

Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

Wprowadzenie do przetwarzania języka naturalnego

Rozwiązanie multi-agent LLM - dokumentacja  
wstępna projektu

Paweł Grzankowski, Igor Szablowski

Warszawa, 2024

# Spis treści

<b>1. Wstęp</b>	2
1.1. Cel projektu	2
1.2. Motywacja i korzyści	2
<b>2. Opis projektu</b>	3
2.1. Założenia systemu	3
2.2. Opis ról agentów	3
2.3. Zakres funkcjonalności	3
<b>3. Technologie i narzędzia</b>	5
3.1. Model językowy	5
3.2. Narzędzia	5
<b>4. Architektura systemu</b>	6
4.1. Główne moduły	6
4.2. Integracja z narzędziami zewnętrznymi	6
<b>5. Wyzwania i ograniczenia</b>	7

# 1. Wstęp

## 1.1. Cel projektu

Projekt ma na celu stworzenie aplikacji opartej na systemie multi-agentowym z wykorzystaniem modelu Llama 3, która umożliwi automatyczną debatę na dowolny zadany temat. W aplikacji jeden agent będzie weryfikował poprawność i sensowność tematu, dwóch kolejnych agentów prowadzić będzie debatę – jeden argumentując "za", a drugi "przeciw", a na koniec jeden agent podsumuje całą debatę w formie raportu. Dodatkowo agenci będą mogli korzystać z narzędzi zewnętrznych, takich jak wyszukiwanie informacji w internecie, aby wzmacniać swoje argumenty. Wyniki debaty zostaną podsumowane w formie raportu, który może być wykorzystany do podejmowania decyzji lub celów edukacyjnych.

## 1.2. Motywacja i korzyści

Motywacją do stworzenia tego projektu jest potrzeba dogłębszej analizy złożonych tematów, które nie mają jednoznacznych odpowiedzi. Tradycyjne systemy analizy opierają się na przedstawieniu pojedynczego stanowiska, co może prowadzić do pomijania ważnych aspektów lub alternatywnych punktów widzenia. Rozwiązanie to sprawdza się szczególnie w sytuacjach, gdzie konieczne jest zbadanie wieloaspektowych tematów lub problemów bez jednoznacznego rozwiązania.

## 2. Opis projektu

Celem aplikacji jest umożliwienie użytkownikowi przeprowadzenia debaty na wybrany temat z udziałem dwóch agentów opartych na modelach językowych. Jeden agent prezentuje argumenty "za", a drugi "przeciw", tworząc dynamiczną i interaktywną symulację debaty. Użytkownik może wprowadzić temat, śledzić wymianę argumentów między agentami oraz ocenić, która strona zaprezentowała lepsze stanowisko.

### 2.1. Założenia systemu

System zakłada następujące główne założenia:

- Użytkownik dostarcza temat debaty w formie tekstowej, a system przetwarza go i przekazuje agentom.
- System wykorzystuje model językowy (LLM) do generowania odpowiedzi, przy czym każdy agent przyjmuje inną perspektywę (argumenty "za" lub "przeciw").
- Interakcja między agentami przebiega według ustalonej liczby rund, w której każda strona reaguje na argumenty przeciwnika.
- Użytkownik ma możliwość śledzenia debaty w czasie rzeczywistym.
- Po zakończeniu debaty generowane jest podsumowanie przez agenta podsumowującego.

### 2.2. Opis ról agentów

Aplikacja zawiera agentów:

- **Agent walidujący temat:** Ten agent ma za zadanie sprawdzenie czy dostarczony temat nadaje się do debaty, jeśli tak to zostaje przeprowadzona, jeśli nie użytkownik zostaje o tym poinformowany.
- **Agent "Za":** Ten agent reprezentuje stanowisko wspierające temat debaty. Jego zadaniem jest generowanie argumentów na poparcie stanowiska oraz odpowiadanie na kontrargumenty przeciwnika.
- **Agent "Przeciw":** Ten agent zajmuje pozycję krytyczną wobec tematu debaty. Formuluje argumenty przeciwne i reaguje na twierdzenia przedstawione przez agenta "Za".
- **Agent podsumowujący:** Zadaniem tego agenta jest podsumowanie całości i ewentualne stwierdzenie strony wygrywającej.

Każdy agent działa w oparciu o specyficzne instrukcje i prompty dostosowane do jego roli, dzięki czemu odpowiedzi są spójne z założonym stanowiskiem. Agenci mogą również korzystać z kontekstu poprzednich wypowiedzi, aby wzmacniać swoje argumenty.

### 2.3. Zakres funkcjonalności

System oferuje następujące funkcjonalności:

- **Wprowadzanie tematu debaty:** Użytkownik może wpisać dowolny temat, który zostanie przeanalizowany przez system pod kątem poprawności i przygotowany do debaty.

- 
- **Generowanie argumentów:** Agenci automatycznie generują argumenty "za" i "przeciw" w oparciu o wprowadzone dane.
  - **Interakcja między agentami:** Debata jest prowadzona w ustalonych rundach, gdzie każdy agent odpowiada na wypowiedzi przeciwnika, tworząc dynamiczną wymianę zdań.
  - **Prezentacja wyników:** Użytkownik ma możliwość śledzenia przebiegu debaty w czasie rzeczywistym, a po jej zakończeniu system generuje podsumowanie argumentów obu stron.

## 3. Technologie i narzędzia

### 3.1. Model językowy

W projekcie planowane jest wykorzystanie modelu językowego Llama 3, który jest zaawansowanym systemem przetwarzania języka naturalnego, stworzonym przez Meta. Model ten został wybrany ze względu na swoją zdolność do generowania spójnych, logicznych i kontekstowo trafnych odpowiedzi, a co ważniejsze jest dostępny open-source i może być wykorzystywany lokalnie.

Llama 3 będzie odpowiadał za kluczowe funkcje aplikacji, takie jak walidacja tematów, generowanie argumentów oraz analizowanie informacji z zewnętrznych źródeł. Model zapewni precyzyjne przetwarzanie języka naturalnego, wspierając agenta "za" i agenta "przeciw" w prowadzeniu merytorycznej i logicznej debaty. Dodatkowo jego elastyczna architektura umożliwi integrację z narzędziami takimi jak wyszukiwanie informacji w internecie, co pozwoli agentom na wzmacnianie swoich argumentów za pomocą aktualnych danych.

### 3.2. Narzędzia

W projekcie planowane jest wykorzystanie środowiska programistycznego Python, które ze względu na swoją wszechstronność i bogaty ekosystem bibliotek doskonale nadaje się do tworzenia aplikacji opartych na modelach językowych i systemach multi-agentowych. Python pozwala na łatwą integrację z zaawansowanymi modelami NLP, narzędziami zewnętrznymi oraz umożliwia szybkie prototypowanie i testowanie rozwiązań.

Do implementacji planuje się użycie bibliotek wspierających pracę z modelami językowymi, takich jak te obsługujące Llama 3, np. Ollama czy Hugging Face Transformers. W celu zarządzania interakcjami między agentami oraz ich integracją z narzędziami zewnętrznymi można zastosować dedykowane frameworki wspierające budowę aplikacji opartych na modelach NLP np.: LangChain oraz LangGraph. Rozważane jest także wykorzystanie narzędzi umożliwiających wyszukiwanie i analizowanie informacji z internetu w celu wzbogacenia argumentacji agentów.

Konkretne biblioteki i narzędzia zostaną dobrane w późniejszym etapie projektu w zależności od wymagań funkcjonalnych i technicznych. Python zapewni wystarczającą elastyczność, aby zrealizować wszystkie założenia projektu w sposób efektywny i skalowalny.

## 4. Architektura systemu

### 4.1. Główne moduły

Architektura systemu opiera się na modularnym podejściu, które pozwala na łatwe zarządzanie i rozwijanie aplikacji. Pierwszym modulem jest moduł wprowadzania danych, który odpowiada za interakcję z użytkownikiem, umożliwiając wprowadzenie tematu debaty. Dane te są następnie przekazywane do agenta walidującego temat. Kolejnym elementem jest moduł walidacji tematu, obsługiwany przez agenta walidującego, który sprawdza, czy temat nadaje się do debaty. Jeśli temat jest nieodpowiedni, użytkownik otrzymuje odpowiednią informację zwrotną. Kluczowym elementem aplikacji jest moduł debaty, który obsługuje interakcje między agentami „Za” i „Przeciw”. Ten moduł zarządza mechanizmem koordynacji wymiany argumentów w ustalonych rundach. Na końcu znajduje się moduł prezentacji wyników, odpowiedzialny za generowanie podsumowania debaty przez agenta podsumowującego oraz wyświetlanie rezultatów w czytelnej i przyjaznej dla użytkownika formie.

### 4.2. Integracja z narzędziami zewnętrznymi

System zakłada integrację z narzędziami zewnętrznymi, które zwiększą efektywność i jakość argumentów generowanych przez agentów. Narzędzia takie jak Hugging Face Transformers umożliwią zaawansowaną obróbkę tekstu i dopasowanie modeli językowych do specyfiki projektu. W celu koordynacji interakcji między agentami planuje się wykorzystanie frameworków dla systemów multi-agentowych, takich jak LangChain, które ułatwią zarządzanie przepływem danych i kontekstem wypowiedzi. Integracja z tymi narzędziami pozwoli na dynamiczne dostosowanie systemu do nowych potrzeb i stopniowe ulepszanie jego funkcjonalności.

## 5. Wyzwania i ograniczenia

Zapewnienie wysokiej jakości i merytoryczności generowanych argumentów jest jednym z głównych wyzwań tego projektu. Ograniczenia modeli językowych, takich jak Llama 3, mogą powodować generowanie nieistotnych, błędnych lub powierzchownych argumentów. Dodatkowym problemem jest brak pełnego kontekstu w przypadku tematów wymagających specjalistycznej wiedzy, co może ograniczać zdolność modeli do formułowania trafnych odpowiedzi. Ponadto, generowanie odpowiedzi przez agentów w czasie rzeczywistym może obciążać system i negatywnie wpływać na jego responsywność.