Duże modele językowe w systemie wspomagania decyzji

Maksim Makaranka
Instytut Informatyki
Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska
Warszawa, Polska
maksim.makaranka.stud@pw.edu.pl

***Streszczenie*—Duże modele językowe (LLM) znajdują zastosowanie w systemach wspomagania decyzji, lecz ich wykorzystanie ogranicza brak mechanizmów szacowania ryzyka, trudności w uwzględnianiu priorytetów użytkownika oraz niska transparentność. Przedstawiam podejście DeLLMa, łączące klasyczną teorię decyzji z LLM, umożliwiające modelowanie niepewności, prognozowanie stanów natury i budowę funkcji użyteczności. Proponuję rozszerzenia DeLLMa: automatyczne uzupełnianie przestrzeni akcji na podstawie zewnętrznych źródeł oraz zaawansowaną agregację danych internetowych. Omawiam architekturę prototypowego, wieloagentowego systemu decyzyjnego oraz kierunki dalszego rozwoju w celu zwiększenia skuteczności i wyjaśnialności rekomendacji generowanych przez LLM.**

***Index Terms*—LLM, DeLLMa, podejmowanie decyzji, teoria decyzji, niepewność, systemy wieloagentowe, agregacja danych, explainable AI**

I. Wprowadzenie

Duże modele językowe (LLM, ang. Large Language Models) rewolucjonizują analizę tekstu, automatyczne podsumowywanie dokumentów czy generowanie raportów biznesowych. Dzięki zdolności do przetwarzania ogromnych zbiorów danych i generowania rekomendacji zbliżonych do ludzkiego rozumowania, LLM są atrakcyjnym narzędziem w zadaniach wymagających podejmowania decyzji pod presją ryzyka i niepewności.

Jednak klasyczne LLM mają istotne ograniczenia w kontekście decyzji: opierają się głównie na statystycznym dopasowywaniu wzorców, nie posiadają mechanizmów szacowania ryzyka, oceny niepewności czy równoważenia celów użytkownika. Działają często jak „czarna skrzynka", co utrudnia wyjaśnienie procesu rekomendacji i obniża zaufanie użytkowników w sytuacjach wymagających przejrzystości.

W artykule przedstawiam nowoczesne rozwiązania tych problemów, koncentrując się na frameworku DeLLMa, który łączy klasyczną teorię decyzji z potencjałem LLM. Framework ten umożliwia modelowanie niepewności, prognozowanie scenariuszy i budowanie funkcji użyteczności na podstawie preferencji użytkownika. Opisuję także własne rozszerzenia tej koncepcji, obejmujące automatyczne poszerzanie przestrzeni akcji oraz zaawansowaną agregację danych, co pozwala na budowę transparentnych i skutecznych systemów wspomagania decyzji w warunkach wysokiej niepewności.

II. LLM w zadaniach decyzyjnych – ograniczenia klasycznych podejść

Współczesne LLM, takie jak GPT-4.1, osiągają wysoką skuteczność w analizie tekstu czy podsumowywaniu dokumentów, bazując na statystycznym przewidywaniu kolejnych tokenów. W takich zadaniach nie wymagają głębokiego rozumienia kontekstu.

W zastosowaniach decyzyjnych, gdzie kluczowe są analiza ryzyka, niepewności i optymalizacja względem preferencji użytkownika, klasyczne LLM napotykają poważne ograniczenia. Nawet techniki takie jak Retrieval-Augmented Generation (RAG), które wzbogacają kontekst o aktualne dane, często prowadzą do powierzchownych, nadmiernie pewnych siebie odpowiedzi nieuwzględniających niejednoznaczności [1]. Brak kalibracji skutkuje zbyt dużym zaufaniem modelu do własnych rekomendacji, nawet przy niepełnych lub sprzecznych danych [2].

W odpowiedzi powstały modele rozumujące, np. OpenAI o4 czy DeepSeek R1, wykorzystujące technikę Chain-of-Thought (CoT) [3], która rozbija proces myślowy na sekwencję logicznych kroków, zwiększając transparentność rozwiązywania problemów.

Trening takich modeli obejmuje Supervised Fine-Tuning (SFT) na przykładach poprawnych rozumowań oraz uczenie ze wzmocnieniem (RL), zwłaszcza RLHF (Reinforcement Learning from Human Feedback). W RLHF kluczowy jest model krytyka (reward model), trenowany na ocenach ludzkich, który następnie ocenia odpowiedzi LLM, a model jest optymalizowany pod kątem maksymalizacji przewidywanej „nagrody". Pozwala to lepiej dostosować LLM do ludzkich preferencji, ale nie rozwiązuje problemów szacowania ryzyka i niepewności.

Nawet rozumujące LLM nie eliminują kluczowych ograniczeń w zadaniach decyzyjnych:

- Brak kompleksowego szacowania ryzyka i wyważania konkurencyjnych celów użytkownika – modele nie równoważą efektywnie potencjalnych strat i zysków [4].
- Brak natywnych mechanizmów probabilistycznych do prognozowania scenariuszy i przypisywania im prawdopodobieństw.
- Nadmierna ostrożność – modele unikają ryzykownych decyzji, co prowadzi do zachowawczych rekomendacji.

- Problemy ze skalowalnością przy dużej liczbie możliwych akcji – nawet CoT staje się niewydajny przy wielu wariantach.
- Ograniczona przejrzystość i wyjaśnialność procesu decyzyjnego, co utrudnia zaufanie do systemu.

W efekcie klasyczne i reasoning-centric LLM nie spełniają wymagań zaawansowanych systemów wspomagania decyzji. Potrzebne są nowe podejścia, umożliwiające modelowanie niepewności, wyważanie ryzyka i transparentność – przykładem jest framework DeLLMa, omówiony w kolejnych sekcjach.

III. Framework DeLLMa – integracja teorii decyzji i LLM

Framework DeLLMa (Decision-making Large Language Model assistant) [5] stanowi uniwersalne narzędzie wspomagania decyzji, łączące klasyczną teorię decyzji z możliwościami dużych modeli językowych. Jego celem jest umożliwienie przejrzystego i wyjaśnialnego podejmowania decyzji w warunkach niepewności, przy wykorzystaniu zarówno wiedzy eksperckiej, jak i informacji zawartych w promptach użytkownika.

A. Formalizacja problemu decyzyjnego

W DeLLMa każda sytuacja decyzyjna opisywana jest przez tzw. user prompt, formalnie oznaczony jako:

$$\mathcal{P} = (\mathcal{G}, \mathcal{A}, \mathcal{C})$$

gdzie:

- $\mathcal{G}$ – cel użytkownika,
- $\mathcal{A}$ – lista dostępnych akcji,
- $\mathcal{C}$ – kontekst decyzyjny, obejmujący raporty, dane historyczne lub inne istotne informacje.

Kluczowym elementem formalizacji jest modelowanie niepewności poprzez wprowadzenie przestrzeni stanów natury Θ. Każdy stan $\theta \in \Theta$ reprezentuje konkretną kombinację wartości ukrytych czynników, które mogą mieć wpływ na wynik decyzji, a których wartości nie są znane w momencie podejmowania decyzji. Przestrzeń stanów Θ jest dyskretyzowana przez k ukrytych czynników $(f_1, ..., f_k)$, z których każdy może przyjmować ℓ możliwych wartości. W efekcie liczba wszystkich możliwych stanów wynosi $|\Theta| = \ell^k$.

Funkcja użyteczności, oznaczona jako

$$U : \Theta \times \mathcal{A} \to \mathbb{R},$$

przypisuje wartość liczbową dowolnej parze stan-akcja (θ, a), oceniając, na ile preferowany jest wynik wynikający z tej pary. Celem jest wybór akcji $a^* \in \mathcal{A}$, która maksymalizuje oczekiwaną użyteczność, uwzględniając niepewność związaną z nieznanymi stanami $\theta \in \Theta$.

B. Podstawowa procedura decyzyjna DeLLMa

Poniżej przedstawiono podstawowe kroki procedury decyzyjnej DeLLMa:

Krok 1: Enumeracja stanów natury: Celem tego kroku jest stworzenie pełnej listy możliwych stanów natury $\Theta = (\theta_1, \ldots, \theta_m)$, które są nieznanymi wielkościami przewidywanymi jako mające wpływ na realizację celu użytkownika $\mathcal{G}$. Proces ten polega na identyfikacji i dyskretyzacji ukrytych czynników, które opisują przestrzeń stanów, umożliwiając prognozowanie różnych scenariuszy decyzyjnych w ramach zadania decyzyjnego. Na początku duży model językowy analizuje prompt $\mathcal{P}$, aby zidentyfikować zestaw kluczowych czynników ukrytych, które mogą wpływać na cel użytkownika. Następnie dla każdego zidentyfikowanego czynnika model generuje zestaw możliwych wartości, które mogą symbolizować różnorodność rzeczywistości w tym kontekście. Te wartości są używane do dyskretyzacji przestrzeni stanów, tworząc różne kombinacje reprezentujące szeroki wachlarz możliwych rezultatów.

Matematyczny opis procedury polega na tym, że z kontekstu $\mathcal{P}$ LLM identyfikuje k ukrytych czynników, które mogą wpływać na cel użytkownika $\mathcal{G}$, oznaczanych jako $(f_1, \ldots, f_k)$. Dla każdego ukrytego czynnika f_j, model generuje ℓ możliwych wartości, które są oznaczane jako $\tilde{f}_j^{1:\ell}$. Te wartości, przedstawione jako ciągi znaków (słowa bądź frazy), dyskretyzują przestrzeń stanów, a każdy stan θ_j w przestrzeni Θ stanowi kombinację jednej wartości z każdego z k czynników, co można wyrazić jako:

$$\theta = (\tilde{f}_1^{i_1}, \ldots, \tilde{f}_k^{i_k}), \quad i_j = 1, \ldots, \ell.$$

Łączna liczba możliwych stanów wynosi $|\Theta| = m = \ell^k$, co skutkuje bardzo rozbudowaną przestrzenią, którą należy rozpatrzyć. Pomimo dużej liczby stanów, w kolejnych krokach przedstawiona zostanie procedura prognozowania prawdopodobieństw tych stanów w sposób skalowalny, co umożliwia praktyczne zastosowanie w rzeczywistych scenariuszach decyzyjnych.

Krok 2: Prognozowanie prawdopodobieństw stanów: W tym etapie prognozowane są prawdopodobieństwa wystąpienia poszczególnych stanów θ, biorąc pod uwagę kontekst $\mathcal{C}$.

Stosowana jest do tego prosta metoda, która jest przedstawiona tutaj i potencjalnie może być celem do ulepszenia w ramach dalszej pracy. Dla każdego z k ukrytych czynników oraz każdej z ℓ ich możliwych wartości $\tilde{f}_j^{1:\ell}$, LLM przypisuje werbalne oceny prawdopodobieństwa, takie jak *bardzo prawdopodobne*, *prawdopodobne*, *raczej prawdopodobne*, *raczej mało prawdopodobne*, *mało prawdopodobne*, *bardzo mało prawdopodobne*. Każda z tych ocen jest następnie zamieniana na wartość liczbową przy użyciu słownika $\mathcal{V}$. Określone w ten sposób prawdopodobieństwo dla poszczególnej wartości czynnika ukrytego $\tilde{f}_i$, uwzględniając kontekst C, definiujemy jako $\pi_i(\tilde{f}_i \mid C)$.

Dla uproszczenia zakładamy, że czynniki są wzajemnie niezależne, co oznacza, że prawdopodobieństwo całego stanu θ_j to iloczyn „marginalnych" rozkładów $\pi(\cdot \mid C)$. W efekcie otrzymujemy $\pi^{LLM}(\theta_j \mid \mathcal{C})$ – przybliżony rozkład posteriori, który mówi, jak bardzo LLM „wierzy" w wystąpienie danego wariantu rzeczywistości θ_j.

Po normalizacji, rozkład prawdopodobieństwa dla stanu θ_j wyraża się jako:

$$\pi^{LLM}(\theta_j \mid \mathcal{C}) = \prod_{i=1}^{k} \pi_i(\cdot \mid C),$$

gdzie π_i jest rozkładem prawdopodobieństwa dla poszczególnego czynnika określonym na podstawie kontekstu C. Pomimo upraszczającego założenia o niezależności ukrytych czynników, ta procedura prognozowania wspiera prognozy stanów, które mogą być wykorzystywane w różnych scenariuszach decyzyjnych, co zapewnia solidne podstawy do dalszej analizy.

Istnieją badania, które pokazują, że duże modele językowe są zdolne do generowania dobrze skalibrowanych prognoz na podstawie dostarczonych informacji, szczególnie poprzez nadanie prawdopodobieństw werbalnych [2], [6].

Krok 3: Elicytacja funkcji użyteczności: Na tym etapie dokonujemy elicytacji, czyli skonstruowania, funkcji użyteczności $U : \Theta \times \mathcal{A} \to \mathbb{R}$, przypisującej wartość każdej parze stan-akcja (θ_j, a_i). Funkcja ta ma na celu odzwierciedlenie preferencji użytkownika względem zdefiniowanego celu $\mathcal{G}$. W tym kroku połączono klasyczne metody elicytacji z możliwościami dużych modeli językowych, aby automatycznie i efektywnie skonstruować funkcję użyteczności.

Proces rozpoczyna się od próbkowania stanów z rozkładu prognostycznego $\pi^{\text{LLM}}(\theta \mid \mathcal{C})$ oraz tworzenia zestawu par stan-akcja dla wszystkich $a \in \mathcal{A}$. Następnie pary te są grupowane w minibatche, które przekazywane są do LLM w celu nadania im rang według celu użytkownika $\mathcal{G}$. Ranking przedmiotów na podstawie LLM, gdzie każdy przedmiot składa się z akcji i konkretnej instancji stanu, to procedura o szerokim zastosowaniu, a LLM-y były wcześniej skutecznie wykorzystywane do podobnych porównań [7]. Na podstawie uzyskanych rankingów wyciągane są preferencje parowe, które wykorzystywane są w klasycznych algorytmach elicytacji użyteczności, takich jak model Bradley-Terry.

W oryginalnym artykułu DeLLMa autorzy opisują dwie strategie formatowania rankingów: $\mathrm{Rank2Pairs}$ oraz $\mathrm{One\text{-}vs\text{-}All}$. Strategia $\mathrm{Rank2Pairs}$ konwertuje ranking o malejącej preferencji $\mathcal{R} = \big((\theta, a)_{(1)}, \cdots, (\theta, a)_{(b)}\big)$ na listę porównań, dodając $(\theta, a)_{(i)} \succ (\theta, a)_{(j)}$ dla każdej pary, gdy $i < j$. Z kolei $\mathrm{One\text{-}vs\text{-}All}$ zakłada, że LLM jest obojętny wobec wszystkich par stan-akcja poza najwyżej ocenioną, czyli $\big\{(\theta, a)_{(1)} \succ (\theta, a)_{(i)} \mid \forall\ 2 \leq i \leq b\big\}$. Tę strategię można preferować, gdy dokładne porównania mniej optymalnych par stan-akcja są trudne lub niepewne.

Aby zwiększyć efektywność i skalowalność procesu elicytacji, autorzy proponują wykorzystanie technik *batchingu* oraz *redukcji wariancji*. W batchingu próbki par stan-akcja $S_A = \{(\theta, a)\}$ są dzielone na nakładające się na siebie minibatche, z częścią próbek ($q\%$) wspólną dla kolejnych minibatchów, co pozwala na bardziej precyzyjną i spójną elicytację preferencji. Redukcja wariancji polega natomiast na próbkowaniu niezależnych wartości stanu oraz tworzeniu ich duplikatów dla każdej możliwej akcji, co umożliwia uzyskanie bardziej stabilnych i wiarygodnych oszacowań funkcji użyteczności.

Krok 4: Maksymalizacja oczekiwanej użyteczności: W tej końcowej fazie procesu obliczana jest oczekiwana użyteczność dla każdej możliwej akcji, a następnie wybierana jest ta, która maksymalizuje oczekiwane korzyści. Dla każdej akcji stosuje się oszacowanie oczekiwanej użyteczności metodą Monte Carlo, bazując na próbkach par stan-akcja pobranych z prognostycznego rozkładu stanów $\pi^{\text{LLM}}(\theta \mid \mathcal{C})$ oraz uzyskanej funkcji użyteczności. Obliczenia te przeprowadza się analitycznie, bez bezpośredniego wykorzystania LLM. Oczekiwaną użyteczność $U_{\mathcal{C}}(a)$ można przybliżyć jako:

$$U_{\mathcal{C}}(a) \approx \frac{1}{|S|} \sum_{\theta \in S} U(\theta, a),$$

gdzie $S \subseteq \Theta$ to zbiór próbek stanów wylosowanych z rozkładu prognozowanego przez LLM, który jest przybliżeniem rozkładu post-priori o stanach, uwzględniając kontekst $\mathcal{C}$, czyli $S \overset{\text{i.i.d.}}{\sim} \pi^{\text{LLM}}(\theta \mid \mathcal{C}) \approx \pi(\theta \mid \mathcal{C})$. Po obliczeniu oczekiwanej użyteczności $U_{\mathcal{C}}(a)$ dla każdej akcji wybiera się ostateczną decyzję poprzez maksymalizację:

$$a^* = \mathrm{argmax}_{a \in \mathcal{A}} U_{\mathcal{C}}(a).$$

Ten wybór zapewnia, że podejmowana decyzja maksymalizuje potencjalne korzyści, biorąc pod uwagę prognozowane prawdopodobieństwa stanów w kontekstach, które są analizowane.

C. Podsumowanie

Framework DeLLMa stanowi nowoczesną propozycję integrującą klasyczną teorię decyzji z możliwościami LLM. Pozwala na skalowalne modelowanie niepewności, elicytację preferencji użytkownika oraz wyjaśnialność procesu decyzyjnego, niezależnie od złożoności analizowanego problemu.

IV. Własne adaptacje i rozszerzenia DeLLMa

W celu zwiększenia elastyczności i praktycznej użyteczności frameworku DeLLMa, opracowałem szereg rozszerzeń, które pozwalają na lepsze dostosowanie systemu do rzeczywistych, bardziej złożonych scenariuszy decyzyjnych. Poniżej przedstawiam kluczowe z nich.

A. Automatyczne uzupełnianie przestrzeni akcji

W praktycznych zastosowaniach użytkownicy często nie są w stanie wyczerpująco wskazać wszystkich potencjalnie sensownych akcji prowadzących do osiągnięcia celu decyzyjnego. Wynika to z ograniczonej wiedzy domenowej, braku dostępu do aktualnych informacji lub faktu, że część skutecznych rozwiązań została już wcześniej zastosowana w innych, podobnych problemach, lecz nie jest szeroko znana.

Aby zniwelować to ograniczenie, system wyposażono w moduł automatycznego uzupełniania pierwotnego zbioru akcji $\mathcal{A}_0$. Moduł ten analizuje dostępne źródła wiedzy zewnętrznej i identyfikuje propozycje akcji stosowanych lub rekomendowanych w analogicznych sytuacjach. Sugerowane akcje są następnie oceniane pod kątem ich przydatności i zgodności z celem użytkownika oraz kontekstem decyzyjnym. W efekcie, początkowy zbiór $\mathcal{A}_0$ może zostać rozszerzony do pełniejszego

zbioru $\mathcal{A}$, obejmującego zarówno pierwotne, jak i nowo zidentyfikowane możliwości.

Istotnym elementem tego procesu jest elastyczność: użytkownik decyduje, czy system ma automatycznie rozszerzać zbiór $\mathcal{A}_0$ o propozycje z zewnętrznych źródeł. W praktyce możliwe jest także wdrożenie etapu interakcji, w którym użytkownik otrzymuje listę nowych akcji i wskazuje, które z nich mają zostać uwzględnione w dalszej analizie. Takie podejście pozwala na dynamiczne rozszerzanie przestrzeni decyzyjnej o nieoczywiste, ale potencjalnie wartościowe opcje, przy jednoczesnym zachowaniu kontroli nad ostatecznym kształtem zbioru analizowanych akcji.

Automatyczne uzupełnianie przestrzeni akcji jest szczególnie przydatne w zadaniach o wysokim stopniu niepewności lub w nowych, słabo opisanych domenach, gdzie tradycyjne podejście oparte wyłącznie na wiedzy użytkownika może prowadzić do pominięcia istotnych możliwości decyzyjnych.

B. Dyskretyzacja i uzupełnianie kontekstów stanów

Aby umożliwić bardziej precyzyjne prognozowanie stanów natury, wprowadzam mechanizm automatycznej dyskretyzacji i wzbogacania kontekstów dla każdego czynnika ukrytego f_i. Zamiast korzystać wyłącznie z ogólnego kontekstu $\mathcal{C}_0$, dostarczonego przez użytkownika, system generuje dedykowany kontekst c_i dla każdego czynnika, zgodnie z poniższą procedurą:

1) **Dyskretyzacja:** LLM analizuje $\mathcal{C}_0$ i wyodrębnia fragmenty najbardziej powiązane z danym czynnikiem f_i.
2) **Wzbogacanie:** Dla każdego czynnika system wyszukuje dodatkowe dane w źródłach zewnętrznych.
3) **Tworzenie kontekstów:** Powstałe konteksty $C = (c_1, ..., c_k)$ agregują zarówno informacje wyodrębnione z $\mathcal{C}_0$, jak i nowe, specyficzne dane dotyczące danego czynnika.

Tak uzupełnione, dedykowane konteksty c_i pozwalają LLM na znacznie precyzyjniejsze prognozowanie prawdopodobieństw wartości $\pi_i(\tilde{f}_i \mid c_i)$ dla każdego czynnika, co przekłada się na bardziej wiarygodny rozkład posteriori:

$$\pi^{LLM}(\theta_j \mid \mathcal{C}) = \prod_{i=1}^{k} \pi_i(\cdot \mid c_i)$$

Dzięki temu możliwe jest lepsze wykorzystanie dostępnych danych i zredukowanie niepewności w modelu decyzyjnym.

C. Dynamiczna liczba wartości czynników

W odróżnieniu od klasycznego DeLLMa, który zakłada jednakową liczbę wartości dla każdego czynnika ukrytego, w mojej adaptacji LLM samodzielnie decyduje, ile wartości l_i należy wygenerować dla każdego czynnika f_i, przy założeniu górnego ograniczenia $l_i \leq L$.

Przestrzeń stanów Θ jest budowana wtedy jako iloczyn kartezjański:

$$\Theta = \prod_{i=1}^{k} \{\tilde{f}_i^1, ..., \tilde{f}_i^{l_i}\}$$

Takie podejście umożliwia efektywne modelowanie zarówno czynników o dużej zmienności, jak i tych, które w praktyce przyjmują niewiele możliwych wartości, bez nadmiernego rozbudowywania przestrzeni stanów.

D. Wybór zestawu akcji

W wielu rzeczywistych problemach decyzyjnych optymalnym rozwiązaniem nie jest pojedyncza akcja, lecz zestaw akcji, które mogą być realizowane równolegle lub wzajemnie się uzupełniać i wzmacniać efekt końcowy. Przykłady takich sytuacji obejmują zarządzanie projektami, planowanie strategiczne czy działania optymalizacyjne, gdzie synergia kilku działań daje lepszy rezultat niż każda z nich osobno.

Na obecnym etapie dokładna metoda wyboru zestawu akcji nie została jeszcze ostatecznie ustalona. Jako możliwą opcję rozważam rozszerzenie procedury maksymalizacji oczekiwanej użyteczności na wybór zbioru akcji $A^* \subseteq \mathcal{A}$, według następującego schematu:

1) Wyznaczana jest akcja a^* maksymalizująca oczekiwaną użyteczność $U_{\mathcal{C}}(a)$.
2) Następnie rozważane są kolejne akcje a', które nie kolidują z a^* (tj. mogą być wykonane równocześnie) i mają wysoką oczekiwaną użyteczność.
3) Zbiór optymalnych akcji A^* budowany jest iteracyjnie, aż do wyczerpania dostępnych, współwykonywalnych akcji spełniających kryterium użyteczności:

$$A^* = \{a^*\} \cup \{a' \in \mathcal{A} \setminus \{a^*\} : U_{\mathcal{C}}(a') \geq \tau \wedge a' \text{nie koliduje z } a^*\}$$

gdzie τ to ustalony próg użyteczności.

Rozważane są również inne strategie, które mogą uwzględniać złożone zależności i ograniczenia między akcjami. Takie rozszerzenie pozwoli na rekomendowanie synergicznych zestawów działań, co jest szczególnie istotne w zadaniach, gdzie potencjalne korzyści wynikają z łączenia kilku komplementarnych akcji.

V. Agregator danych internetowych – architektura i zastosowanie

W celu zwiększenia aktualności i kompletności informacji wykorzystywanych przez system decyzyjny, opracowałem własny silnik agregacji danych internetowych. Rozwiązanie to pozwala na automatyczne pozyskiwanie i selekcję wiedzy z szerokiego spektrum źródeł, co przekłada się na bardziej trafne rekomendacje oraz skuteczniejsze uzupełnianie zarówno przestrzeni akcji, jak i kontekstów czynników ukrytych.

Proces agregacji obejmuje następujące etapy:

1) **Wyszukiwanie źródeł:** Google API służy do pobierania listy stron powiązanych z zapytaniem użytkownika.
2) **Pobieranie treści:** Selenium (undetected-chromedriver) umożliwia równoległe otwieranie i pobieranie zawartości stron, minimalizując wpływ zabezpieczeń antybotowych.
3) **Ekstrakcja i segmentacja:** BeautifulSoup wydobywa tekst, który następnie dzielony jest na krótkie fragmenty.

4) **Wektoryzacja i selekcja:** Zarówno fragmenty, jak i zapytanie użytkownika są zamieniane na reprezentacje wektorowe przy użyciu sentence-transformers (all-MiniLM-L6-v2). Na tej podstawie wyliczane jest podobieństwo cosinusowe, co pozwala odrzucić fragmenty nieistotne semantycznie.
5) **Agregacja i podsumowanie:** Najtrafniejsze fragmenty przekazywane są do większego LLM (np. GPT-4.1-nano), który generuje syntetyczną odpowiedź oraz dołącza listę źródeł, zwiększając transparentność procesu.

Silnik agregacji znajduje zastosowanie w uzupełnianiu przestrzeni akcji (identyfikacja nowych opcji decyzyjnych), wzbogacaniu kontekstów czynników (pozyskiwanie aktualnych danych branżowych, raportów, statystyk) oraz – eksperymentalnie – w pozyskiwaniu informacji o potencjalnych ryzykach na etapie enumeracji stanów. Rozwiązanie to pozwala na automatyczne i skalowalne włączanie wiedzy zewnętrznej do procesu decyzyjnego.

VI. Implementacja prototypu i plany rozwoju

Stworzony prototyp systemu decyzyjnego oparty jest na architekturze wieloagentowej, stworzonej przy użyciu frameworku LangGraph. Takie podejście umożliwia podział odpowiedzialności na niezależne, wyspecjalizowane komponenty (agenty), co zwiększa elastyczność i skalowalność rozwiązania. Na obecnym etapie zaimplementowane zostały następujące elementy:

- Agent walidacji i dekompozycji user promptu – odpowiada za wstępną analizę i rozbicie zapytania użytkownika na komponenty decyzyjne.
- Agent enumeracji stanów – odpowiada za budowę przestrzeni stanów natury na podstawie promptu i kontekstu.
- Agent uzupełnienia przestrzeni akcji – wzbogaca zbiór akcji o propozycje zidentyfikowane przez agregator danych.
- Agent ulepszenia kontekstu – (w trakcie) zajmuje się dyskretyzacją i wzbogacaniem kontekstów czynników.

Kolejne etapy rozwoju mogą obejmować:

- Rozbudowę własnego frameworku decyzyjnego na bazie DeLLMa, w tym integrację kolejnych adaptacji i usprawnień.
- Porównanie skuteczności różnych podejść: klasycznego DeLLMa, czystych LLM oraz hybrydowych rozwiązań z zaawansowaną agregacją danych (lub innych rozwiązań opisywanych w literaturze).
- Opracowanie dobrych praktyk implementacyjnych dla systemów decyzyjnych opartych o LLM, wyszukiwarek internetowych oraz architektur wieloagentowych.

Podsumowanie

Duże modele językowe, choć stanowią potężne narzędzie w analizie danych i generowaniu rekomendacji, wymagają odpowiedniej integracji z klasycznymi metodami modelowania niepewności oraz systematycznego pozyskiwania wiedzy zewnętrznej, aby sprostać wyzwaniom realnych zadań decyzyjnych. Framework DeLLMa, wzbogacony o autorskie adaptacje, pozwala na budowę przejrzystych, elastycznych i wysoce skutecznych systemów wspomagania decyzji. Takie rozwiązania otwierają nowe perspektywy dla praktycznych zastosowań LLM w obszarach wymagających zarówno wysokiej jakości rekomendacji, jak i pełnej wyjaśnialności procesu decyzyjnego.

Literatura

[1] Chenglei Si, Chen Zhao, Sewon Min, and Jordan Boyd-Graber. Re-examining calibration: The case of question answering. https://arxiv.org/abs/2205.12507v2

[2] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. https://arxiv.org/abs/2402.02392

[3] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, Xiang Yue. Demystifying Long Chain-of-Thought Reasoning in LLMs. https://arxiv.org/abs/2502.03373

[4] Mykel J Kochenderfer. Decision making under uncertainty: theory and application. MIT press, 2015.

[5] Ollie Liu, Deqing Fu, Dani Yogatama, and Willie Neiswanger. DeLLMa: Decision Making Under Uncertainty with Large Language Models. https://arxiv.org/abs/2402.02392

[6] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? An empirical evaluation of confidence elicitation in llms. https://arxiv.org/abs/2306.13063

[7] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. Large language models are effective text rankers with pairwise ranking prompting, 2023. https://arxiv.org/abs/2306.17563