Akademia Ekonomiczno-Humanistyczna

Wydział Technologii Informatycznych

Kierunek: **Informatyka**

Praca Dyplomowa Magisterska

Wielkie Modele Językowe jako narzędzie wspomagające człowieka

Krzysztof Małczak

Opiekun pracy
dr inż. Krzysztof Rychlicki-Kicior

Słowa kluczowe: Generative AI, Model Assesment, Large Language Models, LLM Evaluation, LLM Benchmark

Warszawa 2025

Spis treści

Spis rysunków

Spis tabel

1.3 Cel pracy

Celem pracy jest zwiększenie świadomości społecznej w kwestii tego czy Wielkie Modele Językowe w praktyce są w stanie wspomagać codzienne funkcjonowanie człowieka poprzez pomoc w procesach decyzyjnych oraz rozwiązywaniu takich problemów jak priorytetyzacja zadań, organizacja czasu czy wyciąganie wniosków.

Praca skupia się na przeprowadzeniu eksperymentów na specjalnie przygotowanym zbiorze danych, analizie metryk oraz porównaniu wyników dla wybranych, najpopularniejszych modeli.

Rozdział 2

Wprowadzenie do Prompt Engineeringu

Ten rozdział skupia się na wprowadzeniu pojęć z dziedziny prompt engineeringu niezbędnych do dogłębnego zrozumienia części praktycznej niniejszej pracy. Każdy przykład promptu i opcjonalnego wyniku przedstawiony jest jak na rysunku 2.1, przy czym kategorie i rodzaje promptów omawiane są w podrozdziale 2.4.

PROMPT:
[*Kategoria, Rodzaj*]

TREŚĆ PROMPTU

OUTPUT (opcjonalny):

TREŚĆ ODPOWIEDZI

Rysunek 2.1: Opis

2.1 Prompt

Prompt to pojęcie ogólnie odnoszące się do treści podawanej jako wejście do Modelu Generatywnej Sztucznej Inteligencji [50, 58], która ma na celu sterować jego działaniem. W zależności od zastosowania może przyjmować różne formy zaczynając od prostego pytania a kończąc na szczegółowym opisie konkretnego zadania (np. opis obrazu, który ma zostać wygenerowany) [4]. Może składać się z tekstu, obrazu, dźwięku lub innych mediów [58] w zależności od modelu.

W kontekście Wielkich Modeli Językowych pojęcie promptu zazwyczaj odnosi się do tekstu w formie stwierdzenia, pytania lub zestawu instrukcji mających na celu wywołanie określonej odpowiedzi. Na rysunku 2.2 przedstawiono jeden z najbardziej podstawowych przykładów promptu, zaś rysunek 2.3 przedstawia tzw. prompt template [63] (ang. template - szablon), w którym zmienne umieszcza się w nawiasach klamrowych - jest to zapis często używany w celu uproszczenia rzeczywistej zawartości promptu.

2.2 Komponenty promptu

Mimo tego, że treść promptu przekazywanego do Wielkiego Modelu Językowego jest dowolna, to istnieją pewne komponenty, które można spotkać w większości przykładów.

PROMPT:
[*ICL, instruction*]

> Napisz e-mail składający się z trzech akapitów na potrzeby kampanii marketingowej dla firmy księgowej.

Rysunek 2.2: Podstawowy prompt

PROMPT:
[*ICL, instruction*]

> Sklasyfikuj jako pozytywny lub negatywny: {KOMENTARZ}

Rysunek 2.3: Prompt template [63]

2.2.1 Dyrektywa

Dyrektywa (ang. directive) to element promptu mający na celu 'zachęcenie' modelu do działania [59]. Występuje zazwyczaj w formie instrukcji lub pytania. Jej zadaniem jest określenie celu. Często w literaturze nazywana jest *intent* (ang. intent - zamiar) [58]. Wśród dyrektyw można wyróżnić dwa rodzaje: jawne (rysunek 2.4) i niejawne (rysunek 2.5).

PROMPT:
[*ICL, instruction*]

> Podaj pięć wartych przeczytania książek.

Rysunek 2.4: Prompt zawierający dyrektywę jawną

PROMPT:
[*ICL, one-shot*]

> Noc: Night
> Poranek:

Rysunek 2.5: Prompt zawierający dyrektywę niejawną (intencja: tłumaczenie słowa 'Poranek' na język angielski)

Tak jak wskazuje na to nazwa w przypadku dyrektywy jawnej cel określony jest wprost, zaś w przypadku niejawnej oczekiwanym jest aby to model wywnioskował intencję promptu.

2.2.2 Przykłady

Przykłady (ang. examples) służą jako demonstracja operacji lub zadania (i jego rozwiązania) jakie ma wykonać model. W literaturze przykłady określane są mianem shotów (ang. shot - strzał) lub examplars [58, 13]. Wtedy prompt może zostać określony jako n-shot, gdzie n jest liczbą przykładów (patrz rysunki 2.6 i 2.7).

PROMPT:
[*ICL, one-shot*]

> drapieżny kot o pomarańczowym futrze w czarne pasy: tygrys
>
> król wszystkich zwierząt:

Rysunek 2.6: One-shot prompt na przykładzie odgadywania zwierząt po opisie

PROMPT:
[*ICL, few-shot*]

> Jasno: Ciemno
> Ciepło: Zimno
> Wesoło: Smutno
> Szybko:

Rysunek 2.7: Few-shot prompt na przykładzie antonimów

Warto zaznaczyć, że prompty skonstruowane tak jak na rysunkach 2.6 i 2.7 w literaturze często określane są mianem promptów spełniających paradygmat wypełniania formularza (ang. form-filling paradigm) [45], ponieważ z perspektywy LLM-a jego zadaniem jest dokończenie kolejnego przykładu tak jakby wypełniał swego rodzaju formularz.

2.2.3 Formatowanie wyjścia

Formatowanie wyjścia (ang. output formatting) odnosi się do elementu promptu, który określa jaki ma być format wygenerowanej treści [58] np. CSV, JSON, XML, Markdown itp. przykład przedstawiono na rysunku 2.8. W praktyce określenie formatowania wyjścia często odbywa się poprzez podanie przykładów, szczególnie gdy pożądany format nie jest istniejącym standardem (patrz rysunek 2.9).

PROMPT:
[*ICL, instruction*]

Imię,Nazwisko,Kraj
Krzysztof,Małczak,Polska
Przedstaw powyższy tekst w formacie JSON

Rysunek 2.8: Prompt z formatowaniem wyjścia jako JSON

PROMPT:
[*ICL, few-shot*]

{AKAPIT}
przedstaw wszystkie rzeczowniki z powyższego akapitu w następującym formacie:
- rzeczownik1
- rzeczownik2
- rzeczownik3

Rysunek 2.9: Prompt z formatowaniem wyjścia na bazie przykładów

2.3 Prompt Engineering

To relatywnie nowa dyscyplina zajmująca się opracowywaniem i optymalizacją promptów w celu efektywnego wykorzystania Wielkich Modeli Językowych [21]. W praktyce można ją sprowadzić do iteracyjnego procesu rozwoju oraz dostosowywania promptu [58] w celu uzyskania pożądanego wyniku generowania treści. Proces ten może być zautomatyzowany lub manualny [58].

Techniki prompt engineeringu wykorzystywane są przez badaczy w celu podnoszenia kompetencji LLM-ów w odpowiadaniu na pytania, rozumowaniu dedukcyjnym jak i arytmetycznym [21] oraz innych złożonych zadaniach jak np. generowanie streszczeń.

Ta dyscyplina wykorzystywana jest również bezpośrednio przez inżynierów oprogramowania nie tylko w ramach wsparcia w generowaniu kodu ale przede wszystkim w budowaniu rozwiązań opartych o LLM-y takie jak np. systemy RAG [1].

Poza określaniem strategii iterowania na promptach w celu uzyskania najlepszych wyników [58], dyscyplina ta zajmuje się przede wszystkim definiowaniem podstawowych technik promptingu, czyli schematów określających w jaki sposób definiować, strukturyzować i sekwencjonować prompty [58].

2.4 Wybrane techniki promptingu

Techniką promptingu nazywamy pewien wzorzec określający w jaki sposób tworzyć i definiować prompty, w jaki sposób strukturyzować ich komponenty (patrz 2.2) oraz jak dynamicznie sekwencjonować wiele promptów w celu uzyskania jak najlepszego wyniku. Techniki promptingu mogą polegać na logice warunkowej, w celu tworzenia rozgałęzień [58].

Ten podrozdział skupia się na technikach promptingu tekstowego dzieląc je na trzy główne kategorie. Ze względu na stale rosnącą ilość różnych metod promptingu podrozdział ten ogranicza się do omówienia jedynie tych, które są istotne dla niniejszej pracy.

[1] RAG (ang. Retrieval Augmented Generation - generowanie wzbogacone wyszukiwaniem) - wzorzec, w którym prompt wysyłany do LLM-a wzbogacany jest informacjami z pewnej bazy wiedzy, która nie zawiera się w jego treningowym zbiorze danych [41]

2.4.1 In-Context Learning (ICL)

Jest to kategoria, w której prompty konstruuje się w taki sposób, aby model był w stanie 'nauczyć się'[2] rozwiązywać zadany problem na bazie przykładów (rysunek 2.7) i instrukcji (rysunek 2.3) zawartych w propmpcie bez konieczności fine-tuningu modelu [55, 58]. Kontekst (ang. context) w tym wypadku oznacza informacje ogólnie zawarte w prompcie (niekoniecznie w formie przykładów).

Few-Shot Prompting

Tak jak wskazuje na to nazwa Few-Shot Prompting [13] jest techniką, w której model uczy się jak rozwiązać zadany problem na podstawie zaledwie kilku (ang. few - niewiele) przykładów [58] (rysunek 2.10)

PROMPT:
[*ICL, few-shot*]

```
2+2: cztery
4+5: dziewięć
8+0:
```

Rysunek 2.10: ICL few-shot prompt

2.4.2 Zero-shot

Jest to kategoria, w której prompty nie zawierają żadnych przykładów (patrz 2.2.2) ani konkretnych instrukcji [58] w kontekście.

Role Prompting

Role Prompting [71] znany również jako persona prompting [70] jest techniką polegającą na przypisaniu modelowi pewnej osobowości, co może pozwalać na generowanie treści wyższej jakości w przypadku zadań otwartych [56, 58] (patrz rysunki 2.11 oraz 2.13).

PROMPT:
[*zero-shot, role/persona prompting*]

```
Jesteś copywriterem z wieloletnim doświadczeniem.
Napisz tekst na stronę internetową kawiarni.
```

Rysunek 2.11: Prompt z określeniem persony

Style Prompting

Style prompting [46] to technika, w której treść promptu określa styl, ton, rodzaj języka itp. Role prompting (2.4.2) może pozwalać na uzyskanie podobnych wyników w zależności od przypisanej persony [58]. Przykłady przedstawiono na rysunkach 2.12 oraz 2.13.

[2] w tym wypadku pojęcie 'uczenia się' mimo, że używane w literaturze jest mylące, ponieważ ICL może być jedynie specyfikacją zadania, którego model nauczył się już w trakcie treningu np. formatowanie jako JSON (rysunek 2.8)

PROMPT:
[*zero-shot, style prompting*]

> Napisz abstrakt pracy magisterskiej. Użyj profesjonalnego, akademickiego języka.

Rysunek 2.12: Prompt z określeniem stylu wypowiedzi

PROMPT:
[*zero-shot, role/persona prompting*]

> Jesteś pracownikiem akademickim. Napisz abstrakt pracy magisterskiej.

Rysunek 2.13: Prompt z określeniem persony w celu wymuszenia stylu wypowiedzi. Alternatywa dla style promptingu

2.4.3 Thought Generation

Jest to kategoria obejmująca szeroką gamę technik promptingu, w których oprócz zadanego problemu wymagane jest od LLM-a przedstawienie uzasadnienia swojej odpowiedzi [75, 58] - tzw. 'toku myśli' stąd też nazwa (ang. Thought Generation - generowanie myśli).

Chain-of-Thought (CoT) Prompting

Rozwiązując złożony problem (np. zadanie matematyczne wymagające wielu kroków) ludzie bardzo często rozkładają go na etapy pośrednie [73], technika CoT stara się zasymulować właśnie taki proces.

Chain-of-Thought [73] lub Chain-of-Thoughts [65] prompting jest metodą, polegająca na generowaniu sekwencji pośrednich kroków rozumowania, dzięki przykładom sekwencji myśli zawartym w prompcie [73]. Innymi słowy technika CoT bezpośrednio wykorzystuje Few-Shot prompting (patrz 2.4.1), za pomocą którego wymusza na LLM-ie przedstawienie swojego 'procesu myślowego' zanim dostarczy ostateczną odpowiedź [73, 58, 76]. Każdy przykład zawarty w prompcie składa się z pytania oraz łańcucha rozumowania z oczekiwaną odpowiedzią [76].

CoT znacząco zwiększa efektywność LLM-ów w rozwiązywaniu problemów matematycznych oraz problemów wymagających umiejętności rozumowania dedukcyjnego [73], dzięki zasugerowanemu w prompcie mechanizmowi dekompozycji problemu [73]. Przykład tej techniki przedstawiono na rysunku 2.14.

PROMPT:
[*thought generation, few-shot CoT (one-shot CoT)*]

> **PYT**: Jacek ma 5 piłek tenisowych. Kupił 2 dodatkowe puszki piłek tenisowych. Każda puszka zawiera 3 piłki. Ile piłek tenisowych ma teraz Jacek?
>
> **ODP**: Na początku Jacek miał 5 piłek. W jednej puszcze znajdują się 3 piłki, więc w dwóch puszkach jest łącznie 6 piłek. 5 + 6 = 11. Odpowiedź to 11.
>
> **PYT**: W stołówce znajdowały się 23 jabłka. 20 jabłek zostało zużytych do przygotowania śniadania oraz 6 jabłek zostało dokupionych. Ile jabłek jest teraz w stołówce?

OUTPUT:

> **ODP**: Na początku w stołówce znajdowały się 23 jabłka. 20 zostało zużytych do przygotowania śniadania, czyli zostało im 23 - 20 = 3. Następnie 6 jabłek zostało dokupionych, więc zostało ich 3 + 6 = 9. Odpowiedź to 9.

Rysunek 2.14: One-shot CoT prompt. Wyróżnione fragmenty to łańcuchy rozumowania charakterystyczne dla CoT. Na podstawie [73]

Zero-shot Chain-of-Thought

Nie podając LLM-owi żadnych przykładowych łańcuchów rozumowania, jednocześnie w inny sposób 'zachęcając' go do dekompozycji problemu np. przy użyciu takiej frazy jak "Zastanówmy się krok po kroku"(ang. Let's think step by step) możemy wykonać tzw. Zero-Shot CoT [40] również osiągający przyzwoite wyniki [76]. W przeciwieństwie do Few-Shot CoT, w przypadku którego format odpowiedzi zdefiniowany jest w ramach przykładów, Zero-Shot CoT wymaga dodatkowego zapytania w celu wydobycia ostatecznej odpowiedzi. Oznacza to, że te technikę można podzielić na dwa etapy. Pierwszym jest ekstrakcja rozumowania (rysunek 2.15), zaś drugim ekstrakcja ostatecznej odpowiedzi na bazie rozumowania (rysunek 2.16) [40].

PROMPT:
[*thought generation, zero-shot CoT (reasoning extraction)*]

PYT: Janek w ciągu minuty zadaje średnio 25 ciosów. Walka trwa 5 rund. Każda runda trwa 3 minuty. Ile ciosów zada Janek?

ODP: Zastanówmy się krok po kroku.

OUTPUT:

W jedną minutę Janek zadaje 25 ciosów.
W 3 minuty Janek zadaje 3 * 25 = 75 ciosów.
W 5 rund Janek rzuca 5 * 75 = 375 ciosów

Rysunek 2.15: Zero-Shot CoT - ekstrakcja rozumowania. Na podstawie [40]

PROMPT:
[*thought generation, zero-shot CoT (answer extraction)*]

PYT: Janek w ciągu minuty zadaje średnio 25 ciosów. Walka trwa 5 rund. Każda runda trwa 3 minuty. Ile ciosów zada Janek?

ODP: Zastanówmy się krok po kroku.
W jedną minutę Janek zadaje 25 ciosów.
W 3 minuty Janek zadaje 3 * 25 = 75 ciosów.
W 5 rund Janek rzuca 5 * 75 = 375 ciosów
Zatem odpowiedź (cyframi arabskimi) to:

OUTPUT:

375

Rysunek 2.16: Zero-Shot CoT - ekstrakcja odpowiedzi z wygenerowanego rozumowania. Na podstawie [40]

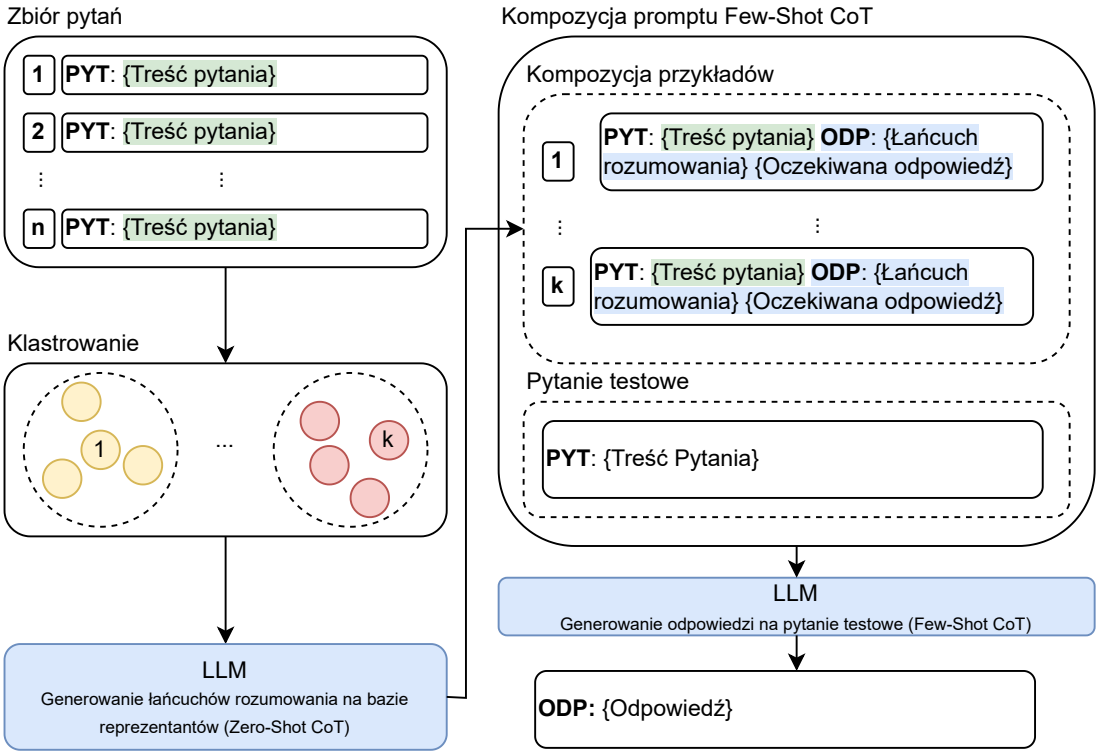
Automatic Chain-of-Thought (Auto CoT) prompting

Wykazano, że Few-Shot CoT (patrz 2.4.3) osiąga lepsze wyniki w generowaniu treści niż Zero-Shot CoT [73, 40, 76]. Niestety w przypadku Few-Shot CoT konieczne jest ręczne zdefiniowanie przykładów[3], czyli pytań wraz z łańcuchem rozumowania i poprawną odpowiedzią, co w przypadku większości złożonych problemów jest nietrywialne [76] czyniąc te technikę bardzo kosztowną. Z tego powodu większość mechanizmów mających na celu automatyzację przy jednoczesnym zachowaniu poprawności i wysokiego poziomu generowanych treści nie jest w stanie wykorzystać tej metody.

Auto Chain-of-Thought ma na celu wyeliminowanie potrzeby ręcznego przygotowywania łańcuchów rozumowania wraz z odpowiedziami poprzez wygenerowanie ich dla każdej klasy pytań[4] za pomocą techniki Zero-Shot CoT (patrz 2.4.3). Następnie wykonywana jest kompozycja pytania, łańcucha rozumowania oraz odpowiedzi w celu uzyskania przykładów. Ostatnim krokiem jest zastosowanie Few-Shot CoT (patrz 2.4.3) przy użyciu skomponowanych przykładów [76]. Na rysunku 2.17 wizualnie przedstawiono zasadę działania tej techniki. Warto zwrócić uwagę, że w przypadku wykorzystania Auto CoT do generowania

[3] z tego powodu w literaturze można te technikę spotkać pod nazwą Manual-CoT (ang. manual - ręczny, manualny) [76]
[4] generowanie dla danej klasy odbywa się na podstawie jednego jej reprezentanta [76]

rozwiązań dla problemów jednej klasy etap ich klastrowania może zostać pominięty zaś jej reprezentant może zostać wybrany ręcznie.



Rysunek 2.17: Schemat działania Auto CoT. Na podstawie: [76]

Rozdział 3

Metody oceny efektywności Wielkich Modeli Językowych

Wielkie Modele językowe stale zyskują na popularności zarówno w środowisku akademickim jak i w różnych domenach przemysłowych [10, 72, 78]. Ze względu na ich rosnącą ilość zastosowań w codziennym życiu ludzi oraz bardziej krytycznych dziedzinach jak medycyna czy prawo, analiza ich efektywności, poprawności i przydatności znacznie zwiększa swoje znaczenie.

Ocena jakości tekstów generowanych maszynowo od dawna stanowi wyzwanie w dziedzinie przetwarzania języka naturalnego, zaś dziś pełni rolę krytycznego filaru niezbędnego do zrozumienia właściwości i zachowania LLM-ów [39, 43, 16, 82, 19, 33]. Ten rozdział skupia się na metodykach związanych z oceną Wielkich Modeli Językowych zaczynając od ogólnego przedstawienia konceptu ewaluacji a kończąc na jego szczególnych zastosowaniach porównawczych. Omawiane są podstawowe metryki służące do oceniania treści generowanych przez modele, różne strategie implementowania tych metryk, zarówno z oraz bez wykorzystania w tych celach sztucznej inteligencji.

3.1 Ewaluacja LLM-ów

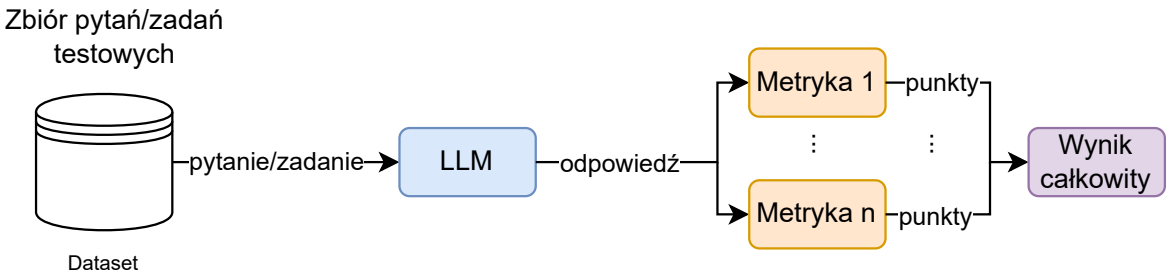
Patrząc na obecne badania [14] obietnica osiągnięcia AGI staje się coraz bardziej realna [16]. W przeciwieństwie do poprzednich modeli, tworzonych i trenowanych z myślą o konkretnej klasie zadań, obecnie LLM-y wykazują wysoką wydajność w rozwiązywaniu szerokiej gamy problemów takich jak ogólne i specyficzne dla danej domeny zadania związane z językiem naturalnym [16], przez co pojawia się szereg specjalistów różnych dziedzin, którzy chcą wykorzystywać LLM-y jako narzędzie do pracy. Jednakże bardzo często te jednostki (np. personel medyczny) wymagają ogromnej precyzji, przez co modele językowe muszą być badane pod kątem efektywności. Podstawą takich badań jest szeroko pojęty koncept ewaluacji.

Ewaluacja Wielkich Modeli Językowych odnosi się ogólnie do procesu oceny ich możliwości, zgodności z wartościami ludzkimi [30] ale przede wszystkim skuteczności w realizacji określonych zadań, takich jak generowanie tekstu, tłumaczenie, klasyfikacja danych, streszczanie, czy bardziej specyficznych [30] np. odpowiadanie na zapytania klienta w roli asystenta sklepu internetowego. Ogólny mechanizm ewaluacji LLM-ów przedstawiono na rysunku 3.1.

3.2 Potrzeba ewaluacji: Motywacje i Korzyści

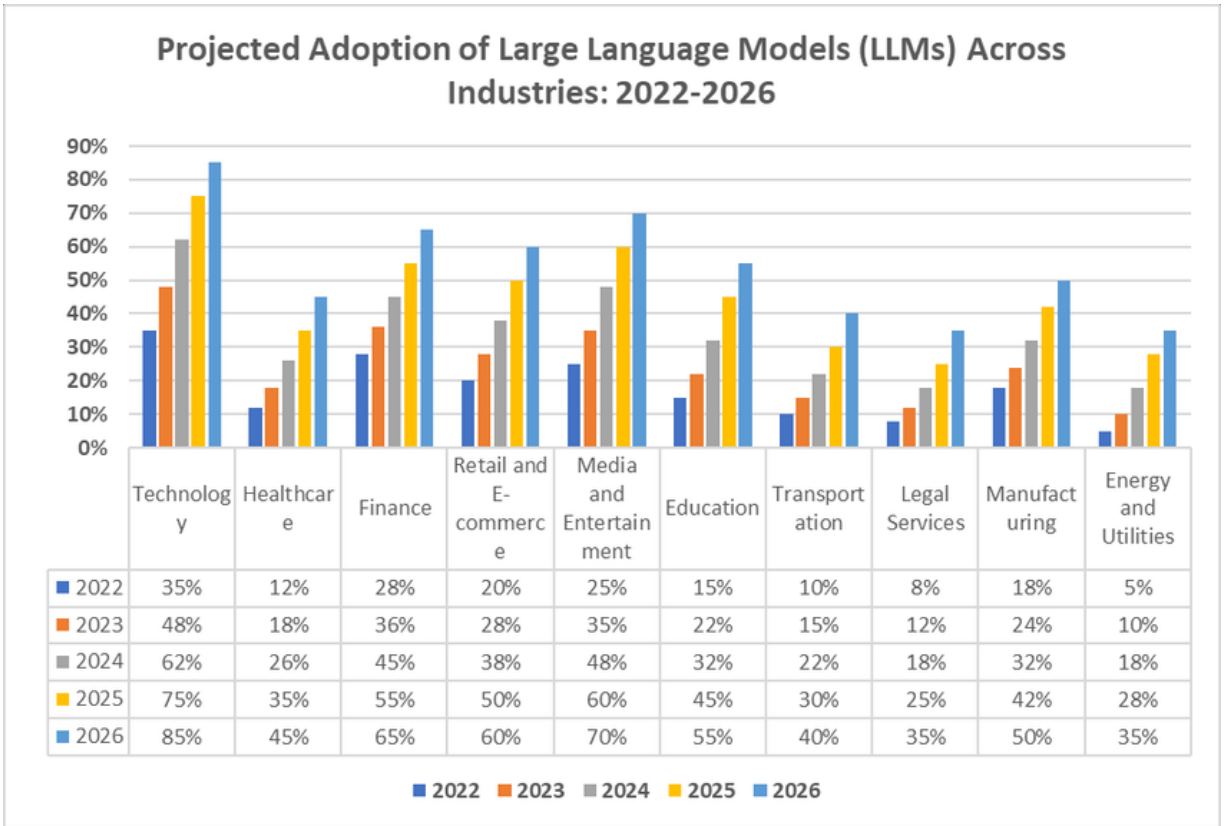
Wielkie Modele Językowe jako sztuczna inteligencja stworzona do celów nieodłącznie związanych z NLP, musi być stale monitorowana. Ewaluacja jest kluczowym etapem w rozwoju i tworzeniu LLM-ów oraz rozwiązań o nich opartych, ponieważ umożliwia zrozumienie potencjalnych zastosowań, jak również mocnych oraz słabych stron konkretnych modeli [30].

Jak przedstawiono na diagramie 3.2 rozwiązania oparte o LLM-y coraz częściej znajdują swoje zastosowanie w niezwykle ważnych sektorach jak medycyna czy finanse [80] [42], gdzie niezawodność oraz bezpieczeństwo są najistotniejszymi aspektami [16]. Dzięki odpowiednio dobranemu zbiorowi danych testowych



Rysunek 3.1: Ogólny mechanizm ewaluacji LLM-ów

oraz stosowanej strategii ewaluacji modele lub systemy o nie oparte mogą być poddawane odpowiedniemu fine-tuningowi, dzięki czemu możliwe jest zwiększenie rzetelności generowanych treści.



	Technology	Healthcare	Finance	Retail and E-commerce	Media and Entertainment	Education	Transportation	Legal Services	Manufacturing	Energy and Utilities
2022	35%	12%	28%	20%	25%	15%	10%	8%	18%	5%
2023	48%	18%	36%	28%	35%	22%	15%	12%	24%	10%
2024	62%	26%	45%	38%	48%	32%	22%	18%	32%	18%
2025	75%	35%	55%	50%	60%	45%	30%	25%	42%	28%
2026	85%	45%	65%	60%	70%	55%	40%	35%	50%	35%

Rysunek 3.2: Przewidywane wdrożenie Wielkich Modeli Językowych w wybranych branżach do 2026r. Źródło: [54]

Dodatkowo naturalnym efektem ubocznym ewaluacji jest szansa na optymalizację generowanego tekstu oraz identyfikację obszarów, w których modele mogą być podatne na błędy i halucynacje.

Potrzeba ewaluacji wynika również z różnorodności zastosowań LLMów – modele te są wykorzystywane zarówno w prostych chatbotach, jak i w bardziej zaawansowanych aplikacjach takich jak systemy wspierające procesy decyzyjne. W takich przypadkach niezbędne jest dokładne zrozumienie poziomu ich trafności, zaufania, wiarygodności [30] itp., które badane są przy użyciu specjalnych metryk.

3.3 Metryki ewaluacyjne

Badając treści generowane przez LLMy niezbędna jest możliwość kwantyfikacji, w celu ocenienia czy dana treść jest wystarczająco dobra [37] dla danego przypadku użycia. Powszechną praktyką jest nadawanie takim treściom pewnej ilości punktów (patrz rysunek 3.1), do czego wykorzystywane są metryki ewaluacyjne wraz z odpowiednim mechanizmem punktowania (ang. scorer) - więcej w podrozdziale 3.5. W literaturze dla prostoty same mechanizmy punktowania często nazywane są metrykami, gdyż są z nimi nieodłącznie związane. Istnieje wiele różnych metryk wykorzystywanych do ewaluacji LLMów, poza popularnymi, które zostały przedstawione w tabeli 3.1 często pojawia się potrzeba definiowania metryk wymagających niestandardowych kryteriów oceny w zależności od przypadku użycia np. ocena generowanych streszczeń [37].

Metryka	Opis
Trafność (ang. Relevancy)	Określa, czy wygenerowana odpowiedź odnosi się do danych wejściowych w sposób informacyjny i zwięzły
Poprawność (ang. Correctness)	Określa, czy wygenerowana odpowiedź jest faktycznie poprawna w oparciu o pewną prawdę (tzw. ground truth)
Halucynacja (ang. Hallucination)	Określa, czy wygenerowana odpowiedź zawiera fałszywe informacje
Toksyczność (ang. Toxicity)	Określa, czy wygenerowana odpowiedź zawiera szkodliwe lub obraźliwe treści

Tabela 3.1: Zestawienie wybranych, najbardziej powszechnych metryk do ewaluacji treści generowanych przez LLM-y

3.4 Metody ewaluacji

Uwzględniając aktorów biorących udział w procesie ewaluacji, jej obecnie istniejące metody można podzielić na dwie podstawowe kategorie:

- Ewaluacja dokonywana przez człowieka
- Ewaluacja automatyczna

3.4.1 Ewaluacja ludzka (ang. Human Evaluation)

Ewaluacja wygenerowanej treści dokonywana przez człowieka (tzw. anotatora[1]) polega na ręcznym przypisywaniu ilości punktów dla każdej z uwzględnianych metryk. Naturalnie w przeciwieństwie do ewaluacji zautomatyzowanej cechuje się obniżoną możliwością skalowania, ponieważ wymaga aktywnego udziału i wysokiego nakładu pracy człowieka, przez co wydłuża się jej czas oraz koszt ale również wprowadza się czynnik subiektywności [30, 16, 69, 52, 7, 22] - ocena jest subiektywna w zależności od oceniającego. Dodatkowym problemem tej metody jest fakt, iż ludzie mają tendencję do postrzegania stanowczych treści jako bardziej zgodnych z faktami, niezależnie od informacji w nich zawartych [34]. Jednakże mimo wspomnianej subiektywności, podatności i potencjalnej omylności anotatorów ewaluacja dokonywana przez człowieka pozostaje konsekwentnie dominującą metodą ze względu na ludzką zdolność uwzględniania subtelnych niuansów w generowanych treściach oraz docenianie takich aspektów jak zwięzłość czy kultura wypowiedzi [39].

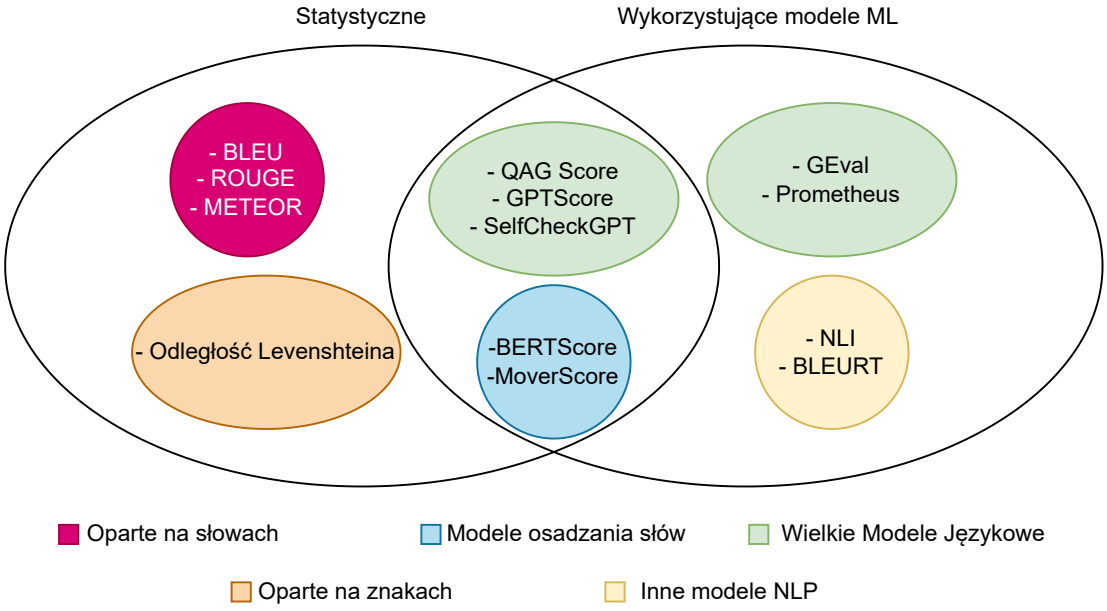
[1] ang. to annotate - dodawać adnotacje, etykietować

3.4.2 Ewaluacja zautomatyzowana (ang. Automatic Evaluation)

Ze względu na ilość manualnej pracy wymaganej w przypadku ewaluacji ludzkiej oraz idącymi za tym kosztami i czasem ewaluacja zautomatyzowana staje się obecnie bardziej powszechna, szczególnie w zastosowaniach biznesowych. Wszystkie strategie pozwalające na zautomatyzowaną ocenę generowanych treści dążą do osiągnięcia możliwie wysokiej zgodności z ocenami ludzkimi (ewaluacją ludzką), z tego powodu wyniki ewaluacji zautomatyzowanej powinny być poddawane ludzkiemu nadzorowi, przynajmniej w ograniczonym stopniu [9, 34].

3.5 Mechanizmy punktowania w ewaluacji zautomatyzowanej

Jedną z zalet ewaluacji zautomatyzowanej jest brak czynnika subiektywności, dzięki czemu możliwy jest nawet pewien stopień standaryzacji [30, 16]. Jednakże ocena pod kątem takich kryteriów jak trafność, czy przydatność (patrz tabela 3.1) wymaga możliwości analizy języka naturalnego lub chociaż jej symulacji przy użyciu metod statystycznych. Z tego powodu do automatycznego punktowania generowanych treści dla takich metryk często wykorzystuje się inny model uczenia maszynowego. Wprowadza to naturalny podział wśród mechanizmów punktowania wykorzystywanych w ewaluacji zautomatyzowanej, jak przedstawiono to na rysunku 3.3.



Rysunek 3.3: Mechanizmy punktowania wykorzystywane w zautomatyzowanej ewaluacji LLM-ów. Na podstawie: [37]

3.5.1 Statystyczne

Mimo obliczeniowej niezawodności czysto statystycznych mechanizmów punktowania, nie są one wystarczające do oceny zazwyczaj długich i złożonych treści generowanych przez LLM-y w przypadku większości istotnych metryk. Wynika to głownie z faktu, że mechanizmy te opierają się na porównywaniu treści wygenerowanej z treścią oczekiwaną, co oznacza, że w zbiorze danych wykorzystywanych do ewaluacji dla każdego wejścia testowego musi być zdefiniowane oczekiwane wyjście - tak zwana referencja [79]. Wynika z tego, że metody te nie nadają się do oceny odpowiedzi wygenerowanych np. na pytania otwarte, dla

których taka referencja nie może być jednoznacznie zdefiniowana [79]. Dodatkowo metody te nie biorą pod uwagę semantyki [2] tekstów, mają ogromne ograniczenia w porównywaniu tekstów wymagających umiejętności rozumowania dedukcyjnego [37]. Ponadto poziom zaawansowania generowanych tekstów jest na tyle wysoki, że porównywanie ich na podstawie cech powierzchniowych może być niemiarodajne [81, 27].

BLEU (BiLingual Evaluation Understudy)

Pozwala ocenić wygenerowaną treść na podstawie pewnej prawdy (tzw. referencji) czyli oczekiwanego wyniku. Obliczana jest precyzja dla każdego n-gramu (ciągu n kolejnych słów), porównując wygenerowaną treść z jedną lub kilkoma referencjami przygotowanymi przez ludzi. Precyzja oznacza, jaki procent n-gramów w wygenerowanej treści występuje również w referencji. Końcowy wynik oblicza się jako średnią geometryczną precyzji dla różnej długości n-gramów. Dodatkowo aby uniknąć zawyżania precyzji w przypadku krótkich treści stosuje się tzw. "brevity penalty"(ang. karę za zwięzłość). Wynik znajduje się w zakresie od 0 do 1 (lub w formie procentowej), gdzie wyższy wynik oznacza lepsze dopasowanie [53].

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

Głównie służy do oceny generowanych streszczeń i tłumaczeń maszynowych. Oblicza tzw. 'recall' (ang. odwołanie), który określa jak dobrze wygenerowana treść pokrywa się z oczekiwanym wynikiem, poprzez porównanie n-gramów (ciągów słów) wygenerowanego tekstu z n-gramami występującymi w tekście referencyjnym. Wynik ROUGE mieści się w przedziale od 0 do 1, gdzie wyższa wartość oznacza większe podobieństwo tekstów. Wysoki wynik wartości recall wskazuje na to, że model skutecznie uchwycił istotne informacje zawarte w treści referencyjnej [44].

METEOR (Metric for Evaluation of Translation with Explicit Ordering)

Ocenia wygenerowaną treść względem oczekiwanego wyniku (referencji) na podstawie wartości zarówno precyzji jak i recall, jednocześnie biorąc pod uwagę możliwą różnicę w kolejności słów. Dodatkowo wykorzystuje zewnętrzne lingwistyczne zbiory danych jak WordNet [3] w celu znajdowania synonimów. Ostatecznym wynikiem jest średnia harmoniczna wartości precyzji oraz recall obarczona karą za rozbieżności w kolejności lub użyciu innej formy słów [8].

Odległość Levenshteina (edycyjna)

Ocenia jak bardzo różni się treść wygenerowana z oczekiwanym wynikiem (referencją), poprzez określenie ile operacji (wstawienie, usunięcie lub zamiana znaku) trzeba wykonać, aby przekształcić jeden ciąg znaków w drugi [31].

3.5.2 Wykorzystujące modele uczenia maszynowego

Implementacja statystycznego mechanizmu punktowania, czy też proceduralnego algorytmu efektywnie oceniającego odpowiedzi na pytania otwarte w postaci języka naturalnego jest niezwykle wymagająca [79]. Z tego powodu w takich przypadkach do oceny generowanych tekstów często wykorzystuje się modele uczenia maszynowego. Patrząc na diagram 3.5 wśród nich można wyróżnić te, które wykorzystują Wielkie Modele Językowe i te, które tego nie robią. Podobnie jak w przypadku czysto statystycznych mechanizmów, te które nie wykorzystują LLM-ów uzyskują niższy poziom korelacji z ocenami ludzkimi, szczególnie w kontekście zadań kreatywnych i pytań o charakterze otwartym [45, 37, 39].

NLI (Natural Language Inference)

Odnosi się do wykorzystania modeli lub technik NLP, które pozwalają na określenie jak dobrze LLM radzi sobie z określaniem relacji między zdaniami lub tekstami. Relacje klasyfikowane są zazwyczaj do trzech głównych kategorii:

[2] Semantyka - dział językoznawstwa, którego przedmiotem jest analiza znaczeń wyrazów [2]
[3] WordNet - baza angielskich słów oraz relacji między nimi zapoczątkowana przez George'a Millera w 1978r.

- ciągłość logiczna (ang. entilement) - zdanie kontynuuje ciąg logiczny swojego poprzednika (poprzedników)
- sprzeczność (ang. contradiction) - zdanie jest sprzeczne z pozostałymi w tekście
- neutralność (ang. neutral) - zdanie jest neutralne względem pozostałych w tekście

Ostateczny wynik pochodzi z przedziału od 0 (sprzeczność) do 1 (ciągłość logiczna) [11].

BLEURT (BiLingual Evaluation Understudy with Representations from Transformers)

Mechanizm wykorzystujący modele takie jak np. BERT [4] , które poddawane są kolejny fazom treningu, na początku na bazie danych syntetycznych a następnie danych etykietowanych przez ludzi. Ostatnia faza jest opcjonalna, lecz pozwala na uzyskanie lepszych wyników dla danego przypadku użycia. Oceny BLEURT są bardziej trafne niż w przypadku użycia BLEU lub ROUGE, nawet w przypadku niskiej jakości danych treningowych [61, 60, 1].

G-Eval

G-Eval [45] znacznie różni się od pozostałych metod, ponieważ nie funkcjonuje jedynie jako mechanizm punktowania ale raczej pełnoprawny framework ewaluacyjny [45], dokładnie określający poszczególne etapy prowadzące do uzyskania efektywnej oceny generowanych treści. G-Eval bezpośrednio wykorzystuje Wielki Model Językowy[5], którego zadaniem jest ocenienie wcześniej wygenerowanej treści przy użyciu zaawansowanych technik promptingu. Na rysunku 3.4 wizualnie przedstawiono mechanizm działania frameworka G-Eval.

W celu oceny wygenerowanej treści wykorzystywana jest strategia Auto CoT (patrz 2.4.3), która na bazie wprowadzenia do zadania (ang. task introduction) oraz określonych kryteriów ewaluacyjnych (ang. evaluation criteria), pozwala na wygenerowanie szczegółowych kroków niezbędnych do wykonania w ramach procesu oceny. Następnie wygenerowane kroki wraz z wejściem z testowego zbioru danych w formie promptu spełniającego paradygmat wypełniania formularza (patrz 2.2.2) ponownie trafiają do LLM-a, który jako odpowiedź zwraca ocenę treści. Opcjonalnie ostateczna ocena zwrócona przez LLM może zostać znormalizowana na bazie prawdopodobieństwa każdego ze zwracanych tokenów poprzez obliczenie sumy ważonej:

$$score = \sum_{i=1}^{n} p(s_i) \times s_i$$

gdzie s_i należy do zbioru możliwych ocen (np. od 1 do 5) $S = \{s_1, s_2, \ldots, s_n\}$ zdefiniowanych bezpośrednio w prompcie, zaś $p(s_i)$ to prawdopodobieństwo wystąpienia obliczane przez LLM dla każdego tokenu będącego wynikiem. Wykorzystanie G-Eval pozwala na uzyskanie wyników o znacznie wyższej korelacji z ocenami ludzkimi [45], co pozwala na bardziej niezawodną ewaluację zautomatyzowaną.

Prometheus

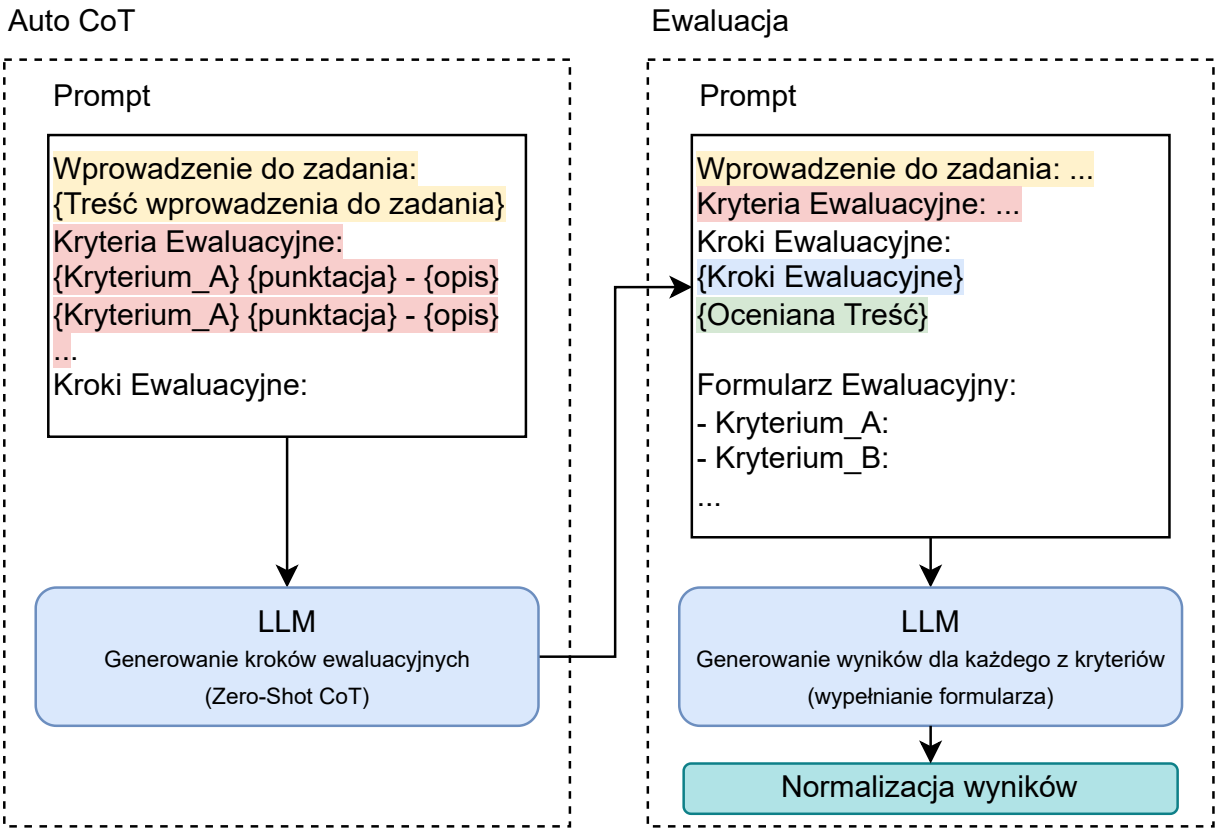
Prometheus [39] jest Wielkim Modelem Językowym posiadającym $13 * 10^9$ parametrów, który pozwala na ewaluację treści na bazie tzw. rubryki punktów (ang. score rubric) oraz odpowiedzi referencyjnej, przekazywanej do niego w treści promptu ewaluacyjnego (patrz rysunek 3.5).

W rzeczywistości modelem bazowym jest LLama-2-chat, który poddany został fine-tuningowi z wykorzystaniem specjalnie przygotowanego datasetu nazwanego 'Feedback Collection', w którym każda para składa się z wejścia i wyjścia. Wejście stanowi:

- **Instrukcja**: treść zadania podawana przez użytkownika jako wejście do LLM-a
- **Treść do ewaluacji**: odpowiedź na zadanie zdefiniowane w instrukcji wygenerowana przez LLM, która ma zostać oceniona

[4] BERT (Bidirectional Encoder Representations from Transformers) - model językowy opracowany przez Google. Pozwala m.in. na: analizę sentymentu, rozpoznawanie emocji, generowanie tekstu, generowanie streszczeń [51].

[5] w oryginalnej implementacji wykorzystano modele GPT-4 oraz GPT-3.5 [45]



Rysunek 3.4: G-Eval - machanizm działania. Na podstawie: [45]

PROMPT:
[Prometheus evaluation prompt]

Instrukcja:
{TREŚĆ ZADANIA}
Rubryka Punktów: Czy odpowiedź wykazuje wrażliwość kulturową?

- 1 punkt: {KRYTERIA PRZYZNANIA}
- ...
- 5 punktów: {KRYTERIA PRZYZNANIA}

Odpowiedź referencyjna:
{TREŚĆ OCZEKIWANEJ ODPOWIEDZI}

Rysunek 3.5: Schemat promptu ewaluacyjnego dla modelu Prometheus. Na podstawie [39]

- **Dostosowana rubryka punktów**: składająca się z opisu kryteriów oraz wysokości punktów (od 1 do 5)
- **Odpowiedź referencyjna**: oczekiwana odpowiedź - lub też odpowiedź której model powinien przyznać 5 punktów

zaś wyjście stanowi:

- **Feedback** (ang. informacja zwrotna): uzasadnienie dlaczego treść poddawana ewaluacji otrzymałaby dany wynik - analogicznie do Chain-of-Thought (patrz 2.4.3) - pozwala to na interpretacje uzyskanych wyników

- **Wynik**: ilość punktów (liczba naturalna między 1 a 5)

Warto zaznaczyć, że większość elementów zbioru 'Feedback Collection' zostało wygenerowanych przy użyciu modelu GPT-4, co pozwoliło na efektywne dostrojenie modelu a w efekcie uzyskanie Wielkiego Modelu Językowego, służącego jako ewaluator o zdolności oceny generowanych treści zbliżonej do GPT-4 [39], z tą różnicą, że Prometheus jest w pełni open-source'owym rozwiązaniem [39]. Podobnie do G-Eval efektywność ewaluacji przy użyciu modelu Prometheus nie jest zależna od zadania [37], jednakże między tymi rozwiązaniami istnieją znaczące różnice, które przedstawiono w tabeli 3.2.

G-Eval	Prometheus
G-Eval jest frameworkiem, który w swojej oryginalnej implementacji bezpośrednio wykorzystuje model GPT-3.5/GPT-4 jako podmiot oceniający generowane treści	Prometheus jest LLM-em poddanym treningowi, który czyni go wydajnym ewaluatorem
W przypadku G-Eval kryteria ewaluacyjne definiowane są już na etapie generowania kroków ewaluacyjnych w ramach procesu Auto CoT (patrz 3.4, 2.4.3)	W przypadku Prometheus kryteria ewaluacyjne w postaci rubryki punktów przekazywane są bezpośrednio do modelu jako część promptu
G-Eval charakteryzuje się znacznie większą elastycznością, gdyż nie wymaga definiowana oczekiwanej odpowiedzi (jednocześnie tego nie zabraniając), co czyni go lepszym kandydatem dla problemów otwartych	Prometheus wymaga zdefiniowania odpowiedzi referencyjnej, która pozwala mu na dokonanie oceny
G-Eval skupia się na dostarczeniu ewaluacji możliwie najbliższej do ewaluacji ludzkiej, bezpośrednio wykorzystując jeden z najnowocześniejszych komercyjnych modeli	Prometheus został stworzony w celu zapewnienia rozwiązania ewaluacyjnego uniezależnionego od komercyjnego modelu o zbliżonych możliwościach ewaluacyjnych

Tabela 3.2: Różnice między G-Eval a Prometheus. Na podstawie [45, 39]

3.5.3 Mieszane

TODO: zamiast opisywać wszystkie dokładnie napisać dwa zdania podsumowujące i sugerujące dlaczego ich nie będziemy używać w tym benchmarku (CLUE: CHCEMY UZYSKAĆ OCENY MOŻLIWIE BLISKIE OCENOM JAKIE PRZYDZIELILI BY LUDZIE!!!!!! HELPS)

QAG Score

(TODO: [68])

GPTScore

(TODO: GPTScore [26])

SelfCheckGPT

(TODO: [48])

BERTScore

(TODO: [74] [45])

MoverScore

(TODO: [77] [45])

3.6 LLM jako sędzia w ocenie generowanych treści

TODO: nawiązać do G-Eval i prometheus - te dwa rozwiązania punktujące są dosłowną implementacją tego konceptu!!!

Wykazano, że tradycyjne mechanizmy punktowania takie jak BLEU [53], ROUGE [44] czy METEOR [8] szeroko stosowane do ewaluacji systemów NLG osiągają stosunkowo niską korelację z ocenami ludzkimi [39], w szczególności w problemach o charakterze otwartym oraz zadaniach nie posiadających referencji (tzw. reference-free), które w przypadku tych metryk muszą być zdefiniowane dla każdego wejścia testowego [45, 79]. Wraz z rozwojem LLM-ów, rośnie ich potencjał w całkowitym zastąpieniu ludzkich anotatorów w procesach ewaluacyjnych [28, 35, 79]. Ostatnie badania wprost sugerują bezpośrednie wykorzystanie Wielkich Modeli Językowych jako 'sędziów' dokonujących oceny treści generowanych przez LLM-y [26] na podstawie zdefiniowanych kryteriów. Ten wzorzec w literaturze często nazywany jest 'LLM as a judge' (ang. LLM w roli sędziego)[79].

3.6.1 Korzyści

Wykorzystanie LLM-a do oceny tekstu generowanego przez inny model, podobnie jak w przypadku większości zautomatyzowanych metod ewaluacji (patrz 3.5), pozwala na kompletne usunięcie ludzkiego aktora z procesu ewaluacji, co skutkuje:

- lepszą skalowalnością i szybszą iteracją - LLM-y mogą relatywnie szybko przetwarzać ogromne ilości danych [79]
- redukcją kosztów - zmniejszając potrzebę intensywnej pracy ludzkiej
- większą elastycznością - fine-tuning może pozwolić na dostosowanie do danego rodzaju ewaluacji poprzez zwiększenie trafności
- usunięciem czynnika ludzkiej subiektywności - ocena nie jest dokonywana przez anotatora - ograniczana jest stronniczość
- możliwością automatycznego generowania uzasadnień - LLM może automatycznie wygenerować wyjaśnienia dla swoich ocen, pozwalając na interpretacje [79]

3.6.2 Ograniczenia

Idea wykorzystania Wielkiego Modelu Językowego jako podmiotu przypisującego punkty generowanym treściom, ma rację bytu jedynie przy założeniu, że obecny stan wytrenowania LLM-ów rzeczywiście pozwala im na przypisywanie wyższego prawdopodobieństwa wysokiej jakości tekstom [45]. Jednakże poprawność oraz niezawodność LLM-ów jako sędziów nie jest jeszcze dokładnie przebadana, co potwierdzają meta-ewaluacje [6] , które pokazują, że niektóre metryki ewaluacyjne oparte o LLM-y mogą wykazywać niższą korelację z ocenami ludzkimi niż metryki oparte o średniej wielkości modele uczenia maszynowego [36]. Poza tym podobnie jak w przypadku ewaluacji dokonywanej przez ludzi, pojawia się problem uprzedzeń - tzw. Self-enhancement bias (ang. autowaloryzacja) [12]. Jest to zjawisko, w którym LLM działający jako sędzia faworyzuje odpowiedzi generowane przez samego siebie (przez ten sam model) [79], co może negatywnie wpływać np. na testy mające na celu porównanie różnych modeli, wśród których jeden z ewaluowanych jest również wykorzystany jako sędzia.

[6] Meta-ewaluacja - proces oceniania użyteczności metryk ewaluacyjnych wraz z mechanizmami punktowania [18].

Bibliografia

[1] Metric: bleurt. https://huggingface.co/spaces/evaluate-metric/bleurt.

[2] Słownik Języka Polskiego - PWN. https://sjp.pwn.pl/slowniki/semantyka.html.

[3] How does LLM benchmarking work? An introduction to evaluating models. https://symflower.com/en/company/blog/2024/llm-evaluation-101, 2024.

[4] X. Amatriain. Prompt design and engineering: Introduction and advanced methods, 2024.

[5] S. An, Z. Ma, Z. Lin, N. Zheng, J.-G. Lou. Make your llm fully utilize the context, 2024.

[6] T. Ayodele. *Introduction to Machine Learning*. 02 2010.

[7] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, J. Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

[8] S. Banerjee, A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. J. Goldstein, A. Lavie, C.-Y. Lin, C. Voss, redaktorzy, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, strony 65–72, Ann Arbor, Michigan, Czerw. 2005. Association for Computational Linguistics.

[9] P. Bhavsar, B. Gheorghe. LLM-as-a-Judge vs Human Evaluation. https://www.galileo.ai/blog/llm-as-a-judge-vs-human-evaluation, 2024.

[10] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, P. Liang. On the opportunities and risks of foundation models, 2022.

[11] S. R. Bowman, G. Angeli, C. Potts, C. D. Manning. A large annotated corpus for learning natural language inference, 2015.

[12] J. D. Brown. Evaluations of self and others: Self-enhancement biases in social judgments. *Social cognition*, 4(4):353–376, 1986.

[13] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei. Language models are few-shot learners, 2020.

[14] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, Y. Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[15] A. Cardillo. List of the Best 21 Large Language Models (LLMs) (September 2024). https://explodingtopics.com/blog/list-of-llms, 2024.

[16] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, X. Xie. A survey on evaluation of large language models, 2023.

[17] G. H. Chen, S. Chen, Z. Liu, F. Jiang, B. Wang. Humans or llms as the judge? a study on judgement biases, 2024.

[18] S. Chern, E. Chern, G. Neubig, P. Liu. Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate, 2024.

[19] Y. K. Chia, P. Hong, L. Bing, S. Poria. Instructeval: Towards holistic evaluation of instruction-tuned large language models, 2023.

[20] A. Chockalingam. A Beginner's Guide to Large Language Models Part 1. https://www.amax.com/content/files/2024/03/llm-ebook-part1-1.pdf, 2023.

[21] DAIR.AI. Prompt Engineering Guide. https://www.promptingguide.ai/, 2024.

[22] S. Diao, R. Pan, H. Dong, K. S. Shum, J. Zhang, W. Xiong, T. Zhang. Lmflow: An extensible toolkit for finetuning and inference of large foundation models, 2024.

[23] K. K. S. S. Diksha Khurana, Aditya Koli. Natural language processing: state of the art,current trends and challenges. https://rdcu.be/dZdv3, 2022.

[24] C. Dong, Y. Li, H. Gong, M. Chen, J. Li, Y. Shen, M. Yang. A survey of natural language generation. *ACM Computing Surveys*, 55(8):1–38, Gru. 2022.

[25] Y. Dubois, B. Galambosi, P. Liang, T. B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators, 2024.

[26] J. Fu, S.-K. Ng, Z. Jiang, P. Liu. Gptscore: Evaluate as you desire, 2023.

[27] S. Gehrmann, E. Clark, T. Sellam. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text, 2022.

[28] F. Gilardi, M. Alizadeh, M. Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30), Lip. 2023.

[29] B. Goertzel. Artificial general intelligence: Concept, state of the art, and future prospects. *Journal of Artificial General Intelligence*, 0, 01 2014.

[30] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, Supryadi, L. Yu, Y. Liu, J. Li, B. Xiong, D. Xiong. Evaluating large language models: A comprehensive survey, 2023.

[31] R. Haldar, D. Mukhopadhyay. Levenshtein distance technique in dictionary lookup methods: An improved approach, 2011.

[32] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, J. Steinhardt. Measuring massive multitask language understanding, 2021.

[33] A. Holtzman, P. West, L. Zettlemoyer. Generative models as a complex systems science: How can we make sense of large language model behavior?, 2023.

[34] T. Hosking, P. Blunsom, M. Bartolo. Human feedback is not gold standard, 2024.

[35] F. Huang, H. Kwak, J. An. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. *Companion Proceedings of the ACM Web Conference 2023*, WWW '23, strona 294–297. ACM, Kwi. 2023.

[36] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023.

[37] J. Ip. LLM Evaluation Metrics: The Ultimate LLM Evaluation Guide. https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation, 2024.

[38] M. R. J, K. VM, H. Warrier, Y. Gupta. Fine tuning llm for enterprise: Practical guidelines and recommendations, 2024.

[39] S. Kim, J. Shin, Y. Cho, J. Jang, S. Longpre, H. Lee, S. Yun, S. Shin, S. Kim, J. Thorne, M. Seo. Prometheus: Inducing fine-grained evaluation capability in language models, 2024.

[40] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa. Large language models are zero-shot reasoners, 2023.

[41] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[42] Y. Li, S. Wang, H. Ding, H. Chen. Large language models in finance: A survey, 2024.

[43] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, Y. Koreeda. Holistic evaluation of language models, 2023.

[44] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, strony 74–81, Barcelona, Spain, Lip. 2004. Association for Computational Linguistics.

[45] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, 2023.

[46] A. Lu, H. Zhang, Y. Zhang, X. Wang, D. Yang. Bounding the capabilities of large language models in open text generation with prompt constraints, 2023.

[47] B. Lutkevich. What is framework? https://www.techtarget.com/whatis/definition/framework, 2020.

[48] P. Manakul, A. Liusie, M. J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023.

[49] C. McClain. Americans' use of ChatGPT is ticking up, but few trust its election information. https://www.pewresearch.org/short-reads/2024/03/26/americans-use-of-chatgpt-is-ticking-up-but-few-trust-its-election-information/, 2024.

[50] B. Meskó. Prompt engineering as an important emerging skill for medical professionals: Tutorial. *J Med Internet Res*, 25:e50638, Oct 2023.

[51] B. Muller. BERT 101: State Of The Art NLP Model Explained. https://huggingface.co/blog/bert-101, 2022.

[52] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe. Training language models to follow instructions with human feedback, 2022.

[53] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. P. Isabelle, E. Charniak, D. Lin, redaktorzy, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, strony 311–318, Philadelphia, Pennsylvania, USA, Lip. 2002. Association for Computational Linguistics.

[54] D. Peringanji. The impact of large language models (llms) on everyday applications: Opportunities, challenges, and considerations, 2024.

[55] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever. Language models are unsupervised multitask learners. 2019.

[56] L. Reynolds, K. McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA, 2021. Association for Computing Machinery.

[57] C. S. Rina Caballar. What are LLM benchmarks? https://www.ibm.com/think/topics/llm-benchmarks, 2024.

[58] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff, P. S. Dulepet, S. Vidyadhara, D. Ki, S. Agrawal, C. Pham, G. Kroiz, F. Li, H. Tao, A. Srivastava, H. D. Costa, S. Gupta, M. L. Rogers, I. Goncearenco, G. Sarli, I. Galynker, D. Peskoff, M. Carpuat, J. White, S. Anadkat, A. Hoyle, P. Resnik. The prompt report: A systematic survey of prompting techniques, 2024.

[59] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, UK, 1969.

[60] T. Sellam. BLEURT: a Transfer Learning-Based Metric for Natural Language Generation. https://github.com/google-research/bleurt/blob/master/README.md, 2022.

[61] T. Sellam, D. Das, A. P. Parikh. Bleurt: Learning robust metrics for text generation, 2020.

[62] D. H. Shane Peckham, Jeff Day. Getting started with LLM fine-tuning. https://learn.microsoft.com/en-us/ai/playbook/technology-guidance/generative-ai/working-with-llms/fine-tuning, 2024.

[63] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. B. Webber, T. Cohn, Y. He, Y. Liu, redaktorzy, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, strony 4222–4235, Online, List. 2020. Association for Computational Linguistics.

[64] B. E. Team. ChatGPT / OpenAI Statistics: How Many People Use ChatGPT? https://backlinko.com/chatgpt-stats, 2024.

[65] R. Tutunov, A. Grosnit, J. Ziomek, J. Wang, H. Bou-Ammar. Why can large language models generate correct chain-of-thoughts?, 2024.

[66] S. Uspenskyi. Large Language Model Statistics And Numbers. https://springsapps.com/knowledge/large-language-model-statistics-and-numbers-2024, 2024.

[67] K. Vongthongsri. LLM Benchmarks Explained: Everything on MMLU, HellaSwag, BBH, and Beyond. https://www.confident-ai.com/blog/llm-benchmarks-mmlu-hellaswag-and-beyond, 2024.

[68] A. Wang, K. Cho, M. Lewis. Asking and answering questions to evaluate the factual consistency of summaries, 2020.

[69] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, H. Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

[70] Z. Wang, S. Mao, W. Wu, T. Ge, F. Wei, H. Ji. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration, 2024.

[71] Z. M. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, J. Yang, M. Zhang, Z. Zhang, W. Ouyang, K. Xu, S. W. Huang, J. Fu, J. Peng. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models, 2024.

[72] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, W. Fedus. Emergent abilities of large language models, 2022.

[73] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[74] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi. Bertscore: Evaluating text generation with bert, 2020.

[75] Z. Zhang, Y. Yao, A. Zhang, X. Tang, X. Ma, Z. He, Y. Wang, M. Gerstein, R. Wang, G. Liu, H. Zhao. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents, 2023.

[76] Z. Zhang, A. Zhang, M. Li, A. Smola. Automatic chain of thought prompting in large language models, 2022.

[77] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, S. Eger. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance, 2019.

[78] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen. A survey of large language models, 2024.

[79] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[80] Y. Zheng, W. Gan, Z. Chen, Z. Qi, Q. Liang, P. S. Yu. Large language models for medicine: A survey, 2024.

[81] M. Zhong, Y. Liu, D. Yin, Y. Mao, Y. Jiao, P. Liu, C. Zhu, H. Ji, J. Han. Towards a unified multi-dimensional evaluator for text generation, 2022.

[82] W. Zhong, R. Cui, Y. Guo, Y. Liang, S. Lu, Y. Wang, A. Saied, W. Chen, N. Duan. Agieval: A human-centric benchmark for evaluating foundation models, 2023.

Załącznik A

Zawartość płyty CD

W tym rozdziale należy krótko omówić zawartość dołączonej płyty CD.