest to istotna informacja, która umożliwia stwierdzenie, czy przeprowadzone badanie nie zawiera przekłamań.

Listing 18. Wyznaczanie skrótu instrukcji

```
export function computePromptHash(): string {
  const template = buildUserPrompt(['__TEMPLATE_PROBE__']);
  return createHash('sha256')
    .update(SYSTEM_PROMPT)
    .update('\0')
```

```

        .update(template)
        .digest('hex')
        .slice(0, 16);
    }

```

### 5.6.2 Przyrostowość przebiegu i postać wyników

Ze względu na ograniczenia narzucane przez zewnętrzne API, takie jak dobowe limity, pozyskiwanie danych badań trwało zdecydowanie dłużej niż jeden dzień. W związku z tym narzędzie zostało zaprojektowane tak, aby jego praca mogła zostać wznowiona od ostatniego uruchomienia.

Każda faza pomija wcześniej wykonaną pracę. Jeżeli dane zagadnienie posiada już materiał źródłowy lub przeprowadzoną analizę przez dany model, nie jest ono wywoływane ponownie. Tożsamość wyniku określają takie parametry jak:

- zbiór zagadnień;
- model;
- zagadnienie;
- numer powtórzenia.

Oznacza to więc, że wznowienie nie wymaga przekazywania żadnych danych pomiędzy uruchomieniami celem określenia stopnia wykonania kolejnych pozycji.

Zagadnienia zakończone niepowodzeniem nie zostają zapisane, ponieważ rekord z pustymi ocenami przy wznowieniu byłby nieodróżnialny od zagadnienia przetworzonego pomyślnie. W takim przypadku zostaną one oznaczone jako nieprzetworzone i nadrobione w kolejnym uruchomieniu. Istotne jest określenie okoliczności niepowodzenia zapisu i braku danych. Sytuacja, w której źródło zwróci informację o przekroczeniu dobowego limitu, jest zachowaniem spodziewanym, a nie awarią. W takiej sytuacji pobieranie danych źródłowych zatrzymuje się i skrypt przechodzi do przetwarzania treści, które nie posiadają odpowiadających im analiz. Rozstrzygnięcie opiera się na rozpoznaniu rodzaju niedostępności, opisanym w podrozdziale 5.5.2, gdzie wyczerpany limit i nieprawidłowy klucz dostępu wymagają odmiennej reakcji, choć oba uniemożliwiają dalsze pozyskiwanie.

Dodatkowo należy zwrócić uwagę na zróżnicowanie zachowania pomiędzy etapami w kontekście osiągnięcia limitu. W przypadku etapu analizy działanie nie jest przerywane, a zamiast tego przetwarzana jest kolejna pozycja. Dzieje się to ze względu na fakt, że jeden niedziałający model nie uniemożliwia pracy pozostałym. Informacja o błędzie zostanie zaraportowana na koniec pracy skryptu.

W celu weryfikacji dostępności każdego z dostawców rozwiązań na początku pracy sprawdzane jest, czy każda odpytywana pozycja jest dostępna. Weryfikacji podlegają dwie okoliczności rozpatrywane niezależnie: obecność zadeklarowanej wersji modelu w wykazie udostępnianym przez dostawcę oraz powodzenie faktycznego wywołania z przyjętymi parametrami. Rozstrzygające jest przy tym drugie sprawdzenie. Tylko w sytuacji, gdy ten warunek zostanie spełniony, praca z danym modelem może być kontynuowana.

Dane, które zostają zgromadzone podczas uruchomienia, dostępne są w trzech różnych postaciach zestawionych w tabeli 5. Podział ten wynika ze zróżnicowanych zastosowań dla uzyskiwanych danych. Przykładowo analiza statystyczna wymaga układu, w którym każda obserwacja stanowi odrębny wiersz, podczas gdy zestawienia porównawcze wymagają postaci wiążącej wiersz z zagadnieniem, a kolumny z poszczególnymi modelami.

Tabela 5. Postaci udostępniania zgromadzonych danych

| Postać       | Jednostka wiersza               | Zastosowanie                         |
|--------------|---------------------------------|--------------------------------------|
| Długa        | zagadnienie, model, powtórzenie | analiza statystyczna                 |
| Szeroka      | zagadnienie                     | zestawienia porównawcze, korelacje   |
| Strukturalna | zagadnienie, model, powtórzenie | dane zagnieżdżone, w tym perspektywy |

Postać długa zawiera 2550 wierszy, jednak nie odpowiada to liczbie niezależnych obserwacji. Oceny wystawione przez różne modele dla tego samego zagadnienia opierają się na identycznym materiale wejściowym, a ich wzajemna zależność stanowi wręcz przedmiot badania.

## 6. Badanie porównawcze modeli językowych

Rozdział przedstawia proces badawczy oraz jego wyniki, którego przedmiotem jest analiza zbieżności pomiędzy ocenami sentymentu zwracanymi przez różne modele językowe weryfikujące ten sam materiał. Pytanie to zostało postawione w rozdziale 1 jako jeden z dwóch celów pracy. W tym celu na przestrzeni rozdziałów 3 i 5 opisane zostało narzędzie umożliwiające dokonanie tego badania.

Źródłem analizy jest zestaw 510 zagadnień z siedemnastu kategorii tematycznych, które zostały przetworzone przez pięć modeli, co pozwoliło uzyskać 2550 wyników. Każda pozycja została przeanalizowana jednokrotnie, na materiale utrwalonym przed rozpoczęciem badania, dzięki czemu wszystkie modele oceniały identyczne treści.

Pierwszym krokiem jest omówienie metodyki ze szczegółowym opisem schematu eksperymentu, sposobu doboru zagadnień oraz zastosowanych dla jego potrzeb miar. Następnie przybliżono wyniki uzyskane z faktycznego porównania analiz wraz ze sprawdzeniem zgodności zarówno ocen liczbowych, jak i klasyfikacji wydźwięku wypowiedzi. Odrębną część poświęcono wykrywaniu odmiennych perspektyw, będącemu wymiarem różnicującym modele niezależnie od samej oceny. Rozdział zamykają ograniczenia badania oraz wnioski.

### 6.1 Metodyka badania

Wiarygodność porównania modeli zależy od tego, czy zaobserwowane różnice można przypisać samym modelom, a nie okolicznościom, w jakich przeprowadzono pomiar. Podrozdział przedstawia przyjęty schemat badania, sposób doboru materiału oraz miary, na których oparto wnioski.

#### 6.1.1 Schemat eksperymentu i warunki pomiaru

Przeprowadzone badanie miało na celu udzielenie odpowiedzi na trzy pytania badawcze:

- w jakim stopniu modele językowe zgadzają się co do oceny wydźwięku tego samego materiału;
- czy zaobserwowane rozbieżności mają charakter przypadkowy, czy też poszczególne modele wykazują odchylenia systematyczne;
- czy wykrywane są odmienne perspektywy wobec analizowanego zagadnienia i czy modele rozpoznają je w porównywalnym zakresie.

Udzielenie odpowiedzi na te pytania umożliwiło zbudowanie obrazu ukazującego zależności pomiędzy poszczególnymi modelami.

Schemat pomiarów zastosowany na potrzeby analizy miał na celu powiązanie ich pomiędzy sobą. Każdy z modeli ocenił dokładnie ten sam materiał wejściowy, w związku z czym pojedynczy pomiar nie jest niezależny. Okoliczność ta przesądza o doborze metod wnioskowania, omówionym w podrozdziale 6.1.3.

Zmienną analizowaną w badaniu jest ocena wystawiona bezpośrednio przez model w skali od  $-10$  do  $+10$ , przyjęta w postaci nieprzetworzonej. Celowo pominięta została ocena ważona ze względu na jej metodykę wyznaczania. W zależności od liczby perspektyw wykrytych w danej analizie, przy czym odnajdywane są średnio od 2,80 do 3,74 perspektywy, wynik będzie bardziej lub mniej zbliżony do środka skali. W związku z tym ocena taka zawiera więcej informacji w sobie niż sama miara sentymentu, takie jak skłonność do wyodrębniania perspektyw czy nadawanych im wag. Rozbieżność między obiema ocenami wynosi średnio od 0,64 do 1,08 punktu zależnie od modelu, co przy średnich różnicach między modelami rzędu

jednego punktu stanowi wielkość porównywalną z mierzonym efektem. W związku z tym wartości te muszą być weryfikowane odrębnie, tak aby rozdzielić różnicę w ocenianiu od różnicy w wykrywaniu perspektyw, którym poświęcono podrozdział 6.4.

Tabela 6 prezentuje czynniki mogące wpłynąć na wyniki pomiaru niezależnie od zastosowanych modeli.

Tabela 6. Warunki pomiaru

| Czynnik                       | Sposób kontroli                                             |
|-------------------------------|-------------------------------------------------------------|
| Materiał wejściowy            | pozyskany jednokrotnie i utrwalony przed analizą            |
| Zestaw źródeł                 | trzy źródła dla każdego zagadnienia, bez wyjątków           |
| Dobór pozycji z materiału     | wyznaczony z treści zapytania, powtarzalny                  |
| Instrukcja przekazana modelom | jedna dla wszystkich, potwierdzona skrótem kryptograficznym |
| Temperatura generowania       | 0,3 u wszystkich dostawców przyjmujących ten parametr       |
| Liczba powtórzeń              | jedno dla każdego zestawienia zagadnienia z modelem         |

Wskazuje to na konieczność utrwalenia danych wynikowych pochodzących ze źródeł przed wykonaniem badania. Podczas wielokrotnych uruchomień dla tego samego zagadnienia pozycje odnajdywane mogą drastycznie różnić się między sobą, ze względu na ciągłe generowanie nowych zawartości w sieci. Sytuacja taka nałożyłaby dodatkową różnicę w kontekście danych wejściowych, co zakłóciłoby przeprowadzane badanie. Mechanizm realizujący to założenie opisano w podrozdziale 5.6.1. Kolejnymi elementami ujednolicającymi analizę są wspólna instrukcja oraz parametry wywołania. Zgodność instrukcji potwierdza skrót kryptograficzny zapisany wraz z każdym wynikiem, gdzie wszystkie rekordy posiadają dokładnie tę samą wartość.

Problematyczne okazały się dwa zagadnienia. Pierwszym z nich jest brak obsługi parametru temperatury przez część modeli, przy czym każde wywołanie niesie informację o tym, przy jakim ustawieniu było wykonane. Dodatkowo ważne jest odnotowanie faktu, że pomimo niskiego ustawienia modele nie są deterministyczne, w związku z czym jednokrotne wywołanie analizy nie pozwala jednoznacznie oddzielić zmienności modelu od różnic między nimi. Następstwa obu tych ograniczeń omówiono w podrozdziale 6.5.

Dodatkowo z każdym wynikiem zapisywana była dokładna informacja o wersji modelu wykorzystywanej do przeprowadzenia badania. Jest to o tyle istotne, że różne modele pochodzące od tego samego dostawcy mogą zasadniczo różnić się działaniem, wobec czego porównywanie ich jako tożsamych stanowiłoby błędne założenie. Dokładna lista wersji została zamieszczona w tabeli 7.

Tabela 7. Modele objęte badaniem

| Nazwa w systemie | Oznaczenie wersji  | Dostawca  |
|------------------|--------------------|-----------|
| chatgpt          | gpt-5.4-2026-03-05 | OpenAI    |
| claude           | claude-sonnet-4-6  | Anthropic |

|            |                  |            |
|------------|------------------|------------|
| gemini     | gemini-3.6-flash | Google     |
| grok       | grok-4.5         | xAI        |
| perplexity | sonar-pro        | Perplexity |

### 6.1.2 Zbiór zagadnień

Na potrzeby badania opracowany został autorski zbiór 510 zagadnień, rozłożonych równomiernie na siedemnaście kategorii tematycznych, po trzydzieści w każdej. Wykaz kategorii wraz z przykładowymi zagadnieniami przedstawia tabela 8, natomiast pełną listę zamieszczono w dodatku A.

Tabela 8. Kategorie tematyczne objęte badaniem

| Kategoria                    | Liczba | Przykłady                       |
|------------------------------|--------|---------------------------------|
| Elektronika użytkowa         | 30     | iPhone 17, Samsung Galaxy S25   |
| Motoryzacja                  | 30     | Tesla Model Y, Rivian R1S       |
| Serwisy strumieniowe         | 30     | Netflix, Disney Plus            |
| Gry komputerowe              | 30     | Baldur's Gate 3, Elden Ring     |
| Podróże                      | 30     | Ryanair, Wizz Air               |
| Żywność i napoje             | 30     | Oatly, Beyond Meat              |
| Aplikacje finansowe          | 30     | Revolut, Monzo                  |
| Narzędzia programowe         | 30     | Notion AI, Obsidian             |
| Kosmetyki i higiena          | 30     | CeraVe, The Ordinary            |
| Odzież i obuwie              | 30     | Nike Air Max, Adidas Samba      |
| Zdrowie i sprawność          | 30     | Whoop, Peloton                  |
| Handel elektroniczny         | 30     | Amazon, Temu                    |
| Dobra luksusowe              | 30     | Rolex Submariner, Louis Vuitton |
| Artykuły dla zwierząt        | 30     | Purina Pro Plan, Royal Canin    |
| Media społecznościowe        | 30     | TikTok, Instagram Reels         |
| Sprzęt gospodarstwa domowego | 30     | Instant Pot, KitchenAid         |
| Zabawki i artykuły dziecięce | 30     | LEGO, Barbie                    |

Zaprezentowany zestaw obejmuje zagadnienia dotyczące marek, produktów i usług. Jest to działanie celowe podyktowane dwoma głównymi okolicznościami. Zachowana została spójność z obszarem zastosowania narzędzia, opisanym w rozdziale 1. Po drugie zagadnienia pochodzące z obszarów polityki czy światopoglądowych posiadają pewne ograniczenia zaszyte wewnątrz modeli. Bardzo często podczas analizy takiej tematyki modele kierowane są ku nieobieraniu strony i neutralizacji danego zagadnienia, tak aby nie urazić żadnej z grup. W

przypadku analizy przedmiotów takich jak elektronika czy portale społecznościowe nie ma żadnych ograniczeń z tym związanych.

Kategorie dobrano tak, aby objąć obszary różniące się prowadzoną wokół nich dyskusją. Wybrane zostały zagadnienia oceniane przez użytkowników według odmiennych kryteriów, takich jak:

- jakości obsługi;
- przedmioty o silnym ładunku statusowym;
- narzędzia oceniane w kategoriach użyteczności.

Pozwala to na weryfikację zachowania modeli podczas analizy różnych dziedzin oraz tego, czy uzyskiwane wyniki są w związku z tym zbieżne. Każda kategoria zawiera dokładnie taką samą liczbę zagadnień. Rozstrzygnięcie to nie ma znaczenia przy analizie zbiorczej, umożliwia natomiast porównywalność pomiędzy kategoriami.

Wszystkie zagadnienia zostały sformułowane w języku angielskim. Wybór ten został podyktowany przewagą w występowaniu tego języka w sieci. Materiały anglojęzyczne występują w liczbie wystarczającej, aby możliwe było pobranie aktualnych treści dla każdej frazy. Wybór języka polskiego mógłby ograniczyć pulę i uniemożliwić badanie, czego następstwa zostały opisane szerzej w podrozdziale 6.5.

### 6.1.3 Miary zgodności

Pomiar zgodności ocen wystawianych temu samemu materiałowi przez różnych oceniających stanowi zagadnienie rozwijane od kilkadziesiąt lat, pierwotnie na potrzeby badań, w których materiał opisywany był niezależnie przez kilka osób [41]. Przy zadaniach pozbawionych jednoznacznie poprawnej odpowiedzi zgodność wyraża się jako nadwyżkę ponad poziom przypadku, nie zaś jako trafność, ponieważ nie istnieje wzorec, wobec którego oceny mogłyby zostać zweryfikowane. Aparat ten znajduje obecnie zastosowanie również przy porównywaniu modeli językowych występujących w roli oceniających, co stanowi przedmiot odrębnego nurtu badawczego określanego jako *LLM-as-a-judge* [45]. Prowadzone w jego ramach prace wskazują, że spójność ocen zależy w istotnym stopniu od wybranego modelu, co uzasadnia rozpatrywanie każdego z nich osobno, zamiast traktowania ich jako rozwiązań wymiennych.

W ramach rozważanej zbieżności ocen wyróżnić można więcej miar niż tylko liczbowa. Zgodność oceniać można również poprzez bliskość tych ocen czy też przypisywane materiałom etykiety ogólnego wydźwięku. Każda z tych miar odpowiada na inne pytanie i prowadzi do odmiennych wniosków, co wykazano w podrozdziale 6.3.

Podstawowym wyznacznikiem jest odchylenie standardowe ocen wystawionych przez wszystkie modele dla pojedynczego konkretnego zagadnienia. W tym przypadku zero oznacza pełną zgodność, a im wyższa wartość tym rozejście się wyników jest większe. Jest to wskazanie ogólnej zbieżności pomiędzy modelami, nie informuje jednak, który z nich odstaje najbardziej. Dla czytelniejszej prezentacji odchylenie przeliczono na skalę od 0 do 100 według zależności opisanej w podrozdziale 5.4.3, gdzie wartość 100 odpowiada pełnej zgodności.

Średnia różnica bezwzględna pary modeli to miara wyznaczona dla każdej pary modeli informująca, jak bardzo zbieżne są wyniki liczbowe. Umożliwia to weryfikację, które modele oceniają podobnie, a które rozchodzą się najbardziej. Wraz z nią odnotowywana jest liczba zagadnień stanowiących podstawę wyliczenia.

Istotnym zagadnieniem jest zgodność przypisywanych klasyfikacji wydźwięku. Ważne jest tutaj wykluczenie czynnika losowego. Jeżeli jedna z etykiet występuje wyraźnie częściej

od pozostałych, dwa modele mogą przypisać ją niezależnie od siebie, pomimo braku faktycznego związku między ich ocenami. Istotne jest zatem zweryfikowanie, czy spójność w przypisaniu takich oznaczeń wynika z faktycznego pokrycia ocen. W celu weryfikacji tego zastosowano współczynnik kappa Cohena [41], który uwzględnia zgodność oczekiwaną przy niezależnym przypisywaniu etykiet. Wartość zerowa oznacza zbieżność na poziomie przypadku, natomiast jedność — pełne pokrycie. Przyjęto powszechnie stosowaną interpretację przedziałów zaproponowaną przez Landisa i Kocha [42], przedstawioną w tabeli 9.

Tabela 9. Interpretacja wartości współczynnika kappa Cohena

| Przedział    | Poziom zgodności      |
|--------------|-----------------------|
| poniżej 0,00 | brak zgodności        |
| 0,00–0,20    | zgodność nikła        |
| 0,21–0,40    | zgodność słaba        |
| 0,41–0,60    | zgodność umiarkowana  |
| 0,61–0,80    | zgodność znaczna      |
| 0,81–1,00    | zgodność niemal pełna |

Należy jednak mieć na uwadze, że samo wskazanie różnicy pomiędzy średnimi ocenami nie przesądza o tym, czy różnica ta nie wynika z przypadku. Ze względu na schemat pomiarów powiązanych, opisany w podrozdziale 6.1.1, zastosowano metody przewidziane dla prób zależnych.

W celu weryfikacji różnic pomiędzy modelami rozpatrywanymi jako grupa wykorzystany został test Friedmana [43]. Jest to nieparametryczny odpowiednik analizy wariancji dla pomiarów powtarzanych. Dodatkowo celem porównania par modeli zastosowany został test kolejności par Wilcoxona [44]. Pierwszy z testów odpowiada na pytanie, czy modele są równoważne, drugi natomiast weryfikuje, które dokładnie modele różnią się od siebie. Metody nieparametryczne zostały wybrane ze względu na postać oceny. Przyjmuje ona wartości z określonego przedziału, przy rozkładzie odbiegającym od normalnego.

Ponadto zweryfikowana została zdolność modeli do wyodrębniania perspektyw. Opisano to dwiema wielkościami:

- liczbą perspektyw wykrytych dla danego zagadnienia;
- rozbieżnością między oceną wystawioną przez model a oceną wyznaczoną jako średnia ważona perspektyw.

Druga z nich obrazuje, na ile spójne są dwie oceny wygenerowane przez ten sam model dla tego samego materiału.

## 6.2 Charakterystyka wyników poszczególnych modeli

Przed porównaniem zgodności pomiędzy modelami omówiona została charakterystyka wyników uzyskanych przez każdy model osobno. Umożliwia to rozpoznanie indywidualnych właściwości i ułatwia interpretację rozbieżności opisanych w podrozdziale 6.3.

Badanie zostało oparte na zbiorze opisanym w podrozdziale 6.1.2, który okazał się kompletny. Odczyt odpowiedzi powiódł się w przypadku każdego z pięciu modeli, co daje 2550

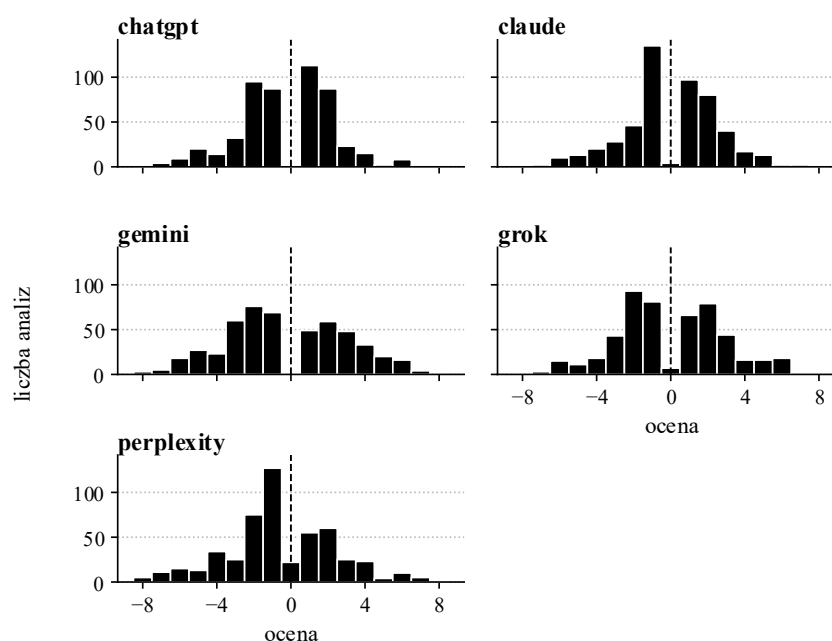
wyników. Wszystkie kategorie posiadają komplet danych, dzięki czemu kwalifikują się do badania. Warto również zaznaczyć, że mechanizm sygnalizowania opisany w podrozdziale 5.3.2 nie wykazał ani jednego niepowodzenia odczytu, co potwierdza skuteczność ujednolicenia instrukcji oraz sposobu odczytu odpowiedzi. Wywołania zakończone przejściowym niepowodzeniem stanowiły zaledwie 2,4% wszystkich pozycji, przy czym 56 z nich wymagało jednego powtórzenia, a jedynie 4 dwóch. Wydarzenia te nie mają wpływu na postać wyników ze względu na mechanizm zamrażania danych wejściowych.

Statystyki opisowe ocen wystawionych przez poszczególne modele przedstawia tabela 10.

Tabela 10. Statystyki opisowe ocen poszczególnych modeli

| Model      | n   | Średnia | Mediana | Odch. std. | Min. | Maks. |
|------------|-----|---------|---------|------------|------|-------|
| chatgpt    | 510 | -0,25   | -1,0    | 2,53       | -7   | 7     |
| claude     | 510 | 0,03    | 0,0     | 2,54       | -7   | 8     |
| gemini     | 510 | -0,25   | -1,0    | 3,29       | -8   | 7     |
| grok       | 510 | -0,05   | -1,0    | 2,86       | -7   | 6     |
| perplexity | 510 | -0,58   | -1,0    | 2,94       | -8   | 8     |

Średnie wszystkich ocen mieszczą się w przedziale od -0,58 do 0,03, przy czym cztery z pięciu modeli mają średnią ujemną. Mediana tych samych czterech modeli wynosi -1, natomiast dla modelu claude 0. Wskazuje to na zbieżność ogólnego poziomu ocen. W podrozdziale 6.3 są one opisane szerzej w oparciu o dane kategorie i zagadnienia. Większą różnicę można dostrzec w wartościach odchylenia standardowego mieszczącego się w przedziale od 2,53 dla chatgpt do 3,29 dla gemini. Rozkład tych ocen zaprezentowany jest na rysunku 8.



Rysunek 8. Rozkłady ocen poszczególnych modeli. Źródło: opracowanie własne

Model gemini charakteryzuje się zwiększoną tendencją do przypisywania wartości skrajnych. Oceny o wartości bezwzględnej co najmniej pięć przypisuje w 18,2% przypadków, podczas gdy claude i chatgpt odpowiednio w 8,4% i 8,6%. Modele grok oraz perplexity plasują się pośrodku, z udziałem 12,4% i 12,5%. Odpowiada to zaobserwowanym różnicom w odchyleniu standardowym i wskazuje nie tylko na różnice w kierunku ocen, ale również w stanowczości ich wyrażania.

Odrębnego omówienia wymaga występowanie ocen zerowych. Instrukcja przekazana modelom, opisana w podrozdziale 5.3.1, ogranicza przypisywanie oceny neutralnej do treści całkowicie pozbawionych zabarwienia emocjonalnego. Wskazanie to okazało się skuteczne, ponieważ chatgpt oraz gemini ani razu nie zastosowały tej wartości, natomiast claude uczynił to czterokrotnie, a grok siedmiokrotnie. Jedynym odstępstwem okazał się model perplexity, który wystawił tę ocenę aż 22 razy, co stanowi 4,3% jego wyników. Zestawienie rozkładów klasyfikacji wydźwięku przedstawia tabela 11.

Tabela 11. Rozkład klasyfikacji wydźwięku [%]

| Model      | mieszany | negatywny | pozytywny | neutralny |
|------------|----------|-----------|-----------|-----------|
| chatgpt    | 86,1     | 9,2       | 4,7       | 0,0       |
| claude     | 81,0     | 10,4      | 6,5       | 2,2       |
| gemini     | 66,3     | 16,9      | 16,9      | 0,0       |
| grok       | 82,5     | 7,6       | 9,8       | 0,0       |
| perplexity | 88,2     | 6,1       | 4,5       | 1,2       |

Widoczna jest wyraźna przewaga etykiety mieszanej, obejmującej od 66,3% do 88,2% wyników. Wynika to z reguły klasyfikacji opisanej w podrozdziale 5.3.3, przypisującej tę klasyfikację zagadnieniom, dla których wykryto perspektywy o przeciwnych znakach przy ocenie bliskiej zeru. Jest to sytuacja typowa dla materiału, w którym skrajne opinie wzajemnie się równoważą. Model gemini wyraźnie odstaje pod tym względem od pozostałych, o piętnaście punktów procentowych wobec kolejnej pozycji. Różnica ta rozkłada się przy tym równomiernie na kategorię pozytywną i negatywną, obie wynoszące 16,9%.

Tak duża przewaga etykiety nad pozostałymi niesie ze sobą konsekwencje w kontekście kolejnych decyzji podejmowanych w badaniu. Dwa modele przypisujące klasyfikację niezależnie od siebie osiągają w takich warunkach wysoką zgodność wyłącznie za sprawą częstości jej występowania, co uzasadnia zastosowanie współczynnika uwzględniającego zgodność przypadkową, omówione w podrozdziale 6.3.2.

## 6.3 Zgodność ocen między modelami

Podrozdział ten skupia się na zasadniczej części badania porównawczego. Opisuje on pomiar stopnia, w jakim modele są ze sobą zbieżne w ocenie tego samego materiału źródłowego. Omówienie obejmuje kolejno zbieżność ocen liczbowych, zgodność przypisanych etykiet wydźwięku oraz zagadnienia, dla których rozbieżność okazała się największa.

### 6.3.1 Zgodność ocen liczbowych

Dla każdego zagadnienia wyznaczone zostało odchylenie standardowe na podstawie pięciu wystawionych mu przez modele ocen. Średnia wartość tego odchylenia wynosi 0,96 punktu, podczas gdy mediana 0,89 przy skali o rozpiętości 20 punktów, od -10 do +10. Osiągnięte wartości skrajne to 0,00, oznaczające pełną zgodność, oraz 3,61.

Po przeliczeniu na skalę od 0 do 100, opisaną szerzej w podrozdziale 6.1.3, daje to średnią 90,4 oraz medianę 91,1. Próg 70, odpowiadający wysokiej zgodności, został przekroczony przez 99,4% zagadnień, natomiast poniżej progu oznaczającego zgodność niską nie znalazł się żaden rekord. Rozkład jest przy tym wąski, co oznacza, że zagadnienia budzące wyraźną rozbieżność stanowią wyjątek, nie zaś regułę. Wartości te wskazują na wysoką zbieżność ocen liczbowych pomiędzy modelami. Należy jednak pamiętać o wynikach przedstawionych w podrozdziale 6.2, gdzie mediana czterech z pięciu modeli wynosiła -1. Sugeruje to, że wysoka zgodność wynika po części z samego skupienia rozkładu ocen wokół środka skali.

Średnie różnice bezwzględne ocen dla każdej z dziesięciu par modeli przedstawia tabela 12.

Tabela 12. Średnia różnica bezwzględna ocen dla par modeli

| Para modeli          | Średnia różnica | Zgodność |
|----------------------|-----------------|----------|
| gemin – grok         | 0,87            | 95,6     |
| chatgpt – grok       | 0,93            | 95,4     |
| chatgpt – claude     | 1,00            | 95,0     |
| claude – grok        | 1,09            | 94,5     |
| chatgpt – gemini     | 1,15            | 94,2     |
| chatgpt – perplexity | 1,28            | 93,6     |
| grok – perplexity    | 1,38            | 93,1     |
| claude – gemini      | 1,40            | 93,0     |
| claude – perplexity  | 1,42            | 92,9     |
| gemin – perplexity   | 1,59            | 92,0     |

Wartości niezgodności mieszczą się w przedziale od 0,87 do 1,59 punktu, w wyniku czego rozpiętość pomiędzy najmniej i najbardziej zgodnymi parami to 0,72. Zwraca uwagę model perplexity, występujący we wszystkich czterech parach o najniższej zgodności, co odpowiada wcześniejszej obserwacji dotyczącej jego tendencji do ocen negatywnych. Największą zbieżnością cechują się modele gemini oraz grok, pomimo wyraźnej dysproporcji w rozproszeniu ocen.

Zestawione średnie różnice oraz zgodność nie rozstrzygają jednak, czy niewspółmierność występująca między modelami nie wynika z przypadku. W tym celu przeprowadzony został test Friedmana, który wykazał, że dysproporcje występujące pomiędzy ocenami wyznaczanymi przez modele nie mieszczą się w granicach zmienności losowej ( $\chi^2 = 62,15$ ;  $p < 0,001$ ). Dowodzi to, że wybór modelu wykorzystanego do analizy danego zagadnienia realnie wpływa na uzyskiwane wyniki. Aby dokładnie wyznaczyć, między którymi modelami różnice te są

widoczne, zastosowano porównanie par za pomocą testu Wilcoxona. Wyniki przedstawiono na rysunku 9 w postaci macierzy, gdzie każda komórka odpowiada jednej parze modeli.

|            | chatgpt | claude  | gemini  | grok    | perplexity |
|------------|---------|---------|---------|---------|------------|
| chatgpt    | —       | < 0,001 | 0,837   | 0,001   | < 0,001    |
| claude     | < 0,001 | —       | < 0,001 | 0,382   | < 0,001    |
| gemini     | 0,837   | < 0,001 | —       | < 0,001 | < 0,001    |
| grok       | 0,001   | 0,382   | < 0,001 | —       | < 0,001    |
| perplexity | < 0,001 | < 0,001 | < 0,001 | < 0,001 | —          |

brak różnicy istotnej  
 różnica istotna ( $p < 0,05$ )

Rysunek 9. Wyniki porównań par modeli testem Wilcoxona. Źródło: opracowanie własne

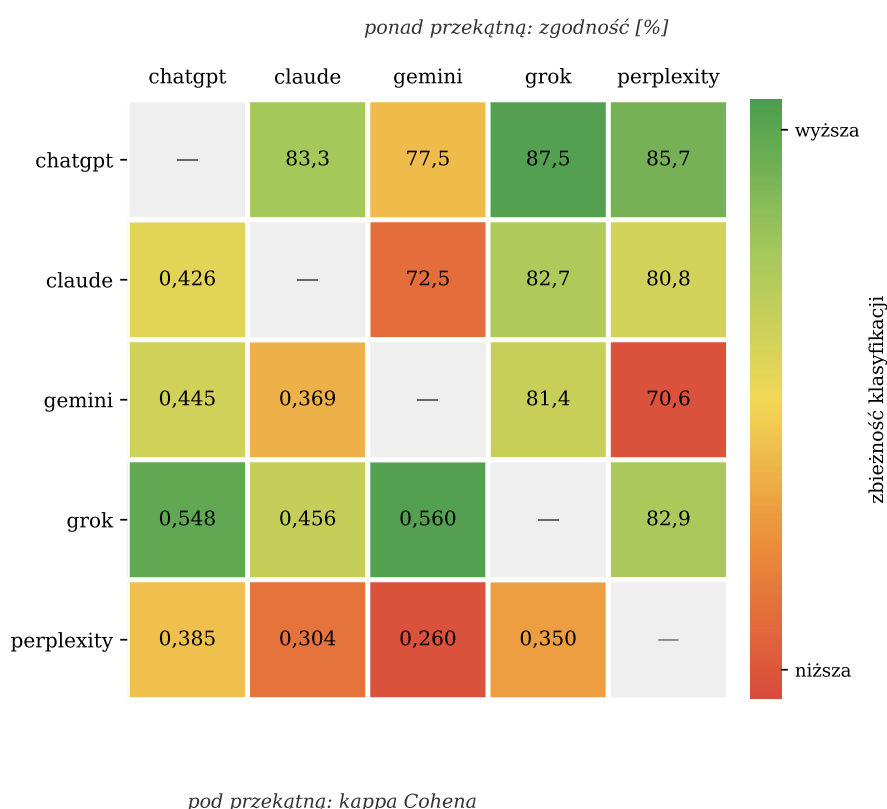
Osiem z dziesięciu par wykazuje różnice istotne statystycznie. Wyjątkiem są pary chatgpt i gemini oraz claude i grok, wyróżnione w macierzy zielonym tłem. Pierwsza z nich zasługuje na odrębne omówienie ze względu na wykazanie ograniczenia w zastosowanej metryce. Modele te cechują się identycznymi średnimi ocen na poziomie  $-0,25$ , podczas gdy ich odchylenia standardowe różnią się o ponad trzydzieści procent ( $2,53$  wobec  $3,29$ ). Test ten bada przesunięcie rozkładu, nie zaś jego rozproszenie, wobec czego nie wykrywa różnicy tego rodzaju. Wskazuje to, że modele oceniają przeciętnie tak samo, jednak oceny te mają wyraźnie odmienny rozrzut, czyli na podobną średnią składają się inne wartości. Spostrzeżenie to uzasadnia posługiwanie się w niniejszym rozdziale kilkoma miarami. Pojedyncza wielkość opisująca zgodność, niezależnie od jej doboru, pomija te wymiary różnicowania, których nie obejmuje.

### 6.3.2 Zgodność klasyfikacji wydźwięku

Zgodność ocen liczbowych opisana w podrozdziale 6.3.1 nie jest wystarczającym wyznacznikiem do określenia zgodności w zakresie przypisanych etykiet wydźwięku. Dzieje się tak, ponieważ wyznaczenie odpowiedniej etykiety opiera się na spełnieniu progu klasyfikacyjnego. Dwie bliskie sobie liczby mogą jednak trafić do dwóch sąsiednich przedziałów, w wyniku czego otrzymają całkowicie odmienne etykiety.

Ta sama etykieta została przypisana 307 zagadnieniom, co stanowi 60,2% całego zbioru. W 198 przypadkach wyznaczone zostały dwie różne etykiety, a zaledwie w pięciu trzy. Żadna pozycja nie doczekała się czterech różnych klasyfikacji. Dane te prezentują obraz odmienny niż wcześniej omawiane wartości liczbowe. W tamtym przypadku wysoki próg zgodności został przekroczony dla 99,4% zagadnień, podczas gdy tutaj jest to niespełna dwie trzecie. Wynika to głównie z nieciągłego charakteru klasyfikacji: różnica między dwiema ocenami może być marginalna, jednak jeżeli przypada ona na granicę progu, obie wartości trafiają do odmiennych kategorii.

Zgodność mieści się w przedziale od 70,6% do 87,5%, co wskazywałoby na wysoką zbieżność modeli. Współczynnik kappa prowadzi jednak do odmiennego wniosku: jego najwyższa wartość wynosi 0,560, najniższa 0,260, przy średniej 0,410. Zestawienie obu miar dla poszczególnych par przedstawia rysunek 10. Rozbieżność ta wynika w dużej mierze z nierównomiernego rozkładu etykiet, co szerzej omówiono w podrozdziale 6.2. Przykładowo para chatgpt i perplexity cechuje się wysokim udziałem etykiety mieszanej, wynoszącym odpowiednio 86,1% oraz 88,2%, wobec czego zgodność oczekiwana przy braku jakiegokolwiek związku między modelami wynosi 0,767. Zaobserwowana wartość 0,857 przekracza ją o 0,090 przy możliwej do osiągnięcia nadwyżce 0,233, co daje współczynnik kappa równy 0,385. Oznacza to, że spośród zgodności możliwej do zdobycia ponad poziom przypadku modele te osiągnęły niespełna dwie piąte.



Rysunek 10. Zbieżność klasyfikacji wydźwięku dla par modeli. Źródło: opracowanie własne

Należy również zwrócić uwagę na rozbieżność w kontekście przyporządkowania par względem samych wartości wskaźników. Para chatgpt i perplexity osiąga zgodność procentową wyższą niż para gemini i grok, 85,7% wobec 81,4%, lecz wyraźnie niższą wartość współczynnika kappa, 0,385 wobec 0,560. Wynika to z wysokiej zgodności przypadkowej, będącej następstwem zbliżonych rozkładów etykiet. Prowadzi to do wniosku, że posługiwanie się wyłącznie odsetkiem zgodności może zawyżyć ocenę zbieżności modeli oraz doprowadzić do błędnego wskazania pary klasyfikującej najbardziej podobnie.

Zestawienie wyników z obecnego podrozdziału oraz poprzedniego buduje niespójny obraz. Modele oceniają treści podobnie w kontekście liczbowym, jednak etykiety przypisywane są na poziomie co najwyżej umiarkowanym. Sprzeczność ta wynika ze sposobu powstawania klasyfikacji. Większość ocen skupia się wokół środka skali, co przedstawiono w podrozdziale

6.2, wobec czego niewielkie różnice liczbowe prowadzą w tym obszarze do przypisania odmiennych etykiet.

### 6.3.3 Przypadki największej rozbieżności

Opisane wcześniej miary skupiają się na opisie zgodności w kontekście zbiorczym. Odrębnego omówienia wymagają zagadnienia, dla których oceny uzyskane od poszczególnych modeli rozeszły się najmocniej. Pozwalają one dostrzec okoliczności, których ujęcie zbiorcze nie obejmuje.

Zestawienie dziesięciu zagadnień o największym odchyleniu standardowym ocen przedstawia tabela 13.

Tabela 13. Zagadnienia o największej rozbieżności ocen

| Zagadnienie   | chatgpt | claude | gemini | grok | perplexity | Odch. std. |
|---------------|---------|--------|--------|------|------------|------------|
| Kindle Scribe | -2      | 1      | 3      | 2    | -7         | 3,61       |
| Curve card    | -1      | 8      | -1     | 0    | -1         | 3,52       |
| Pinterest     | -1      | 7      | 1      | 1    | 6          | 3,12       |
| Postman       | 2       | -6     | 1      | -1   | -1         | 2,76       |
| Burberry      | 6       | -1     | 6      | 6    | 3          | 2,76       |
| Hermès Birkin | 1       | 7      | 2      | 1    | -1         | 2,68       |
| Dior          | 3       | 5      | 6      | 6    | -1         | 2,64       |
| L'Oréal Paris | 4       | 5      | 6      | 5    | -1         | 2,48       |
| Eurostar      | -4      | -3     | 2      | 1    | -2         | 2,32       |
| Klarna        | -2      | 3      | -4     | -2   | -1         | 2,32       |

Różnice w ocenach sięgają w skrajnych sytuacjach dziesięciu punktów, co stanowi połowę skali. Przykładem może być zagadnienie „Kindle Scribe”, gdzie model gemini wystawił ocenę +3, podczas gdy perplexity -7. Analogiczna sytuacja wystąpiła w przypadku zagadnienia „Curve card”, gdzie model claude wystawił ocenę +8, podczas gdy pozostałe modele mieszczą się w przedziale od -1 do 0. Należy jednak zauważyć, że różnice te mają charakter odosobniony. Pozycji, w których odchylenie przekracza wartość 2,5 punktu, jest zaledwie siedem na 510, co stanowi 1,4% zbioru. Nie jest to zatem wystarczający argument do podważenia wcześniej wskazanej wysokiej zbieżności.

Analiza pięćdziesięciu zagadnień o najwyższym rozproszeniu ocen wykazała, że modele odpowiadają za nie w nierównym stopniu. Zestawienie liczby przypadków, w których dany model wystawił ocenę skrajną, przedstawia tabela 14.

Tabela 14. Udział modeli w rozbieżnościach (50 zagadnień o najwyższym rozproszeniu)

| Model  | Ocena najwyższa | Ocena najniższa | Razem |
|--------|-----------------|-----------------|-------|
| gemini | 18              | 8               | 26    |

|            |    |    |    |
|------------|----|----|----|
| claude     | 15 | 9  | 24 |
| chatgpt    | 12 | 7  | 19 |
| perplexity | 3  | 26 | 29 |
| grok       | 2  | 0  | 2  |

Zaprezentowane dane ujawniają dwie tendencje. Model perplexity odpowiada za skrajność w 29 przypadkach, przy czym w 26 z nich wystawił ocenę najniższą, co pokrywa się z wcześniejszymi spostrzeżeniami. Z drugiej strony znajduje się model grok, który jedynie dwukrotnie znalazł się na krańcu rozkładu, przy czym ani razu nie wystawił oceny najniższej. Wskazuje to, że spośród badanych modeli najkonsekwentniej zajmuje on położenie pośrednie.

Zagadnienia cechujące się najwyższym rozproszeniem ocen nie są rozłożone równomiernie po kategoriach tematycznych. Wśród badanej puli najliczniej reprezentowane były zagadnienia dotyczące dóbr luksusowych, narzędzi programowych oraz gier. Największym skupieniem ocen cechują się natomiast motoryzacja i elektronika. Tendencje te wynikają w dużej mierze z charakteru dyskusji, która jest prowadzona wokół tych zagadnień. Sprzęt opisywany jest za pomocą właściwości podanych w specyfikacji, które nie budzą kontrowersji i dają mniejsze pole do ocen skrajnych. Dobra luksusowe posiadają natomiast zarówno wielu zwolenników doceniających jakość wykonania, jak i krytyków konsumpcjonizmu.

## 6.4 Wykrywanie perspektyw

Wartości ocen analizowane w podrozdziale 6.3 obejmowały miarę zwracaną bezpośrednio przez model. Jednak rozwiązanie opisane w podrozdziale 3.4.2 wskazuje, że schemat analizy obejmuje wyodrębnianie perspektyw wraz z ich oceną. Niniejszy podrozdział omawia tę funkcjonalność jako dodatkowy wymiar różnicujący wyniki uzyskiwane w analizie sentymentu.

Zestawienie liczby stanowisk wyodrębnionych przez poszczególne modele przedstawia tabela 15.

Tabela 15. Liczba perspektyw wykrywanych przez modele

| Model      | Średnia | Mediana | Odch. std. | Maks. |
|------------|---------|---------|------------|-------|
| claude     | 3,74    | 4       | 0,69       | 6     |
| grok       | 3,49    | 4       | 0,73       | 6     |
| chatgpt    | 3,45    | 4       | 0,61       | 4     |
| gemini     | 3,10    | 3       | 0,53       | 4     |
| perplexity | 2,80    | 3       | 0,80       | 5     |

Widoczna jest dysproporcja pomiędzy poszczególnymi modelami. Claude wyodrębnia średnio 3,74 punktów widzenia, podczas gdy perplexity jedynie 2,80. Jest to różnica sięgająca blisko jednej perspektywy na zapytanie, przy zachowaniu identycznego materiału źródłowego.

Rozkłady przedstawione w tabeli 16 pokazują, że nie jest to przesunięcie równomierne, lecz odmienna skłonność do rozdrabniania opinii.

Tabela 16. Rozkład liczby wykrytych perspektyw [%]

| Model      | 0   | 1   | 2    | 3    | 4    | 5 lub więcej |
|------------|-----|-----|------|------|------|--------------|
| chatgpt    | 0,0 | 0,2 | 5,3  | 43,5 | 51,0 | 0,0          |
| claude     | 1,2 | 0,0 | 2,0  | 22,5 | 69,8 | 4,5          |
| gemini     | 0,2 | 1,0 | 5,3  | 75,7 | 17,8 | 0,0          |
| grok       | 1,6 | 0,2 | 1,8  | 42,9 | 51,8 | 1,8          |
| perplexity | 2,5 | 0,6 | 25,5 | 58,0 | 12,0 | 1,4          |

Ponownie wyróżnia się model claude, który w 69,8% analiz wskazuje cztery perspektywy i jako jedyny przekracza tę wartość w zauważalnym zakresie. Gemini najczęściej poprzestaje na trzech, czyniąc tak w 75,7% zagadnień. Model perplexity ogranicza się natomiast do dwóch w 25,5% analiz, podczas gdy kolejny model w tej kategorii zaledwie w 5,3%. Zgodność pomiędzy modelami jest znikoma, gdyż tylko dla 6,5% zagadnień wskazały one dokładnie tę samą liczbę. Przeciętna rozpiętość wynosi 1,42, co świadczy o różnicach zdecydowanie większych niż przy samej ocenie sentymentu. Warto wskazać, że instrukcja przekazywana modelom wyraźnie zaleca wyodrębnienie od dwóch do czterech perspektyw. Zalecenie to okazało się skuteczne w 97,0% analiz, nie zniwelowało jednak wyraźnych różnic między modelami.

Zgodnie z opisanymi wcześniej założeniami projektowymi każdy wynik analizy zawiera dwie odrębne oceny. Jedna z nich wskazywana jest bezpośrednio przez model, podczas gdy druga powstaje poprzez wyliczenie średniej ważonej. Ich rozbieżność wskazuje, na ile spójne jest rozpoznanie modelu na obu poziomach szczegółowości. Zestawienie przedstawia tabela 17.

Tabela 17. Rozbieżność między oceną całościową a ważoną

| Model      | n   | Średnia | Mediana | Maks. |
|------------|-----|---------|---------|-------|
| grok       | 502 | 0,64    | 0,53    | 2,96  |
| chatgpt    | 510 | 0,90    | 0,72    | 5,09  |
| claude     | 504 | 0,91    | 0,70    | 7,56  |
| gemini     | 509 | 0,92    | 0,75    | 9,00  |
| perplexity | 497 | 1,08    | 0,84    | 6,95  |

Różnice pomiędzy wartościami wynoszą od 0,64 do 1,08 punktu i są odwrotnie zależne od liczby odszukanych perspektyw. Wynika to w zasadniczej mierze z zależności rachunkowej: im więcej wykrytych grup, tym mniejsza waga każdej z nich w wartości końcowej, przez co przy czterech wartości skrajne są wygładzane silniej niż przy dwóch. Wielkość tej rozbieżności zasługuje na uwagę w zestawieniu z wynikami podrozdziału 6.3.1, gdzie średnie różnice między modelami wyniosły od 0,87 do 1,59 punktu. Widoczne jest więc, że zróżnicowanie

wewnętrzne pojedynczego modelu jest porównywalne z odchyleniem osiąganym pomiędzy dwoma odrębnymi modelami. Wartości te potwierdzają zasadność założenia przyjętego w podrozdziale 6.1.1, gdzie zdecydowano się na analizowanie ocen wystawianych bezpośrednio przez modele, a nie tych wyznaczanych na ich podstawie.

Reguła klasyfikacji opisana w podrozdziale 5.3.3 przypisuje etykietę wydźwięku mieszanego zagadnieniom, dla których wykryto stanowiska o przeciwnych znakach przy ocenie bliskiej zeru. Zależność pomiędzy ich liczbą a przypisaną etykietą jest zatem oczekiwana. Nie są to jednak różnice znaczące, jako że różnica średniej liczby perspektyw między przypadkami sklasyfikowanymi jako mieszane a pozostałymi wynosi od 0,10 dla modelu gemini do 0,52 dla modelu grok.

## 6.5 Ograniczenia badania

Badanie przedstawione w niniejszym rozdziale wiąże się z ograniczeniami wyznaczającymi zakres, w jakim uzyskane wyniki dają się uogólniać. Okoliczności te omówiono poniżej.

Każde z 510 zagadnień zostało przeanalizowane przez modele dokładnie jeden raz, ze względu na ograniczenia techniczne oraz kosztowe. Jest to o tyle istotne, że narzędzia te nie są deterministyczne nawet przy założeniu niskiego parametru sterującego losowością. W związku z tym różnice zaobserwowane pomiędzy modelami wynikają nie tylko z odmienności samych modeli, ale również ze zmienności będącej ich cechą. Ograniczenie to nie podważa wniosku o istnieniu systematycznych różnic, jednak wpływa na bezpośrednie porównywanie pojedynczych zagadnień. Mechanizm powtarzania został przewidziany przy konstrukcji narzędzia badawczego, jednak powtórzeń nie wykonano ze względu na koszt wywołań.

Parametr ograniczający długość generowanej odpowiedzi przyjmowany był przez cztery z pięciu modeli. Gemini nie obsługuje takiej wartości, w wyniku czego pracował na ustawieniu domyślnym dostawcy. Zostało to opisane szerzej w podrozdziale 5.2.1 oraz zaznaczono, że informacja ta była dołączana do każdej pozycji.

Jednym z elementów instrukcji było ograniczenie przypisywania oceny neutralnej tekstom, które były pozbawione zabarwienia emocjonalnego. Jest to bezpośrednia ingerencja w zachowanie modeli, w wyniku czego przedstawione wyniki opisują ich działanie w określonych warunkach, nie zaś swobodne. Wszystkie pięć badanych modeli ograniczono jednakowo, wobec czego nie zaburza to porównania uzyskanych wyników.

Badanie objęło wyłącznie zagadnienia produktowe sformułowane w języku angielskim, z przyczyn omówionych w podrozdziale 6.1.2. Wnioski nie obejmują wobec tego innych obszarów tematycznych ani treści w pozostałych językach, dla których zdolność rozpoznawania wydźwięku bywa zróżnicowana.

Utrwalenie materiału wejściowego wyeliminowało zmienność badanej treści pomiędzy modelami, nie usunęło jednak zależności wyników od charakterystyki wykorzystanych źródeł. Materiał pochodzący z serwisów o innym charakterze mógłby prowadzić do odmiennych rezultatów.

## 6.6 Wnioski

Podrozdział przedstawia odpowiedzi na pytania badawcze postawione w podrozdziale 6.1.1 oraz wskazuje ich znaczenie dla praktycznego wykorzystania rozwiązania.

Zgodność ocen liczbowych okazała się wysoka. Średnie odchylenie w obrębie zagadnienia wyniosło 0,96 punktu przy skali dwudziestopunktowej, a próg zgodności wysokiej przekroczyło 99,4% zagadnień. Klasyfikacja wydźwięku charakteryzuje się natomiast zgodnością jedynie umiarkowaną: wszystkie modele przypisały tę samą etykietę w 60,2% zagadnień, przy współczynniku kappa Cohena wynoszącym średnio 0,410. Rozbieżność pomiędzy tymi dwoma wynikami wynika ze sposobu powstawania klasyfikacji. Oceny skupiają się wokół środka skali, gdzie przebiegają progi rozdzielające poszczególne etykiety, wobec czego niewielkie różnice liczbowe prowadzą do odmiennej klasyfikacji nieproporcjonalnie często.

Niewspółmierności występujące między modelami są systematyczne i wykraczają poza losowość. Zastosowany test Friedmana odrzucił przypuszczenie o równoważności ( $\chi^2 = 62,15$ ;  $p < 0,001$ ). Dodatkowo bezpośrednie porównanie par wykazało istotne dysproporcje pomiędzy zwracanymi odpowiedziami dla ośmiu z dziesięciu par. Same modele cechują się zróżnicowanymi właściwościami. Model perplexity wykazuje tendencję do zaniżania ocen i odpowiada za 26 spośród 50 ocen najniższych w puli zagadnień najbardziej spornych, podczas gdy grok najkonsekwentniej zajmuje położenie pośrednie. Wykryte rozbieżności zależą od poruszanej tematyki. Odchylenie przekraczające 2,5 punktu wystąpiło dla 1,4% całego zbioru, najczęściej w obszarach budzących wyraźny spór, takich jak dobra luksusowe czy narzędzia programistyczne.

Zdolność wyodrębniania odmiennych perspektyw różnicuje modele silniej niż sama ocena wydźwięku. Jednakową ich liczbę wskazało wszystkie pięć modeli jedynie dla 6,5% zagadnień, przy średniej wynoszącej od 2,80 do 3,74 wykrytych perspektyw. Rozbieżność między oceną całościową a wypadkową perspektyw wynosi od 0,64 do 1,08 punktu, a więc tyle, ile wynoszą średnie różnice między odrębnymi modelami.

Wyniki uzyskane w toku badania wskazują na brak wymienności modeli językowych w zadaniach analizy sentymentu. Zmiana dostawcy znacząco wpływa na wartości zwracane przez system, wobec czego oparcie się wyłącznie na jednym z nich jest rozwiązaniem niewystarczającym. Uzasadnia to prowadzenie analizy kilkoma modelami jednocześnie i zestawianie ich wyników. Rozwiązanie oparte na pojedynczym modelu dostarcza bowiem oceny pozbawionej punktu odniesienia, która może okazać się myląca.

Podejście architektoniczne, które zostało opisane w rozdziale 5 umożliwia powtórzenie przedstawionego badania po dołączeniu kolejnych modeli przy zachowaniu identycznych warunków pomiaru. Materiał wejściowy zebrany podczas realizacji testów pozostaje dostępny wraz z uzyskanymi wynikami. Dzięki temu w przyszłości mogą być one porównywalne z przedstawionymi. Pozwala to śledzić zmiany zbieżności modeli wraz z pojawianiem się ich kolejnych wersji, co ma znaczenie wobec tempa rozwoju tej technologii.

## 7. Podsumowanie

Celem pracy było przeprowadzenie badania nad możliwościami analizy sentymentu wypowiedzi internetowych za pomocą aplikacji webowej, w której skład wchodzi różne modele językowe oraz źródła danych. Realizacja tego celu odbyła się poprzez zaprojektowanie oraz wykonanie systemu łączącego pięć modeli językowych oraz trzy zróżnicowane źródła treści. Komunikacja pomiędzy usługami zewnętrznymi odbywa się równolegle, podczas gdy wyniki prezentowane są obok siebie.

Drugim równorzędnym celem było zbadanie oraz porównanie odpowiedzi modeli na podstawie tych samych danych wejściowych. Do realizacji tego przedsięwzięcia stworzone zostało dedykowane narzędzie badawcze, opisane w podrozdziale 5.6, wykorzystujące mechanizmy przygotowane na potrzeby samej aplikacji. Samo badanie zostało przeprowadzone na zbiorze 510 zagadnień złożonych z siedemnastu różnych kategorii tematycznych. Uzyskane wyniki liczące 2550 rekordów zostały następnie poddane analizie przedstawionej w rozdziale 6.

Zrealizowane zostały również pomniejsze cele opisane szerzej w podrozdziale 1.2. Zaprojektowana została architektura trójwarstwowa składająca się z interfejsu, aplikacji serwerowej oraz bazy danych. Zintegrowano pięć modeli językowych umożliwiających równoległe przeprowadzanie analizy oraz zaimplementowano mechanizm pozyskiwania danych z trzech niezależnych źródeł internetowych. Dodatkowo opracowany został model analizy sentymentu, wzbogacony o wykrywanie wielu perspektyw odpowiadających stronom wypowiadającym się w danym tekście. Aplikację uzupełniono o rejestrację i uwierzytelnianie, panel historii zapytań oraz panel administracyjny.

W celu zapewnienia sprawnej rozbudowy systemu o dodatkowe pozycje modeli oraz źródeł wykorzystane zostały punkty rozszerzeń. Umożliwiają one przeprowadzenie operacji dodawania nowych pozycji do systemu bez ingerencji w pozostałe elementy systemu. Całość została zweryfikowana doświadczalnie w podrozdziale 5.2.3.

Postawione w pracy pytanie badawcze skupia się na stopniu zbieżności generowanych ocen przez wykorzystane modele. Wykazano, że w zależności od stosowanej miary zbieżność jest wysoka w przypadku ocen liczbowych lub umiarkowana dla klasyfikacji samego wydźwięku. Wskazane różnice mają charakter systematyczny, potwierdzony statystycznie dla ośmiu z dziesięciu par.

Niniejszy rozdział omawia zalety powstałego rozwiązania, jego wady i ograniczenia oraz możliwe kierunki dalszego rozwoju.

### 7.1 Zalety rozwiązania

Jedną z podstawowych zalet proponowanego systemu jest jednoczesne wykorzystanie wielu modeli językowych do analiz tego samego materiału wejściowego. Wszystkie rozwiązania omawiane w rozdziale 2 opierają swoje analizy wyłącznie na własnych, zamkniętych modelach, w wyniku czego odpowiedziom tym brakuje punktu odniesienia. Zawarte w rozdziale 6 wyniki jednoznacznie wskazują na systematyczne różnice pomiędzy odpowiedziami modeli. Dzięki temu zróżnicowanie to jest wartościowym elementem systemu i umożliwia uwiarygodnienie uzyskanego rezultatu.

Inną zaletą jest możliwość wykrywania odmiennych perspektyw, które dodatkowo posiadają przypisane wagi. Pojedyncza ocena wystawiona całości materiału sprawia, że utracone zostają wartościowe informacje o zróżnicowaniu wydźwięku prowadzonej wewnątrz

niego dyskusji. Materiał zawierający dwie wyraziste, przeciwstawne perspektywy daje w uśrednieniu wartość zbliżoną do treści pozbawionej jakiegokolwiek zabarwienia. Przyjęte w systemie podejście umożliwia rozróżnienie tych dwóch odmiennych sytuacji poprzez wykorzystanie właśnie perspektyw.

Architektura oparta na punktach rozszerzeń, opisana szerzej w podrozdziale 5.2, umożliwia rozbudowę systemu o dodatkowe modele oraz źródła danych przy minimalnych zmianach w kodzie. Rozwiązanie to nie tylko znacząco ułatwia utrzymanie i rozwój systemu, ale ponadto umożliwia powtarzanie badania porównawczego dla nowo dodanych pozycji bez ingerencji w samo narzędzie. Wobec tempa rozwoju tych technologii jest to istotna zaleta rozwiązania.

Odseparowanie warstwy domenowej od infrastruktury, omówione w podrozdziale 5.1.2, pozwoliło na użycie tego samego kodu do realizacji aplikacji oraz narzędzia badawczego. Sprawilo to, że zachowana została jednolitość pomiędzy wynikami uzyskiwanymi przez użytkownika końcowego oraz tymi uzyskanymi na potrzeby badań, a także ograniczyło konieczność dodatkowej implementacji.

Zastosowanie utrwalonego materiału wejściowego pozwoliło wyeliminować różnice wynikające ze zmienności treści internetowych. Zgromadzony materiał pozostaje przy tym dostępny po zakończeniu badania, co umożliwia jego ponowne wykorzystanie bez ponoszenia kosztów pozyskiwania danych.

Powstała aplikacja jest dostępna bez konieczności posiadania wiedzy programistycznej oraz bez opłat licencyjnych. Wykorzystywanie rozwiązań komercyjnych opisanych w podrozdziale 2.1 wiąże się z kosztami subskrypcji ograniczającymi ich dostępność dla mniejszych podmiotów, natomiast biblioteki omówione w podrozdziale 2.2 wymagają samodzielnego wdrożenia i wiedzy technicznej.

## 7.2 Wady i ograniczenia

Zaproponowane rozwiązanie wiąże się z ograniczeniami spowodowanymi zarówno podjętymi decyzjami projektowymi, jak i przyjętym zakresem pracy. Niniejszy podrozdział skupia się na ograniczeniach związanych z systemem, jako że te dotyczące badania omówiono w podrozdziale 6.5.

Podstawowym ograniczeniem aplikacji jest jej zależność od zewnętrznych usług, które znajdują się poza kontrolą systemu. Dostawcy modeli językowych dynamicznie zmieniają swoją ofertę, wprowadzając nowe wersje oraz wycofując dotychczasowe. Modyfikowane są przy tym zarówno parametry wywołania, jak i zasady rozliczania. Ponadto serwisy stanowiące źródła danych narzucają limity dobowe ograniczające liczbę zapytań możliwych do wykonania w danym okresie. Prowadzi to do konieczności utrzymywania integracji w reakcji na zmiany zachodzące poza systemem oraz uzależnienia jego działania od dostępności usług zewnętrznych.

Wagi przypisywane poszczególnym perspektywom przy obliczaniu średniej ważonej generowane są bezpośrednio przez model. To on szacuje zarówno pewność rozpoznania danej perspektywy, jak i jej udział w analizowanym tekście. Sprawia to, że system nie dysponuje żadnym mechanizmem weryfikacji zwracanych wartości.

Czas oczekiwania na wyniki przy równoległym wywołaniu wszystkich modeli może sięgać kilkudziesięciu sekund. Jest to spowodowane czasem odpowiedzi charakterystycznym dla każdego modelu. Wartość ta jest wyznaczana poza systemem i nie ma on na nią żadnego wpływu. Równoległe wykonanie sprawia, że odpowiedź uzależniona jest od najwolniej

wykonywanego elementu. Rozwiązanie to nie nadaje się więc do uzyskiwania natychmiastowych odpowiedzi.

Zakres prac nie obejmował wdrożenia produkcyjnego, co skutkuje określonymi ograniczeniami o charakterze technicznym. Zapisywanie wyników do bazy wymaga, aby pracowała ona w konfiguracji obsługującej transakcje, w wyniku czego wdrożenie na własnej infrastrukturze wiąże się z koniecznością przygotowania środowiska bazodanowego. Podobny charakter ma przechowywanie liczników prób logowania w pamięci procesu. Rozwiązanie to jest wystarczające dla pojedynczej instancji, natomiast przy kilku działających równolegle każda z nich prowadzi własny licznik, przez co dopuszczalna liczba prób rośnie proporcjonalnie do ich liczby.

Z zakresu pracy wyłączone zostały również testy automatyczne. W wyniku tej decyzji jedynymi mechanizmami weryfikującymi poprawność kodu pozostają statyczne typowanie realizowane przez kompilator języka TypeScript wraz z rygorystyczną konfiguracją oraz statyczna analiza kodu wykonywana przez narzędzie ESLint. Rozwiązanie to jest wystarczające do weryfikacji zgodności typów oraz wykrywania konstrukcji potencjalnie błędnych na etapie wytwarzania oprogramowania, nie stanowi jednak odpowiedniej ochrony w czasie wykonania.

Instrukcja przekazywana modelom ogranicza możliwość przypisania oceny neutralnej, co stanowi ingerencję w ich zachowanie. Rozstrzygnięcie to, uzasadnione w podrozdziale 5.3.1, wpływa na rozkład uzyskiwanych ocen i wymaga uwzględnienia przy interpretacji wyników prezentowanych użytkownikowi.

### 7.3 Plany na przyszłość

Omówione w poprzednim podrozdziale ograniczenia oraz właściwości architektury opisane w rozdziale 5 stanowią podstawę do wyznaczenia dalszych kierunków rozwoju aplikacji.

Pierwszy z nich polega na rozszerzeniu narzędzia badawczego o kilkukrotne powtórzenie analiz. Mechanizm ten został uwzględniony zarówno w systemie, jak i w schemacie przechowywanych danych. Wielokrotne przeanalizowanie tych samych pozycji ma na celu oddzielenie różnic wynikających z niedeterminizmu modeli od rozbieżności między nimi samymi.

Naturalnym krokiem jest również dodanie kolejnych modeli językowych oraz źródeł danych do aplikacji. Dzięki zastosowaniu architektury opartej na punktach rozszerzeń, opisanej w podrozdziale 5.2, proces ten sprowadza się do stworzenia dedykowanego pliku dla modelu oraz dopisania go do rejestru. Ponadto narzędzie badawcze umożliwia przeanalizowanie nowych pozycji na tych samych danych i zestawienie ich z wynikami poprzednich badań.

Zebrane dane badawcze, oparte na zdefiniowanych kategoriach tematycznych, umożliwiają porównanie zgodności modeli pomiędzy dziedzinami. Analiza taka nie została objęta zakresem pracy, natomiast zgromadzone wartości pozwalają przeprowadzić ją bez konieczności powtarzania przebiegu badawczego.

Możliwe jest również rozszerzenie badania o treści inne niż anglojęzyczne. Ze względu na zachowanie prostoty oraz jednolitości zdecydowano się na analizę tekstu wyłącznie w jednym języku, najlepiej dostępnym w wykorzystywanych źródłach. Zdolność modeli do analizy sentymentu bywa zróżnicowana w zależności od języka, wobec czego przeprowadzenie analogicznego badania na materiale w innym języku pozwoliłoby sprawdzić, czy zaobserwowane prawidłowości zachowują ważność.

Rozwoju wymaga również warstwa techniczna systemu. W skład takich zmian wchodzi uzupełnienie projektu o testy automatyczne oraz przygotowanie środowiska produkcyjnego. Wymagane jest w tym celu stworzenie konfiguracji bazodanowej obsługującej transakcje oraz przeniesienie liczników liczby żądań do pamięci współdzielonej między instancjami.

Ponadto możliwym kierunkiem rozwoju jest rozbudowa funkcjonalności analitycznej w aplikacji. Przykładowym elementem jest wprowadzenie śledzenia zmian wydźwięku dla danej tematyki w czasie czy rozszerzenie prezentowanych użytkownikowi miar o wskaźniki zgodności między modelami, wyznaczone obecnie wyłącznie na potrzeby badania.

## Bibliografia

- [1]. Leiner B.M., Cerf V.G., Clark D.D. i in. – Brief History of the Internet – Internet Society, 1997 – „<https://www.internetsociety.org/internet/history-internet/brief-history-internet/>” [Dostęp: 12.04.2026].
- [2]. Kemp S. – Digital 2025: Global Overview Report – DataReportal, 2025 – „<https://datareportal.com/reports/digital-2025-global-overview-report>” [Dostęp: 14.04.2026].
- [3]. O'Reilly T. – What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software – O'Reilly Media, 2005 – „<https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>” [Dostęp: 12.04.2026].
- [4]. Sentiment Analytics Market Size, Share, Trends & Forecast – Polaris Market Research, 2025 – „<https://www.polarismarketresearch.com/industry-analysis/sentiment-analytics-market>” [Dostęp: 14.04.2026].
- [5]. Liu B. – Sentiment Analysis: Mining Opinions, Sentiments, and Emotions – Cambridge University Press, 2020, ISBN 978-1-108-48637-8.
- [6]. Jurafsky D., Martin J.H. – Speech and Language Processing – wydanie 3, 2025 – „<https://web.stanford.edu/~jurafsky/slp3/>” [Dostęp: 18.04.2026].
- [7]. Brandwatch – „<https://www.brandwatch.com/>” [Dostęp: 20.04.2026].
- [8]. Sprout Social – „<https://sproutsocial.com/>” [Dostęp: 20.04.2026].
- [9]. Brand24 – „<https://brand24.pl/>” [Dostęp: 20.04.2026].
- [10]. Hutto C.J. – vaderSentiment – repozytorium GitHub – „<https://github.com/cjhutto/vadersentiment>” [Dostęp: 22.04.2026].
- [11]. Hutto C.J. – vaderSentiment Documentation – „<https://vadersentiment.readthedocs.io/en/latest/>” [Dostęp: 22.04.2026].
- [12]. Hutto C.J., Gilbert E. – VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text – Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1, 2014.
- [13]. Hugging Face – Transformers Documentation – „<https://huggingface.co/docs/transformers/en/index>” [Dostęp: 24.04.2026].
- [14]. Liu Y., Ott M., Goyal N. i in. – RoBERTa: A Robustly Optimized BERT Pretraining Approach – arXiv:1907.11692, 2019.
- [15]. Reddit API – „<https://www.reddit.com/dev/api/>” [Dostęp: 08.05.2025].
- [16]. Proferes N., Jones N., Gilbert S. i in. – Studying Reddit: A Systematic Overview of Disciplines, Approaches, Methods, and Ethics – Social Media + Society, vol. 7, no. 2, 2021.
- [17]. YouTube Data API v3 – „<https://developers.google.com/youtube/v3/docs>” [Dostęp: 08.05.2025].
- [18]. NewsAPI – „<https://newsapi.org/>” [Dostęp: 08.05.2025].
- [19]. Jones M., Bradley J., Sakimura N. – JSON Web Token (JWT) – RFC 7519, Internet Engineering Task Force, 2015.

- [20]. React – „<https://react.dev/>” [Dostęp: 22.04.2025].
- [21]. Statista – Most used web frameworks among developers worldwide, as of 2025 – „<https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>” [Dostęp: 24.04.2026].
- [22]. React Router – „<https://reactrouter.com/>” [Dostęp: 05.02.2026].
- [23]. Recharts – „<https://recharts.org/>” [Dostęp: 05.02.2026].
- [24]. TypeScript – „<https://www.typescriptlang.org/docs/>” [Dostęp: 22.04.2025].
- [25]. Tailwind CSS – „<https://tailwindcss.com/>” [Dostęp: 25.04.2025].
- [26]. Vite – „<https://vite.dev/>” [Dostęp: 25.04.2025].
- [27]. Node.js – „<https://nodejs.org/en>” [Dostęp: 28.04.2025].
- [28]. Express – „<https://expressjs.com/>” [Dostęp: 28.04.2025].
- [29]. Zod – „<https://zod.dev/>” [Dostęp: 05.02.2026].
- [30]. Helmet – „<https://helmetjs.github.io/>” [Dostęp: 05.02.2026].
- [31]. express-rate-limit – „<https://express-rate-limit.mintlify.app/>” [Dostęp: 05.02.2026].
- [32]. Axios – „<https://axios-http.com/docs/intro>” [Dostęp: 05.02.2026].
- [33]. cors – „<https://www.npmjs.com/package/cors>” [Dostęp: 05.02.2026].
- [34]. dotenv – „<https://www.npmjs.com/package/dotenv>” [Dostęp: 05.02.2026].
- [35]. MongoDB – „<https://www.mongodb.com/docs/>” [Dostęp: 30.04.2025].
- [36]. OpenAI – ChatGPT – „<https://openai.com/index/chatgpt/>” [Dostęp: 15.05.2026].
- [37]. Anthropic – Claude Developer Platform – „<https://platform.claude.com/docs/en/intro>” [Dostęp: 15.05.2026].
- [38]. Google – Gemini API – „<https://ai.google.dev/gemini-api/docs>” [Dostęp: 15.05.2026].
- [39]. Perplexity – API Platform – „<https://www.perplexity.ai/api-platform>” [Dostęp: 15.05.2026].
- [40]. xAI – Grok API Documentation – „<https://docs.x.ai/>” [Dostęp: 15.05.2026].
- [41]. Cohen J. – A Coefficient of Agreement for Nominal Scales – Educational and Psychological Measurement, vol. 20, no. 1, 1960.
- [42]. Landis J.R., Koch G.G. – The Measurement of Observer Agreement for Categorical Data – Biometrics, vol. 33, no. 1, 1977.
- [43]. Friedman M. – The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance – Journal of the American Statistical Association, vol. 32, no. 200, 1937.
- [44]. Wilcoxon F. – Individual Comparisons by Ranking Methods – Biometrics Bulletin, vol. 1, no. 6, 1945.
- [45]. Zheng L., Chiang W.-L., Sheng Y. i in. – Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena – Advances in Neural Information Processing Systems, vol. 36, 2023.
- [46]. Mongoose – „<https://mongoosejs.com/docs/>” [Dostęp: 30.04.2025].

[47]. bcrypt – „<https://www.npmjs.com/package/bcrypt>” [Dostęp: 30.04.2025].

## Spis ilustracji

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| Rysunek 1. Pulpit aplikacji Brandwatch. Źródło:[7]                                      | 11 |
| Rysunek 2. Pulpit aplikacji Sprout Social. Źródło:[8]                                   | 13 |
| Rysunek 3. Pulpit aplikacji Brand24. Źródło:[9]                                         | 14 |
| Rysunek 4. Diagram architektury systemu. Źródło: opracowanie własne                     | 19 |
| Rysunek 5. Diagram przepływu w systemie. Źródło: opracowanie własne                     | 20 |
| Rysunek 6. Struktura warstwy frontend aplikacji. Źródło: opracowanie własne             | 38 |
| Rysunek 7. Struktura warstwy backend. Źródło: opracowanie własne                        | 39 |
| Rysunek 8. Rozkłady ocen poszczególnych modeli. Źródło: opracowanie własne              | 64 |
| Rysunek 9. Wyniki porównań par modeli testem Wilcoxona. Źródło: opracowanie własne      | 67 |
| Rysunek 10. Zbieżność klasyfikacji wydźwięku dla par modeli. Źródło: opracowanie własne | 68 |

## Spis tabel

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Tabela 1. Zestawienie porównywanych rozwiązań                                       | 16 |
| Tabela 2. Zakres zmian wymaganych przy rozszerzeniu systemu                         | 44 |
| Tabela 3. Zależność doboru pozycji od parametrów zapytania                          | 49 |
| Tabela 4. Rodzaje niedostępności usług zewnętrznych                                 | 55 |
| Tabela 5. Postaci udostępniania zgromadzonych danych                                | 58 |
| Tabela 6. Warunki pomiaru                                                           | 60 |
| Tabela 7. Modele objęte badaniem                                                    | 60 |
| Tabela 8. Kategorie tematyczne objęte badaniem                                      | 61 |
| Tabela 9. Interpretacja wartości współczynnika kappa Cohena                         | 63 |
| Tabela 10. Statystyki opisowe ocen poszczególnych modeli                            | 64 |
| Tabela 11. Rozkład klasyfikacji wydźwięku [%]                                       | 65 |
| Tabela 12. Średnia różnica bezwzględna ocen dla par modeli                          | 66 |
| Tabela 13. Zagadnienia o największej rozbieżności ocen                              | 69 |
| Tabela 14. Udział modeli w rozbieżnościach (50 zagadnień o najwyższym rozproszeniu) | 69 |
| Tabela 15. Liczba perspektyw wykrywanych przez modele                               | 70 |
| Tabela 16. Rozkład liczby wykrytych perspektyw [%]                                  | 71 |
| Tabela 17. Rozbieżność między oceną całościową a ważoną                             | 71 |

## Spis listingów kodu

|                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------|----|
| Listing 1. Definicja skali ocen                                                                     | 40 |
| Listing 2. Sygnatura funkcji porządkującej nazwy źródeł danych                                      | 40 |
| Listing 3. Kontrakt dostawcy analizy sentymentu                                                     | 41 |
| Listing 4. Kontrakt źródła danych                                                                   | 42 |
| Listing 5. Rejestr dostawców analizy                                                                | 42 |
| Listing 6. Punkt końcowy udostępniający zestawienie dostawców                                       | 43 |
| Listing 7. Wyznaczanie progów skali w treści instrukcji                                             | 45 |
| Listing 8. Fragment instrukcji systemowej definiującej wymagany format odpowiedzi modeli językowych | 45 |
| Listing 9. Fragment instrukcji systemowej definiującej zasady wykrywania perspektyw                 | 45 |
| Listing 10. Sygnalizowanie pól nieodczytanych                                                       | 47 |
| Listing 11. Wyznaczanie oceny ważonej                                                               | 48 |
| Listing 12. Klasyfikacja wydźwięku                                                                  | 48 |
| Listing 13. Wyznaczanie stanu początkowego generatora                                               | 49 |
| Listing 14. Koordynacja wywołań źródeł danych                                                       | 50 |
| Listing 15. Niepodzielny zapis wyników analizy                                                      | 52 |
| Listing 16. Warunek kwalifikujący rekordy do zestawień                                              | 53 |
| Listing 17. Warstwa pośrednia sprawdzająca poprawność danych                                        | 54 |
| Listing 18. Wyznaczanie skrótu instrukcji                                                           | 56 |

## Dodatek A. Zbiór zagadnień wykorzystany w badaniu

Dodatek zawiera pełny wykaz 510 zagadnień wykorzystanych w badaniu przedstawionym w rozdziale szóstym, uporządkowany według kategorii tematycznych. Identyfikatory odpowiadają wartościom zapisywanym wraz z wynikami poszczególnych analiz.

### Elektronika użytkowa

|      |                    |      |                     |      |                         |
|------|--------------------|------|---------------------|------|-------------------------|
| t001 | iPhone 17          | t011 | Dell XPS 14         | t021 | Roborock S8 MaxV        |
| t002 | Samsung Galaxy S25 | t012 | Framework Laptop 13 | t022 | Anker Prime power bank  |
| t003 | Google Pixel 10    | t013 | iPad Pro M4         | t023 | Logitech MX Master 4    |
| t004 | Nintendo Switch 2  | t014 | Kindle Scribe       | t024 | Keychron Q1 keyboard    |
| t005 | Steam Deck OLED    | t015 | GoPro Hero 13       | t025 | Raspberry Pi 5          |
| t006 | Sony WH-1000XM6    | t016 | DJI Mini 4 Pro      | t026 | Garmin Fenix 8          |
| t007 | AirPods Pro 3      | t017 | Ring Video Doorbell | t027 | Oura Ring 4             |
| t008 | Apple Vision Pro   | t018 | Sonos Era 300       | t028 | Nothing Phone 3         |
| t009 | Meta Quest 3S      | t019 | LG C4 OLED TV       | t029 | Bose QuietComfort Ultra |
| t010 | MacBook Air M4     | t020 | Dyson V15 Detect    | t030 | Nvidia RTX 5080         |

### Motoryzacja

|      |                      |      |                   |      |                     |
|------|----------------------|------|-------------------|------|---------------------|
| t031 | Tesla Model Y        | t041 | Mercedes EQS      | t051 | BYD Seal            |
| t032 | Tesla Cybertruck     | t042 | Porsche Macan EV  | t052 | Renault 5 E-Tech    |
| t033 | Rivian R1S           | t043 | Lucid Air         | t053 | Fiat 500e           |
| t034 | Ford F-150 Lightning | t044 | Polestar 3        | t054 | Audi Q6 e-tron      |
| t035 | Toyota RAV4 Hybrid   | t045 | Volvo EX30        | t055 | Range Rover Sport   |
| t036 | Honda Civic Type R   | t046 | Subaru Outback    | t056 | Dacia Duster        |
| t037 | Hyundai Ioniq 5      | t047 | Mazda CX-90       | t057 | Skoda Enyaq         |
| t038 | Kia EV9              | t048 | Jeep Wrangler 4xe | t058 | Peugeot 3008        |
| t039 | Volkswagen ID.4      | t049 | Chevrolet Bolt    | t059 | Cupra Born          |
| t040 | BMW i4               | t050 | Nissan Ariya      | t060 | Toyota Land Cruiser |

### Serwisy strumieniowe

|      |                              |      |                   |      |                  |
|------|------------------------------|------|-------------------|------|------------------|
| t061 | Netflix price increase       | t071 | Hulu bundle       | t081 | Kanopy           |
| t062 | Disney Plus password sharing | t072 | Crunchyroll       | t082 | Mubi             |
| t063 | HBO Max rebrand              | t073 | Audible credits   | t083 | Shudder          |
| t064 | Spotify Wrapped              | t074 | Spotify HiFi      | t084 | BritBox          |
| t065 | Apple TV Plus                | t075 | SoundCloud        | t085 | Discovery Plus   |
| t066 | Amazon Prime Video ads       | t076 | Tidal lossless    | t086 | ESPN Plus        |
| t067 | YouTube Premium              | t077 | Pandora radio     | t087 | DAZN             |
| t068 | Twitch subscription changes  | t078 | Deezer            | t088 | Nebula streaming |
| t069 | Paramount Plus               | t079 | Plex media server | t089 | Dropout TV       |
| t070 | Peacock streaming            | t080 | Jellyfin          | t090 | Patreon creators |

### Gry komputerowe

|      |                                  |      |                             |      |                         |
|------|----------------------------------|------|-----------------------------|------|-------------------------|
| t091 | Baldur's Gate 3                  | t101 | Call of Duty Black Ops 6    | t111 | League of Legends patch |
| t092 | Elden Ring Shadow of the Erdtree | t102 | EA FC 25                    | t112 | Dota 2                  |
| t093 | Cyberpunk 2077 Phantom Liberty   | t103 | Diablo 4                    | t113 | Apex Legends            |
| t094 | Starfield                        | t104 | World of Warcraft expansion | t114 | Overwatch 2             |
| t095 | Helldivers 2                     | t105 | Final Fantasy 16            | t115 | Destiny 2               |
| t096 | Palworld                         | t106 | Zelda Tears of the Kingdom  | t116 | No Man's Sky update     |

|      |                       |      |                              |      |                        |
|------|-----------------------|------|------------------------------|------|------------------------|
| t097 | Hades 2               | t107 | Super Mario Bros Wonder      | t117 | Factorio Space Age     |
| t098 | Stardew Valley update | t108 | Animal Crossing New Horizons | t118 | Terraria               |
| t099 | Minecraft update      | t109 | Counter-Strike 2             | t119 | Hollow Knight Silksong |
| t100 | Fortnite season       | t110 | Valorant                     | t120 | GTA 6 trailer          |

## Podróże

|      |                          |      |                     |      |                      |
|------|--------------------------|------|---------------------|------|----------------------|
| t121 | Ryanair customer service | t131 | Delta SkyMiles      | t141 | Expedia              |
| t122 | Wizz Air delays          | t132 | United Airlines     | t142 | Skyscanner           |
| t123 | easyJet baggage policy   | t133 | American Airlines   | t143 | Hostelworld          |
| t124 | Lufthansa strike         | t134 | Southwest boarding  | t144 | Trainline            |
| t125 | British Airways refunds  | t135 | JetBlue             | t145 | Eurostar             |
| t126 | Air France               | t136 | Norwegian Air       | t146 | Deutsche Bahn delays |
| t127 | KLM                      | t137 | Vueling             | t147 | Amtrak               |
| t128 | Emirates business class  | t138 | LOT Polish Airlines | t148 | Uber pricing         |
| t129 | Qatar Airways            | t139 | Airbnb fees         | t149 | Bolt rides           |
| t130 | Turkish Airlines         | t140 | Booking.com         | t150 | Lime scooters        |

## Żywność i napoje

|      |                     |      |                      |      |                    |
|------|---------------------|------|----------------------|------|--------------------|
| t151 | Oatly oat milk      | t161 | Domino's pizza       | t171 | Red Bull           |
| t152 | Beyond Meat burger  | t162 | Papa John's          | t172 | Monster Energy     |
| t153 | Impossible Foods    | t163 | Five Guys            | t173 | Liquid Death       |
| t154 | Starbucks prices    | t164 | Shake Shack          | t174 | Coca-Cola Spiced   |
| t155 | Dunkin coffee       | t165 | Pret A Manger        | t175 | Pepsi Wild Cherry  |
| t156 | Chipotle portions   | t166 | Greggs               | t176 | Ben and Jerry's    |
| t157 | McDonald's McPlant  | t167 | Huel                 | t177 | Haagen-Dazs        |
| t158 | Burger King Whopper | t168 | Soylent              | t178 | Lindt chocolate    |
| t159 | KFC                 | t169 | Celsius energy drink | t179 | Tony's Chocolonely |
| t160 | Subway sandwiches   | t170 | Prime Hydration      | t180 | HelloFresh         |

## Aplikacje finansowe

|      |                |      |                      |      |                         |
|------|----------------|------|----------------------|------|-------------------------|
| t181 | Revolut        | t191 | Robinhood            | t201 | Nubank                  |
| t182 | Wise transfers | t192 | eToro                | t202 | Curve card              |
| t183 | Monzo          | t193 | Coinbase             | t203 | Plum savings            |
| t184 | N26            | t194 | Binance              | t204 | Moneybox                |
| t185 | Chime          | t195 | Kraken exchange      | t205 | Freertrade              |
| t186 | Cash App       | t196 | Trading212           | t206 | Wealthsimple            |
| t187 | Venmo          | t197 | Interactive Brokers  | t207 | SoFi                    |
| t188 | PayPal fees    | t198 | Vanguard funds       | t208 | Marcus by Goldman Sachs |
| t189 | Klarna         | t199 | Fidelity Investments | t209 | Ally Bank               |
| t190 | Afterpay       | t200 | Charles Schwab       | t210 | Chase UK                |

## Narzędzia programowe

|      |                      |      |                  |      |                          |
|------|----------------------|------|------------------|------|--------------------------|
| t211 | Duolingo streak      | t221 | Dropbox          | t231 | JetBrains IDE            |
| t212 | Notion AI            | t222 | 1Password        | t232 | Visual Studio Code       |
| t213 | Obsidian notes       | t223 | Bitwarden        | t233 | Docker Desktop licensing |
| t214 | Figma pricing        | t224 | NordVPN          | t234 | Postman                  |
| t215 | Adobe Creative Cloud | t225 | ExpressVPN       | t235 | Linear app               |
| t216 | Canva Pro            | t226 | Proton Mail      | t236 | Jira                     |
| t217 | Slack pricing        | t227 | Signal messenger | t237 | Asana                    |
| t218 | Microsoft Teams      | t228 | Telegram         | t238 | Trello                   |
| t219 | Zoom meetings        | t229 | Discord Nitro    | t239 | Miro                     |

t220 Google Workspace

t230 GitHub Copilot

t240 Grammarly

## Kosmetyki i higiena

t241 CeraVe cleanser

t251 Glossier

t261 Oral-B iO toothbrush

t242 The Ordinary serum

t252 e.l.f. Cosmetics

t262 Philips Sonicare

t243 Rhode lip peptide

t253 Byoma skincare

t263 Nivea

t244 Fenty Beauty

t254 Paula's Choice

t264 Bioderma micellar water

t245 Rare Beauty blush

t255 La Roche-Posay

t265 Supergoop sunscreen

t246 Drunk Elephant

t256 Cetaphil

t266 Laneige lip mask

t247 Charlotte Tilbury

t257 Aesop hand wash

t267 Medik8

t248 Sol de Janeiro Bum Bum

t258 Native deodorant

t268 Kiehl's

t249 Dyson Airwrap

t259 Dr. Squatch soap

t269 Maybelline Sky High  
mascara

t250 Olaplex

t260 Gillette razors

t270 L'Oreal Paris

## Odzież i obuwie

t271 Nike Air Max

t281 On Cloud running shoes

t291 Nike Dunk

t272 Adidas Samba

t282 Hoka Clifton

t292 Skims

t273 Zara

t283 Crocs

t293 Aritzia

t274 Shein hauls

t284 Dr. Martens

t294 Abercrombie and Fitch

t275 Lululemon leggings

t285 Vans

t295 Vuori

t276 Uniqlo

t286 Levi's 501

t296 Alo Yoga

t277 H&M

t287 Patagonia

t297 Carhartt

t278 Primark

t288 The North Face

t298 Under Armour

t279 Gymshark

t289 Ralph Lauren

t299 Puma

t280 New Balance 550

t290 Tommy Hilfiger

t300 Ugg boots

## Zdrowie i sprawność

t301 Whoop band

t311 Zwift

t321 Withings scale

t302 Peloton bike

t312 Hims

t322 Levels glucose monitor

t303 Fitbit Charge 6

t313 Ro health

t323 AG1 Athletic Greens

t304 Strava subscription

t314 Eight Sleep pod

t324 Ritual vitamins

t305 MyFitnessPal

t315 Theragun

t325 Seed probiotics

t306 Calm app

t316 Hydrow rower

t326 Function Health

t307 Headspace

t317 Tonal home gym

t327 Hinge Health

t308 Planet Fitness

t318 Apple Fitness Plus

t328 ClassPass

t309 Noom

t319 Nike Training Club

t329 Orangetheory Fitness

t310 WeightWatchers

t320 Fitbod

t330 Ultrahuman Ring

## Handel elektroniczny

t331 Amazon returns policy

t341 Instacart fees

t351 Zalando returns

t332 Temu

t342 Wish

t352 ASOS returns

t333 AliExpress

t343 Alibaba

t353 IKEA

t334 eBay

t344 Mercari

t354 Best Buy

t335 Etsy sellers

t345 Poshmark

t355 Sam's Club

t336 Walmart Plus

t346 Depop

t356 Ocado

t337 Target Circle

t347 StockX

t357 Nordstrom

t338 Costco membership

t348 Newegg

t358 Macy's

t339 Wayfair

t349 Rakuten cashback

t359 Kohl's

t340 Chewy

t350 Flipkart

t360 Back Market refurbished

## Dobra luksusowe

|      |                       |      |                      |      |                 |
|------|-----------------------|------|----------------------|------|-----------------|
| t361 | Rolex Submariner      | t371 | Burberry             | t381 | Audemars Piguet |
| t362 | Omega Speedmaster     | t372 | Balenciaga           | t382 | TAG Heuer       |
| t363 | Louis Vuitton         | t373 | Bottega Veneta       | t383 | Tudor Black Bay |
| t364 | Gucci                 | t374 | Saint Laurent        | t384 | Breitling       |
| t365 | Hermes Birkin         | t375 | Versace              | t385 | Montblanc pen   |
| t366 | Chanel handbag        | t376 | Fendi                | t386 | Rimowa luggage  |
| t367 | Cartier Love bracelet | t377 | Loewe                | t387 | Coach           |
| t368 | Prada                 | t378 | Bvlgari              | t388 | Michael Kors    |
| t369 | Tiffany and Co        | t379 | Van Cleef and Arpels | t389 | Longines        |
| t370 | Dior                  | t380 | Patek Philippe       | t390 | Moncler jacket  |

### Artykuły dla zwierząt

|      |                      |      |                      |      |                           |
|------|----------------------|------|----------------------|------|---------------------------|
| t391 | Purina Pro Plan      | t401 | PetSmart             | t411 | Iams                      |
| t392 | Royal Canin          | t402 | Petco                | t412 | Sheba                     |
| t393 | Hill's Science Diet  | t403 | Kong toys            | t413 | Temptations treats        |
| t394 | Blue Buffalo         | t404 | Litter-Robot         | t414 | Greenies dental           |
| t395 | Pedigree             | t405 | Furbo dog camera     | t415 | Cesar dog food            |
| t396 | Whiskas              | t406 | Tractive GPS tracker | t416 | Wellness pet food         |
| t397 | Friskies             | t407 | Rover dog walking    | t417 | Merrick pet food          |
| t398 | The Farmer's Dog     | t408 | Wag pet care         | t418 | Zesty Paws supplements    |
| t399 | Ollie fresh dog food | t409 | Trupanion insurance  | t419 | Embark dog DNA            |
| t400 | Freshpet             | t410 | Nutro                | t420 | Arm and Hammer cat litter |

### Media społecznościowe

|      |                  |      |           |      |               |
|------|------------------|------|-----------|------|---------------|
| t421 | TikTok           | t431 | Mastodon  | t441 | Nextdoor      |
| t422 | Instagram Reels  | t432 | Tumblr    | t442 | BeReal        |
| t423 | Facebook         | t433 | WhatsApp  | t443 | Lemon8        |
| t424 | X Premium        | t434 | Messenger | t444 | Truth Social  |
| t425 | Reddit           | t435 | WeChat    | t445 | Rumble        |
| t426 | Snapchat         | t436 | Weibo     | t446 | OnlyFans      |
| t427 | Pinterest        | t437 | Line app  | t447 | Substack      |
| t428 | LinkedIn Premium | t438 | KakaoTalk | t448 | Medium        |
| t429 | Threads app      | t439 | VK        | t449 | Meta Verified |
| t430 | Bluesky          | t440 | Quora     | t450 | Snapchat Plus |

### Sprzęt gospodarstwa domowego

|      |                        |      |                        |      |                     |
|------|------------------------|------|------------------------|------|---------------------|
| t451 | Instant Pot            | t461 | iRobot Roomba j9       | t471 | Philips air fryer   |
| t452 | Ninja blender          | t462 | Samsung Bespoke fridge | t472 | Cosori air fryer    |
| t453 | KitchenAid stand mixer | t463 | LG washing machine     | t473 | Le Creuset          |
| t454 | Vitamix                | t464 | Whirlpool              | t474 | Lodge cast iron     |
| t455 | Nespresso Vertuo       | t465 | Bosch dishwasher       | t475 | Tefal               |
| t456 | Keurig K-Supreme       | t466 | Miele vacuum           | t476 | Dreame vacuum       |
| t457 | Breville Barista       | t467 | Traeger grill          | t477 | Levoit air purifier |
| t458 | SodaStream             | t468 | Weber grill            | t478 | Blueair purifier    |
| t459 | Shark vacuum           | t469 | Ooni pizza oven        | t479 | GE Profile range    |
| t460 | Bissell CrossWave      | t470 | Cricut Maker           | t480 | Ember mug           |

### Zabawki i artykuły dziecięce

|      |                     |      |                  |      |                    |
|------|---------------------|------|------------------|------|--------------------|
| t481 | LEGO Star Wars sets | t491 | Melissa and Doug | t501 | Crayola            |
| t482 | Barbie doll         | t492 | Hasbro           | t502 | Sylvanian Families |
| t483 | Hot Wheels          | t493 | Mattel           | t503 | Tamagotchi         |

|      |               |      |                   |      |                     |
|------|---------------|------|-------------------|------|---------------------|
| t484 | Nerf blaster  | t494 | Bandai            | t504 | Rubik's Cube        |
| t485 | Play-Doh      | t495 | Beyblade          | t505 | LEGO Technic        |
| t486 | Funko Pop     | t496 | Transformers toys | t506 | Playmobil           |
| t487 | Squishmallows | t497 | My Little Pony    | t507 | Bluey toys          |
| t488 | Pokemon TCG   | t498 | Paw Patrol toys   | t508 | Little Tikes        |
| t489 | Magna-Tiles   | t499 | L.O.L. Surprise   | t509 | Monopoly board game |
| t490 | Fisher-Price  | t500 | Peppa Pig toys    | t510 | Hatchimals          |