



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Dokumentacja projektu grupowego

Dokumentacja techniczna projektu

Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańskia

{wersja dokumentu wzorcowego: wersja 1/2025}

Nazwa i akronim projektu: Rozproszona współpraca LLM inspirowana umysłami rojowymi - HMLLM	Zleceniodawca: Politechnika Gdańskia, wydział Elektroniki, Telekomunikacji i Informatyki, Katedra Architektury Systemów Komputerowych	
Numer zlecenia: ID-352	Kierownik projektu: Nowak, Jakub	Opiekun projektu: Majkutewicz, Jan, mgr inż.

Nazwa dokumentu/akronim: Dokumentacja techniczna projektu – DTP	Nr wersji: 1.2
Odpowiedzialny za dokument: Sobiech, Alicja	Data pierwszego sporządzenia: 9.11.2025
	Data ostatniej aktualizacji: 28.01.2026
	Studia I stopnia, inżynierskie Semestr realizacji Projektu grupowego: 1 i 2

Historia zmian

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.0	Wstępna wersja wprowadzenie	1.2, 1.3, 1.4	Sobiech, Alicja	9.11.2025
1.1	Uzupełnienie dokumentacji	całość	Sobiech, Alicja	27.01.2026
1.2	Rozpisanie działania programu	2.4.2, 2.6	Nowak, Jakub	28.01.2026

Spis treści

1	Wprowadzenie - o dokumencie	3
1.1	Cel dokumentu	3
1.2	Zakres dokumentu	3
1.3	Odbiorcy	3
1.4	Terminologia	3
2	Dokumentacja techniczna projektu	3
2.1	Cele i założenia projektu	3
2.2	Cechy funkcjonalne produktu	3
2.3	Wykorzystane technologie i narzędzia	4
2.4	Oprogramowanie	4
2.4.1	Agenci	4
2.4.2	Generowanie i ewaluacja wyniku	4
2.5	Testy	5
2.5.1	Przykładowe prompty	5
2.5.2	Kryteria poprawności	5
2.6	Wyniki działania	5
3	Załączniki	5

1 Wprowadzenie - o dokumencie

1.1 Cel dokumentu

Celem dokumentu jest udokumentowanie informacji dotyczących produktu, jego cech funkcjonalnych, parametrów technicznych, schematów blokowych, oprogramowania, wyników działania, zdjęć produktu, pomiarów, testów oraz innych elementów wymaganych przez opiekuna i klienta.

1.2 Zakres dokumentu

Dokument obejmuje opis wykorzystanych narzędzi i technologii, testy oraz ich wyniki, przykładowe prompty, cechy rozwiązania oraz opis jego działania

Dokument nie obejmuje informacji dotyczących organizacji i harmonogramu wytwarzania aplikacji opisanej w tym dokumencie.

1.3 Odbiorcy

Adresatami dokumentu są:

- Zleceniodzierca projektu: Katedra Architektury Systemów Komputerowych (klient wewnętrzny PG)
- Opiekun projektu: mgr inż. Jan Majkutewicz
- Członkowie zespołu projektowego: Jakub Nowak, Alicja Sobiech, Oliwier Komorowski, Kacper Skudlarz

1.4 Terminologia

- **Agent** - jednostka koordynująca pracę modelu LLM w celu realizacji określonych zadań. Agent może wysyłać, odbierać i interpretować komunikaty, a także współpracować z innymi agentami lub ze wspólnym stanem globalnym systemu.
- **Benchmark** - zestaw metryk służących do oceny jakości działania modeli LLM. Umożliwiają one porównanie wydajności, spójności i trafności generowanych odpowiedzi w sposób obiektywny i powtarzalny
- **Hive Minds / Umysły rojowe** - koncepcja inspirowana systemami biologicznymi, w której wiele niezależnych jednostek współpracuje w ramach wspólnej przestrzeni umożliwiającej wymianę i synchronizację informacji
- **LLM** - generatywny model językowy oparty na (zwykle dekoderowej) architekturze transformera, posiadający miliardy parametrów. Modele te trenowane są na bardzo dużych zbiorach danych testowych i zdolne są do przetwarzania i generowania języka naturalnego
- **Model** - instancja LLM zdolna do przetwarzania tekstu i generowania odpowiedzi na podstawie dostarczonych promptów. Generuje tekst i przetwarza dane językowe, ale sam nie posiada trwałego stanu ani autonomii decyzyjnej
- **Prompt** - tekst przekazywany do modelu LLM w celu wywołania odpowiedzi. Może on zawierać m.in. instrukcje, pytania, przykłady a jego konstrukcja bezpośrednio wpływa na jakość generowanych wyników
- **Aplikacja konsolowa** - program działający w konsoli, posiadający wyłącznie interfejs tekstowy umożliwiający interakcję z użytkownikiem

2 Dokumentacja techniczna projektu

2.1 Cele i założenia projektu

Celem projektu jest implementacja systemu, który przy pomocy współpracujących ze sobą modeli LLM będzie rozwiązywał problemy matematyczne. Otrzymywane wyniki powinny być lepsze, niż osiągnięte przez pojedynczy model LLM

2.2 Cechy funkcjonalne produktu

- Komunikacja między modelami nie odbywa się w formie dialogu między nimi
- Prompt wprowadzany jest przez użytkownika za pośrednictwem konsoli

2.3 Wykorzystane technologie i narzędzia

- **Python** - język programowania wykorzystany do implementacji logiki i funkcjonalności aplikacji
- **Ollama** - oprogramowanie wykorzystane na potrzeby lokalnego uruchamiania i obsługi wybranych modeli LLM
 - **Model *llama3.1:8b*** - model LLM (8 miliardów parametrów) wykorzystany do obsługi agentów generujących odpowiedzi na prompty
- **Github** - środowisko kontroli wersji wykorzystane do współdzielenia i przechowywania kodu źródłowego i dokumentacji projektowej

2.4 Oprogramowanie

2.4.1 Agenci

Tabela 2.1. Charakterystyka agentów

Lp.	Ilość	Rola	Modele LLM	Opis zleconej roli
1.	1	badacz (researcher)	▪ llama3.1:8bl	Badacz zbierający tylko faktyczne, nierachunkowe informacje na temat danego problemu matematycznego. Głównym celem agenta jest przygotowanie wartościowych informacji w oparciu o prace naukowe, badania, internet itp.
2.	1	kalkulator (calculator)	▪ llama3.1:8bl	Deterministyczny specjalista matematyczny obliczający zadania
3.	1	ewaluator (evaluator)	▪ llama3.1:8bl	Ewaluator otrzymujący wiele wyników dla danego problemu matematycznego. Zadaniem agenta jest ocenienie, która spośród podanych odpowiedzi jest poprawna

2.4.2 Generowanie i ewaluacja wyniku

Program ma celu wyznaczenie końcowej wartości wyniku dla zadanego przez użytkownika problemu matematycznego. Proces ten opiera się na kilkukrotnym generowaniu niezależnych wyników obliczeń i porównaniu ich w celu wyboru najbardziej wiarygodnej wartości.

Proces wyznaczania wyniku rozpoczyna się od zebrania wiedzy kontekstowej (agent *researcher*). Badacz ma za zadanie zebrać prawdziwe, faktyczne informacje dotyczące wskazanego problemu, podać potrzebne wzory matematyczne, wskazać przydatne algorytmy. Nie wykonuje żadnych obliczeń, jedynie wysyła wiedzę.

Następnie przeprowadzana jest seria obliczeń (agent *calculator*) polegająca na kilkukrotnym niezależnym wywołaniu modelu. Każde z tych wywołań realizowane jest z małą, losową (0.03 - 0.01) temperaturą co pozwala na zróżnicowanie otrzymanych odpowiedzi bez jednoczesnej destabilizacji obliczeń i podatności na halucynacje. Wyniki trzymane są we wspólnie dzielonym stringu wraz z zebranymi informacjami i promptem od użytkownika. Każda kolejna odpowiedź jest do niego doklejana.

Otrzymane wyniki (współdzielony string) są dalej przekazywane do ewaluacji (agent *evaluator*). Na tym etapie nie wykonywane są obliczenia. Ewaluacja polega na:

- identyfikacji zbliżonych wyników
- odrzuceniu wartości odstających
- wyborze odpowiedzi najlepiej reprezentującej otrzymane wyniki

Jeśli po weryfikacji evaluator stwierdzi, że żaden wynik nie jest satysfakcyjny, uruchamiany jest ponownie nowy kalkulator z dołączonym feedbackiem o słabych wynikach. Czynność jest powtarzana tak długo, aż evaluator potwierdzi dobry wynik lub przekroczy zostanie ustawiony limit możliwych wywołań.

2.5 Testy

- Zgodność formatu zwracanych odpowiedzi z podanym formatem JSON
- Odporność systemu na błędne odpowiedzi
- Poprawność wyniku końcowego
- Poprawność komunikacji z systemem Ollama i modelami LLM

2.5.1 Przykładowe prompty

- Find a rational approximation of π with denominator less than 1,000
- Find the length of the hypotenuse of a right triangle with legs 3 and 4
- Let ABCD be a regular tetrahedron with side length 1. Let $u = \text{vector AB}$ and $v = \text{vector AC}$. Compute the exact value of $\|u + v\|$

2.5.2 Kryteria poprawności

- Zbieżne i poprawne wyniki zwracane podczas większości wywołań systemu
- Poprawne przerwanie systemu w przypadku wystąpienia zbyt dużej ilości błędnych odpowiedzi

2.6 Wyniki działania

- Program zwraca pojedynczą wartość liczbową proponowaną przez ewaluatora, którą wyselekcjonował z możliwych rezultatów zaprezentowanych przez agentów kalkulatorów.
- Jeśli ewaluator nie znajdzie satysfakcyjnej odpowiedzi, a przekroczony zostanie limit wywołań kalkulatorów – program kończy się zwracając błąd.
- Zdarza się, że dany agent obliczeniowy zaproponuje niepoprawny wynik, dlatego potrzebna jest lista kilku propozycji
- Dla zapytania: “Find a rational approximation of π with denominator less than 1,000”, program zwraca: 355/113
- Dla zapytania: “Find the length of the hypotenuse of a right triangle with legs 3 and 4”, program zwraca: 5
- Dla zapytania: “Let ABCD be a regular tetrahedron with side length 1. Let $u = \text{vector AB}$ and $v = \text{vector AC}$. Compute the exact value of $\|u + v\|$ ”, program zwraca: sqrt(3).

3 Załączniki

Nie dotyczy