Pojęcia kluczowe: *prawnik, chatbot, LLM, model językowy, predykcja, AI, sztuczna inteligencja, transformer*

Jakub Kisiel

LLM to wszystko, czego prawnik potrzebuje – czyli jak działają chatboty

Abstrakt

Chatboty takie jak ChatGPT, Gemini, Copilot, Claude, Llama czy Perplexity AI stają się codziennym narzędziem pracy prawników, a przypisywane im możliwości nierzadko rozmijają się z tym, co wynika z zasady ich działania. Artykuł omawia sposób generowania odpowiedzi w ramach dużych modeli językowych (LLM) opartych na architekturze typu transformer i na tej podstawie odpowiada na pytanie, czy i w jakim zakresie mechanizm ten wyznacza granice wykorzystania tych narzędzi w pracy prawnika. Przeprowadzona analiza prowadzi do wniosku, że żadnego z opisanych ograniczeń nie da się usunąć samym sposobem pracy z chatbotem – można je jedynie łagodzić – wobec czego weryfikacja rezultatu pozostaje elementem nieusuwalnym. Ustalenia te rzutują również na ocenę tezy o zmianie paradygmatu podmiotowego, która dotyczy nie podmiotu wykonującego czynności zawodowe, lecz sposobu ich wykonywania.

I. Wstęp

Powszechność dostępu do chatbotów takich jak: ChatGPT, Gemini, Copilot, Claude, Llama, Perplexity AI czy wiele innych sprawiła, że sens pracy prawników jest kwestionowany. W szczególności jeżeli mamy do czynienia z takimi czynnościami jak analiza przepisów, przegląd orzecznictwa i literatury, pisanie pism procesowych, przygotowywanie pytań do świadków, a nawet sporządzanie uzasadnień rozstrzygnięć. W gruncie rzeczy całkiem sensownie można twierdzić, że limity zastosowania tego typu technologii wynikają raczej z ograniczeń ludzi stosujących ją niż barier technicznych. Generalnie bowiem jakość udzielonej odpowiedzi przez takiego chatbota w dużym stopniu zależy od precyzji i jakości wprowadzanego zapytania, nazywanego

 DOI: 10.54383/0031-0344.2026.08.8

promptem[1]. Innymi słowy, im lepiej zostanie zadane pytanie, tym lepsza informacja zwrotna zostanie wygenerowana. Między innymi z tego powodu powstał cały nowy obszar badawczy określany jako „prompt engineering"[2], co tym bardziej sugeruje, że w gruncie rzeczy to tutaj leży cała sztuka korzystania z tej technologii. Parafrazując Carla Gustava Junga, można stwierdzić, że „zadanie właściwego pytania jest już całością rozwiązania problemu"[3].

Te optymistyczne tendencje w zakresie wykorzystania chatbotów i innych tego rodzaju programów muszą jednak zostać ostudzone. Przypisywanie im właściwości niemalże magicznych, a przynajmniej nadludzkich, musi spotkać się z uzasadnioną krytyką. Jednym z obszarów, z którego ta krytyka może wypływać, jest oczywiście warstwa techniczna.

Problem badawczy, na którym opiera się niniejsze opracowanie, wyraża się w rozbieżności pomiędzy możliwościami przypisywanymi chatbotom w praktyce prawniczej a tymi, które wynikają z zasady ich działania. Od rozpoznania tych ostatnich zależy zarówno zakres uzasadnionego wykorzystania tej technologii w czynnościach zawodowych, jak i zakres weryfikacji, jakiej wymaga wygenerowany rezultat. Pierwszy człon tytułu niniejszego artykułu[4] jest równocześnie hipotezą, która – jak zostanie wykazane – nie znajduje potwierdzenia. W związku z tym pytanie badawcze jest następujące: czy i w jakim zakresie sposób generowania odpowiedzi przez chatboty wyznacza granice wykorzystania tych narzędzi w pracy prawnika. Odpowiedzi na nie służy wyprowadzenie ograniczeń tej technologii z mechanizmu opisanego w dalszej części. Ustalenia te mają również wymiar podmiotowy, ponieważ rzutują na ocenę tezy o zmianie paradygmatu podmiotowego w czynnościach zawodowych prawników, o czym będzie mowa w podsumowaniu. Opracowanie przy tym nie obejmuje analizy przepisów regulujących wykorzystanie tych narzędzi, a przyjęty opis techniczny ma zasadniczo charakter wysokopoziomowy i jest oparty na rozwiązaniach fundamentalnych[5].

1 T. Brown et al., *Language Models are Few-Shot Learners*, „Advances in Neural Information Processing Systems" 2020/33, s. 9; F. Yu, L. Quartey, F. Schilder, *Exploring the Effectiveness of Prompt Engineering for Legal Reasoning Tasks* (w:) *Findings of the Association for Computational Linguistics: ACL 2023*, red. A. Rogers, J. Boyd-Graber, N. Okazaki, Kanada 2023, s. 13589.

2 https://cloud.google.com/discover/what-is-prompt-engineering, https://www.ibm.com/think/topics/prompt-engineering, https://www.ibm.com/think/topics/prompt-engineering (dostęp: 16.09.2025 r.).

3 C.G. Jung, *The Archetypes and the Collective Unconscious*, USA 1959, s. 23.

4 Jest to również parafraza tytułu pracy, w której zaproponowano architekturę transformera, będącą jednym z fundamentów powszechnie dostępnych chatbotów, zob. A. Vaswani et al., *Attention Is All You Need*, 2017, arXiv:1706.03762v7 (dostęp: 16.09.2025 r.).

5 Przedstawiony w dalszej części schemat nie jest schematem uniwersalnym. Z założenia bazuje on nie na najnowszych osiągnięciach, ale raczej na pomysłach, które stanowią (lub stanowiły) fundamenty w tym obszarze. Obecnie znaczny postęp sprawia, że konkretne rozwiązania techniczne (architektury) mogą znacznie się różnić, bez wspominania nawet o odmiennościach pomiędzy modelami konkurencyjnych przedsiębiorstw. Co więcej, rywalizacja pomiędzy gigantami technologicznymi powoduje, że znaczna część rozwiązań wykorzystywanych w obecnych chatbotach jest objęta tajemnicą handlową. To wszystko nie zmienia jednak faktu, że pewne fundamentalne założenia i koncepcje w zakresie swojej istoty pozostają bez zmian.

DOI: 10.54383/0031-0344.2026.08.8

Realizacja tak określonego zadania wymaga czterech kroków. Po pierwsze, uporządkowania terminologii związanej z opisywanym obszarem, co jest konieczne do precyzyjnego oznaczenia przedmiotu analizy. Po drugie, omówienia sposobu generowania odpowiedzi, przeprowadzonego na wysokim poziomie ogólności, co z jednej strony umożliwi uchwycenie zasady działania, a z drugiej pominie skomplikowane zagadnienia matematyczne. Po trzecie, określenia ograniczeń będących konsekwencją tego mechanizmu oraz wskazania, co z nich wynika dla sposobu pracy z chatbotem i dla weryfikacji uzyskanego rezultatu. Po czwarte, wskazania ryzyk wynikających z niezrozumienia sposobu działania tej technologii oraz praktycznych konsekwencji dla czynności zawodowych.

Rozważania na ten temat są istotne z wielu względów, a brakuje publikacji, które wyprowadzałyby możliwości i ograniczenia tego oprogramowania w pracy prawników z opisu jego warstwy technicznej[6]. Pracy, która jeszcze do niedawna wydawała się niemożliwa do zastąpienia przez technologię.

II. Kwestie terminologiczne

Terminologia związana z chatbotami jest rozbudowana, zniuansowana i może się wydawać mało klarowna. Chaos pojęciowy może być źródłem wielu nieporozumień, więc wyjaśnienie kluczowych związanych z tym terminów jest bez wątpienia uzasadnione.

II.I. Sztuczna inteligencja i uczenie maszynowe

Rozważania jak zwykle wypada zacząć od początku, czyli od tego, czym jest sztuczna inteligencja, będąca bezpośrednim tłumaczeniem angielskiego pojęcia *artificial intelligence* (w skrócie AI). Choć termin ten funkcjonuje od 70 lat, to nie istnieje jego jedna powszechnie akceptowana definicja, a temat ten nie przestaje budzić kontrowersji. Co więcej, obecne są również takie głosy, że jest on – ogólnie rzecz biorąc – niepoprawny i powinno się zaniechać jego stosowania[7]. Nie wydaje się konieczne przesądzanie zasadności takiego stanowiska. Nawet zakładając, że omawiane sformułowanie jest błędnie stosowane, to powszechność tego zwrotu w dyskusji publicznej czy

6 Takich odpowiedzi nie dostarczają w szczególności: A. Bryłkowski, J. Klikowski, *Large Language Models in Legislative Content Analysis: A Dataset from the Polish Parliament*, 2025, arXiv:2503.12100 (dostęp: 16.09.2025 r.); B. Kaczmarek-Templin, *Sztuczna inteligencja (AI) i perspektywy jej wykorzystania w postępowaniu przed sądem cywilnym*, „Studia Prawnicze: Rozprawy i Materiały" 2022/2; M. Świerczyński, *Sztuczna inteligencja w prawie prywatnym międzynarodowym – wstępne rozważania*, „Problemy Prawa Prywatnego Międzynarodowego" 2019/25.

7 Np. L. Floridi, *AI as Agency without Intelligence: On Artifcial Intelligence as a New Form of Artifcial Agency and the Multiple Realisability of Agency Thesis*, „Philosophy & Technology" 2025/38.

 DOI: 10.54383/0031-0344.2026.08.8

wszelkiego rodzaju dokumentach[8] przemawia za usankcjonowaniem obecnego stanu rzeczy. Wydaje się, że dużo istotniejsze jest uchwycenie tego, o jakim zjawisku mowa, niż nieustępliwe stanie na straży pedantyzmu języka technicznego. Szczególnie jeżeli celem jest wdrożenie w powszechnie stosowaną terminologię.

Przechodząc do sedna, czyli tego, co dokładnie oznacza to pojęcie, należy zauważyć, że wielu wybitnych naukowców[9] miało wpływ na proces jego kształtowania. Według aspektów praktycznych jednak najistotniejsza jest definicja Stuarta Russella i Petera Norviga, będąca najpowszechniej cytowaną i używaną definicją sztucznej inteligencji na świecie[10]. Jej autorzy wskazują, że jest to „dziedzina badań nad agentami, którzy odbierają bodźce z otoczenia i wykonują działania"[11]. Kluczowym pojęciem jest tutaj agent, czyli system, który jest w stanie odbierać informacje i wykonywać działania[12]. W takim ujęciu agentem może być zarówno człowiek, robot, jak i oprogramowanie[13]. Rozwijając ten wątek, autorzy prezentują cztery możliwe podejścia do sztucznej inteligencji. Może ona dotyczyć: (i) działania jak człowiek, (ii) myślenia jak człowiek, (iii) racjonalnego myślenia i (iv) racjonalnego działania[14]. Racjonalność jest rozumiana jako dążenie do osiągnięcia najlepszego efektu lub, w przypadku niepewności, najlepszego oczekiwanego efektu[15] i jest w gruncie rzeczy utożsamiana z tym, co w nazwie omawianej dziedziny badań kryje się pod pojęciem inteligencji[16].

To, co jest istotne w tym wszystkim, to to, że sztuczna inteligencja jest dziedziną badań, a nie samym oprogramowaniem. W związku z tym nieprecyzyjne jest stwierdzenie, że np. ChatGPT jest sztuczną inteligencją. Stuart Russell i Peter Norvig skupiają się na podejściach akcentujących koncepcję racjonalności. Z kolei największe firmy informatyczne, definiując sztuczną inteligencję, zazwyczaj odwołują się do zdolności naśladowania ludzkich możliwości i utożsamiają ją często z nazwą grupy technologii[17].

8 Dla przykładu akt o sztucznej inteligencji – Rozporządzenie Parlamentu Europejskiego i Rady (UE) 2024/1689 z dnia 13.06.2024 r. w sprawie ustanowienia zharmonizowanych przepisów dotyczących sztucznej inteligencji oraz zmiany rozporządzeń (WE) nr 300/2008, (UE) nr 167/2013, (UE) nr 168/2013, (UE) 2018/858, (UE) 2018/1139 i (UE) 2019/2144 oraz dyrektyw 2014/90/UE (UE) 2016/797 i (UE) 2020/1828 (akt w sprawie sztucznej inteligencji) czy dokumenty OECD – https://www.oecd.org/en/publications/explanatory-memorandum-on-the-updated-oecd-definition-of-an-ai-system_623da898-en.html (dostęp: 16.09.2025 r.).

9 Warto wskazać tutaj samego pomysłodawcę Johna McCarthy'ego, ale również Alana Turinga czy Marvina Minsky'ego.

10 https://aima.cs.berkeley.edu/adoptions.html (dostęp: 16.09.2025 r.).

11 S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Harlow 2021, s. 7.

12 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 54. Chociaż warto wspomnieć, że autorzy oryginalnie posługują się pojęciem „cokolwiek" zamiast „system", jednak ta zmiana została wprowadzona dla bardziej przystępnego rozumienia podstawowych koncepcji.

13 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 54.

14 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 20–23.

15 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 22.

16 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 19.

17 https://www.ibm.com/think/topics/artificial-intelligence (dostęp: 16.09.2025 r.); https://aws.amazon.com/what-is/artificial-intelligence/ (dostęp: 16.09.2025 r.); https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-artificial-intelligence#self-driving-cars (dostęp: 16.09.2025 r.).

DOI: 10.54383/0031-0344.2026.08.8

Jedną z poddziedzin sztucznej inteligencji jest uczenie maszynowe (ang. *Machine Learning, ML*), które skupia się na poprawie wydajności poprzez doświadczenie (uczenie się)[18]. Bardziej techniczne podejście do tego zagadnienia definiuje je jako grupę metod, która umożliwia automatyczne wykrywanie wzorców w danych, w celu ich wykorzystania do przewidywania przyszłych danych lub podejmowania innych decyzji[19]. Ten obszar jest szczególnie istotny, jeśli chodzi o chatboty.

Celem powyższego wprowadzenia było jednak przede wszystkim zwrócenie uwagi na konieczność zachowania konsekwencji terminologicznej w toczącej się dyskusji, ponieważ nie wszystko w otaczającym nas świecie jest sztuczną inteligencją. Pomijając fakt, że termin ten odnosi się formalnie do dziedziny badań, to nie zmienia to faktu, że ma dosyć szerokie znaczenie. Nie oznacza to jednak, że może być używany w odniesieniu do każdej możliwej technologii.

II.II. Pojęcia bazowe: algorytm, oprogramowanie i program

Podstawowym pojęciem technicznym jest algorytm. Jest definiowany jako „abstrakcyjny sformalizowany opis procedury obliczeniowej”[20], w bardziej opisowych kategoriach jako skończona „sekwencja dokładnych instrukcji, które operują na danych”[21], albo „sekwencja kroków obliczeniowych, które przekształcają dane wejściowe w dane wyjściowe”[22]. Wynika z tego, że jest to pojęcie, które odnosi się do sposobu postępowania.

Ponieważ algorytmy nie są same w sobie zrozumiałe dla komputera[23], należy je zapisać w formie, która umożliwi mu obliczenia. Takie zestawy instrukcji, które mogą być wykonywane przez komputer, to oprogramowanie[24].

Pojęciem węższym (zakresowo) jest program, czyli realizacja algorytmu napisana w języku programowania[25]. Dwie kwestie są tutaj istotne. Programy są z jednej strony czymś więcej niż algorytmy, ponieważ obejmują jeszcze dodatkowe elementy wynikające właśnie z konieczności ich implementacji w danym języku programowania.

18 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 19; J. Franchitti, *Introduction to Computer Science*, Houston 2024, s. 21.

19 K. Murphy, *Machine Learning. A Probabilistic Perspective*, Massachusetts 2012, s. 1.

20 P. Dourish, *Algorithms and their others: Algorithmic culture in context*, „Big Data & Society” 2016/2, s. 3.

21 J. Franchitti, *Introduction to...*, s. 92.

22 T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to algorithms*, Cambridge 2009, s. 5.

23 Oczywiście nie jest to ograniczone do komputera *sensu stricto*, a raczej należy mieć na myśli wszelkie rozwiązania sprzętowe, które umożliwiają dokonywanie obliczeń (por. J. Franchitti, *Introduction to...*, s. 13). Dla zapewnienia precyzji wywodów w dalszej części posłużono się właśnie – konsekwentnie – pojęciem komputera.

24 J. Franchitti, *Introduction to...*, s. 440.

25 J. Franchitti, *Introduction to...*, s. 92.

 DOI: 10.54383/0031-0344.2026.08.8

Z drugiej strony powoduje to, że programy mogą być równocześnie związane ograniczeniami wynikającymi z reguł tego języka[26].

Oczywiście program może obejmować realizacje wielu algorytmów tak, jak jest w przypadku wielu powszechnie używanych programów, takich jak: MS Word, Google Chrome, Edge czy Adobe Acrobat Reader. Zamiennie może funkcjonować w odniesieniu do nich pojęcie aplikacji.

II.III. Architektura, model i model językowy

Powyższe pojęcia stanowią samą podstawę, która jest jednak konieczna do zagłębienia się w dalsze zagadnienia terminologiczne.

Program jest pojęciem technicznym w tym sensie, że odnosi się do szczegółowej implementacji. Pozostając w tym nurcie – jeżeli celem jest przedstawienie jej istotnych elementów, to warto posłużyć się terminem „architektura". Opisuje ona na wyższym poziomie abstrakcji główne komponenty, wzajemne relacje i metody współpracy danego programu. Jest to pewien schemat (szkielet), który opisuje, jak program działa[27]. W przypadku architektury, w centrum uwagi pozostają jednak kwestie techniczne.

Z tego względu, chcąc w większym stopniu odwołać się do matematycznych zależności działania programu, stosuje się pojęcie modelu. Co istotne, w informatyce ma on różne znaczenia w zależności od kontekstu, w jakim występuje. Kontekst, który jest istotny w perspektywie niniejszego artykułu, to model w uczeniu maszynowym[28]. Omawiane pojęcie można zdefiniować jako hipotezę dotyczącą świata, która ma za zadanie jak najlepiej dopasować się do danych. Równocześnie jest to fragment oprogramowania, który umożliwia rozwiązywanie problemów[29]. Innymi słowy, model to po prostu matematyczna reprezentacja[30], która ma za zadanie opisywać różne zjawiska[31]. Model może być jednak rozumiany na różnych poziomach szczegółowości. Po pierwsze, może się odnosić do abstrakcyjnej grupy technologii (jak modele uczenia maszynowego). Po drugie – do pewnej konkretnej grupy (np. modele językowe) albo do konkretnych przykładów modeli (jako programów), które mają już określone wszystkie parametry[32].

26 P. Dourish, *Algorithms and...*, s. 2.

27 J. Franchitti, *Introduction to...*, s. 453.

28 Warto wspomnieć, że słowo „model" może występować w wielu różnych znaczeniach i kontekstach – por. J. Franchitti, *Introduction to...*, s. 146. Oczywiście również w potocznym rozumieniu.

29 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 669; K. Murphy, *Machine Learning...*, s. 1.

30 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 232.

31 J. Franchitti, *Introduction to...*, s. 102.

32 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 684.

DOI: 10.54383/0031-0344.2026.08.8

Model definiuje działanie programu od strony matematycznej – jest to opis, w jaki sposób dane wejściowe są przekształcane w dane wyjściowe[33]. Model jest budowany przez algorytm. Natomiast architektura określa, jak ten program jest zbudowany – jaka jest jego struktura.

Specjalnymi rodzajami modeli są modele językowe (ang. *Language Models, LM*), które w pewnym uproszczeniu określają prawdopodobieństwo, z jakim kolejne słowa mogą się pojawić w danym ciągu słów. Dzięki temu można przewidzieć, jakie będą potencjalne następne słowa[34]. Modele językowe, które są trenowane na ogromnych zbiorach danych i posiadają wiele parametrów, są nazywane dużymi modelami językowymi (ang. *Large Languge Models*)[35]. Głównymi zastosowaniami modeli językowych są: generowanie tekstu, tłumaczenie maszynowe, rozpoznawanie mowy czy generowanie języka naturalnego[36].

III. Zasada działania chatbotów

Powyższe wprowadzenie terminologiczne powinno ułatwić zrozumienie działania chatbotów, które zgodnie z przytoczonymi powyżej definicjami można określić jako aplikacje (ewentualnie systemy). Analizując schemat działania tej technologii, warto postrzegać to w pewien usystematyzowany sposób. Centralnym elementem chatbotów takich jak Gemini czy ChatGPT itp. jest LLM. Innymi słowy, duży model językowy jest jednym z komponentów składowych. Jest to silnik tych aplikacji, którego celem jest generowanie tekstu na podstawie podanych danych[37]. Odpowiedź wyjściowa LLM jest funkcją tego, jak „świat" tego modelu jest w nim zdefiniowany. Ta definicja jest oczywiście wyrażona za pomocą matematycznej reprezentacji. Można jednak powiedzieć, że LLM jest „sterowany" przez całą aplikację w tym sensie, że ma ona wpływ na to, jakie dane wejściowe są mu dostarczane i jak dane wyjściowe są w dalszym ciągu wykorzystywane. Nie zmienia to jednak faktu, że centralną koncepcją stojącą za najpopularniejszymi chatbotami jest właśnie LLM[38]. Warto jednak pamiętać, że działanie tej aplikacji wymaga szeregu innych funkcji, które – technicznie rzecz ujmując – nie mieszczą się w tym komponencie.

33 W niektórych środowiskach słowa „model" używa się zamiennie z „program" – por. https://www.mathworks.com/discovery/machine-learning-models.html (dostęp: 16.09.2025 r.), https://www.coursera.org/articles/machine-learning-models (dostęp: 16.09.2025 r.).

34 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 875; K. Murphy, *Machine Learning...*, s. 591.

35 K. Thakur, H. Barker, i A.-S. Pathan, *Artificial Intelligence and Large Language Models: An Introduction to the Technological Future*, Boca Raton 2024, s. 174.

36 Ch. Wei, Y.-C. Wang, B. Wang, C.-C. Kuo, *An Overview of Language Models: Recent Developments and Outlook*, 2023, arXiv:2303.05759, s. 1.

37 H. Naveed et al., *A Comprehensive Overview of Large Language Models*, „ACM Transactions on Intelligent Systems and Technology" 2025, s. 2.

38 Z tego względu w dalszej części artykułu pojęcia chatbot, LLM i model (językowy) będą używane zamiennie.

 DOI: 10.54383/0031-0344.2026.08.8

Celem poniżej przedstawionego opisu jest odpowiedź na pytanie, w jaki sposób generowana jest odpowiedź takiego programu na zapytanie wprowadzone przez użytkownika. Co ważne jednak, niniejsze opracowanie nie ma na celu maksymalnie szczegółowego omówienia wszystkich elementów tego systemu. Przeciwnie. Jego zadaniem jest skupienie się na tych elementach, które są najistotniejsze z perspektywy **zasady działania**, z pominięciem mniej istotnych (często wyłącznie czysto technicznych zagadnień). Zacznijmy jednak od początku, czyli od zapytania stworzonego przez użytkownika.

III.I. Tokenizacja, osadzenie wektorowe i osadzenie pozycyjne

Zapytanie użytkownika wyrażone jest w języku naturalnym i jest wprowadzone do okna kontekstowego (ang. *context window*), które jest krótkotrwałą pamięcią roboczą chatbota[39]. Już w tym miejscu pojawia się pierwszy problem, ponieważ komputery, operując tylko na liczbach, nie rozumieją słów. W związku z tym konieczne jest dokonanie pewnych czynności, które umożliwią im operacje na tych słowach w sposób, który będzie (mniej lub bardziej) oddawał istotę języka.

W pierwszej kolejności niezbędne jest dokonanie zamiany zapytania (sekwencji) użytkownika na jednostki językowe, które będą zrozumiałe dla modelu. Te jednostki to tokeny, a sam proces jest określany mianem tokenizacji[40]. Prawdopodobnie najbardziej intuicyjną metodą byłoby podzielenie sekwencji wprowadzonej przez użytkownika na poszczególne słowa, jednak takie rozwiązanie ma wiele wad, gdzie przede wszystkim należy wskazać zbyt dużą liczbę elementów. Rozwiązaniem po drugiej „stronie" byłoby przyjęcie, że tokenem mogłyby być po prostu pojedyncze znaki, ale to z kolei spowodowałoby spadek wydajności modelu językowego. W efekcie często stosuje się tokenizację subwyrazową (ang. *subwords tokenization*). Można się posłużyć prostym przykładem w celu zobrazowania takiego rozwiązania. Załóżmy, że w szczególności potrzebujemy umieścić w słowniku cztery słowa: luźno, luźniej, tanio, taniej. Każde z nich ma inne znaczenie, więc co do zasady trzeba by było zastosować cztery różne pozycje w takim słowniku, żeby zakodować ich znaczenie. Można jednak przyporządkować do przyrostka „niej" znaczenie „bardziej". Wtedy te cztery słowa można zamienić (w pewnym uproszczeniu) na trzy tokeny „luźno", „tanio" oraz „niej". Słowo „luźniej" byłoby w takiej sytuacji po prostu reprezentowane przez dwa tokeny: „luźno"

39 T. Brown et al., *Language Models...*, s. 8. Warto tutaj zauważyć, że istotnym parametrem przy korzystaniu z chatbota jest rozmiar jego okna kontekstowego (wejściowego). Jego wielkość oznacza, jak dużo informacji model będzie w stanie uwzględnić w generowaniu odpowiedzi. Jeżeli zostanie udostępnionych więcej danych, to niektóre zostaną „zapomniane". Wskazać jednak można, że obecnie rozmiar okna kontekstowego sięga 750 tysięcy słów, więc przy standardowym korzystaniu szanse przekroczenia tego limitu są niewielkie, zob. https://ai.google.dev/gemini-api/docs/long-context (dostęp: 16.09.2025 r.).

40 L. Tunstall, L. von Werra, T. Wolf, *Natural Language Processing with Transformers*, Sebastopol 2022, s. 29; J. Franchitti, *Introduction to...*, s. 347; S. Russell, P. Norvig, *Artificial Intelligence...*, s. 876; Ch. Wei, Y.-C. Wang, B. Wang, C.-C. Kuo, *An Overview...*, s. 4.

i „niej". Korzyści z tego płynące są oczywiste. Tokenizacja umożliwia ograniczenie liczby pozycji w słowniku, co przekłada się na oszczędności w wykorzystywanych zasobach i energii, przy jednoczesnym zachowaniu treści przekazywanej informacji[41]. W ramach subwyrazowej wersji średnio każdym trzem słowom z zapytania użytkownika odpowiadają cztery tokeny[42].

Tokeny są jednowymiarowymi symbolami i sama taka zamiana jest niewystarczająca, ponieważ nie pozwala zakodować wielu istotnych informacji. W związku z tym w dalszej kolejności dokonywane jest ich osadzenie wektorowe (ang. *vector embedding*). Polega ono na zamianie tokena na indywidualny wektor (czyli uporządkowany ciąg wartości), co pozwala na „wzbogacenie" tego tokena o pewne dodatkowe informacje. Wektory są elementami wielowymiarowych przestrzeni wektorowych[43]. Reprezentacja tokena w wielu wymiarach umożliwia przedstawienie zależności semantycznych i syntaktycznych pomiędzy różnymi słowami[44]. Jedną z konsekwencji tej koncepcji jest to, że podobne słowa mają podobne wektory. Innymi słowy, w wielowymiarowej przestrzeni podobne słowa są blisko siebie. Dla przykładu można wskazać, że słowa „dzień" i „słońce" mogą być w niewielkiej odległości. Równocześnie przeciwieństwa do tych słów (np. „noc" i „księżyc") powinny być podobnie daleko od swoich antonimów, gdzie te ostatnie powinny być podobnie blisko siebie nawzajem[45]. Wprowadzenie wielowymiarowości pozwala na uwzględnienie szeregu różnych zależności, które mogą być bardzo przydatne do zaawansowanej analizy językowej.

Na tym etapie każdy wektor (czyli w przybliżeniu pojedyncze słowa) ma swoją reprezentację matematyczną, czyli nic innego jak ciąg wartości, które opisują go w sposób zrozumiały dla komputera. Jednym z pytań, które może się pojawić w takim miejscu, jest to, jak reprezentowane są słowa, które w zależności od kontekstu mają inne znaczenie (jak np. „zamek"). Ta kwestia została omówiona w dalszej części artykułu, ale już teraz wskazać można, że początkowo wektor jest tożsamy, ale w toku działania LLM jest on aktualizowany. W związku z tym ostatecznie token odpowiadający słowu, które może mieć różne znaczenie w zależności od kontekstu, ma ostatecznie różne reprezentacje stosownie do tego, o które znaczenie chodzi[46].

Jednak nawet takie przedstawienie nie obejmuje wszystkich istotnych informacji. Poniżej omówiono to bardziej szczegółowo, ale jedną z zasad działania chatbotów jest odejście od sekwencyjnego przetwarzania danych. Takie podejście oznacza jednak, że

41 Ch. Wei, Y.-C. Wang, B. Wang, C.-C. Kuo, *An Overview...*, s. 4; L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 32–33; H. Naveed et al., *A Comprehensive...*, s. 4.

42 https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them (dostęp: 16.09.2025 r.).

43 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 1075.

44 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 907.

45 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 907–908.

46 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 924; L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 61.

 DOI: 10.54383/0031-0344.2026.08.8

tracona jest część informacji związanych z kolejnością słów. Dla przykładu, ogromne znaczenie ma, czy powiemy, że „Ziemia porusza się dookoła Słońca" czy „Słońce porusza się dookoła Ziemi". Rozwiązaniem dla tego problemu jest osadzenie pozycyjne (ang. *positional embedding*[47]), które polega na dodaniu informacji o pozycji danego wektora (reprezentującego token) w danej sekwencji (zapytaniu użytkownika)[48]. Dzięki temu model otrzymuje informacje o tym, jakie jest względne i bezwzględne położenie danego wektora w sekwencji[49].

III.II. Predykcja tokenów – transformer

Po zakodowaniu zapytania użytkownika w powyższy sposób można przystąpić do „właściwego" przetwarzania danych. Przypomnijmy, że celem działania LLM (a co za tym idzie – chatbota) jest wygenerowanie odpowiedzi dla wprowadzonej sekwencji. Nie polega to jednak na dokonaniu tego od razu w całości. Celem tego typu modeli językowych jest dostarczanie nowych tokenów pojedynczo w oparciu o obliczone prawdopodobieństwo. W szczególności za to przewidywanie odpowiedzialny jest transformer, który stanowi fundament działania chatbotów takich jak np. ChatGPT, Gemini, Perplexity czy Claude[50].

Transformer to rodzaj architektury, który w istotny sposób wpłynął na rozwój modeli językowych dzięki zastosowaniu specjalnego mechanizmu określanego jako „samouwaga". Umożliwia on przetwarzanie różnych (powiązanych ze sobą) informacji w tym samym czasie niezależnie od tego, jak daleko są od siebie. Stanowi to alternatywne podejście względem wykrywania wyłącznie sekwencyjnych zależności pomiędzy słowami, które do tej pory było stosowane w tego typu zadaniach[51]. Innymi słowy, ten mechanizm umożliwia uwzględnienie kontekstu oddalonych od siebie słów.

LLM oparty na strukturze transformera składa się z wielu warstw, gdzie każda z nich w szczególności obejmuje (modelowo) dwa elementy. Pierwszym z nich jest wspominany mechanizm samouwagi (ang. *self-attention*), a drugim specjalna sieć (FFN, ang. *feedforward network*). Bardzo szczegółowa analiza tych zagadnień jest

47 Dla porządku wskazać należy, że zastosowanie może tutaj znaleźć wiele metod, na przykład kodowanie pozycyjne (ang. *positional encoding)*. Różnice między nimi nie mają jednak znaczenia z perspektywy przyjętego opisu, ponieważ istotna jest funkcja, jaką pełnią.

48 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 921–922; L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 73; H. Naveed et al., *A Comprehensive...*, s. 4. W celu wyjaśnienia wskazać trzeba, że obecnie stosowane są techniki takie jak np. obrotowe osadzenie pozycyjne (ang. *Rotary Position Embeddings*), por. S. Zhang, T. Parcollet, R. van Dalen, S. Bhattacharya, *Benchmarking Rotary Position Embeddings for Automatic Speech Recognition*, 2025, arXiv:2501.06051v1, s. 1. To, w jaki konkretnie sposób ta informacja jest dodawana, nie jest jednak istotne z perspektywy schematu działania.

49 L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 74.

50 M. Shao, A. Basit, R. Karri, M. Shafique, *Survey of Different Large Language Model Architectures: Trends, Benchmarks, and Challenges*, „IEEE Access" 2024/12, s. 1; K. Thakur, H. Barker, i A.-S. Pathan, *Artificial Intelligence...*, s. 173; J. Franchitti, *Introduction to...*, s. 780.

51 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 919; M. Raeini, *The Evolution of Language Models: From N-Grams to LLMs, and Beyond*, s. 8, https://ssrn.com/abstract=4625356 (dostęp: 16.09.2025 r.); M. Shao, A. Basit, R. Karri, M. Shafique, *Survey of...*, s. 2.

niepotrzebna, ponieważ dotyczy wyłącznie zaawansowanych zagadnień matematycznych. Mając jednak na względzie cel niniejszego opracowania, jakim jest wyjaśnienie zasady działania chatbotów, poniżej przestawiono krótki opis tych elementów.

III.II.I. Mechanizm samouwagi

Przed wyjaśnieniem, czym jest samouwaga, zasadne wydaje się w pierwszej kolejności wytłumaczenie, czym jest sam mechanizm uwagi (ang. *attention*). Jest to specjalna funkcja, która umożliwia wskazanie, które tokeny w sekwencji wejściowej (zapytaniu użytkownika) są istotne dla poszczególnych komponentów sekwencji wyjściowej (czyli odpowiedzi modelu). Natomiast mechanizm samouwagi pozwala na wzajemne (w ramach tej samej sekwencji) określenie, które tokeny w sekwencji są istotne dla innych tokenów w tej sekwencji[52]. Warto podać przykład działania tej funkcji. Załóżmy, że mamy zdanie „Wczoraj było pochmurno, ale dzisiaj świeci już słońce". Dzięki mechanizmowi samouwagi LLM „wie", że dla „słońce" istotne jest „świeci", ale już niekoniecznie „pochmurno". Jest to oczywiście tylko jeden z przykładów zależności, jakie mogą być „dostrzeżone" przez model.

Samouwaga pozwala np. powiązać słowo występujące na początku zdania z innym oddalonym od niego, nawet jeżeli występują daleko od siebie. W celu umożliwienia dostrzegania wielu relacji równocześnie stosuje się wielogłową uwagę (ang. *multi-head attention*), czyli wiele mechanizmów samouwagi równocześnie[53]. Dzięki temu jedna z głowic może np. dostrzec powiązanie pomiędzy podmiotem i czasownikiem, a inna – znaleźć przymiotniki opisujące rzeczownik[54]. Każda z głowic specjalizuje się bowiem w wykrywaniu innego typu zależności[55]. W skrócie można powiedzieć, że mechanizm samouwagi umożliwia przekazywanie informacji pomiędzy wektorami. Efektem działania jest zaktualizowanie dotychczasowego wektora na podstawie wszystkich innych istotnych dla niego wektorów. Właśnie w taki sposób token odpowiadający słowu „zamek" będzie zmieniał swoją reprezentację wektorową w zależności od otaczającego go kontekstu. Co więcej jednak, mechanizm samouwagi powoduje, że nawet jeżeli w dwóch różnych sekwencjach to samo słowo zostało użyte w tym samym znaczeniu, to będzie ono miało inną zindywidualizowaną reprezentację wektorową,

52 Ch. Wei, Y.-C. Wang, B. Wang, C.-C. Kuo, *An Overview...*,s. 7; M. Raeini, *The Evolution...*, s. 8; L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 4–6; S. Russell, P. Norvig, *Artificial Intelligence...*, s. 916–917.

53 Wskazać jednak trzeba, że obecnie stosuje się również takie mechanizmy jak uwagę z wieloma zapytaniami (ang. *multi-query attention*) i uwagę z pogrupowanymi zapytaniami (ang. *grouped-query attention*), por. J. Ainslie et al., *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*, 2023, arXiv:2305.13245 s. 1 (dostęp: 16.09.2025 r.).

54 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 919–920; L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 61 i 67; H. Naveed et al., *A Comprehensive...*, s. 4.

55 J. Ren et al., *Identifying Semantic Induction Heads to Understand In-Context Learning* (w:) *Findings of the Association for Computational Linguistics ACL 2024*, Bangkok 2024, s. 6924.

 DOI: 10.54383/0031-0344.2026.08.8

ale wykazującą daleko idące podobieństwo[56]. Natomiast sama wiedza o tym, które słowa (tokeny) są istotne dla innych słów (tokenów), jest zdobywana przez LLM w trakcie treningu.

III.II.II. Sieć *feedforward*

Następnie tak przetworzone wektory są w dalszym ciągu przekształcane przez sieć feedforward, czyli prostą sieć neuronową realizującą jednokierunkowy przepływ informacji[57]. Jej zadaniem jest m.in. wprowadzenie nieliniowości, odpowiadającej za rozpoznawanie skomplikowanych zależności w danych[58]. Uważa się też, że to właśnie tutaj zgromadzona jest „wiedza" LLM[59].

Takich warstw transformera jest w LLM wiele, więc powyższy proces jest wielokrotnie powtarzany[60]. Co ciekawe, głębsze warstwy wykazują tendencję do skupiania się na bardziej abstrakcyjnych zależnościach niż te płytsze (początkowe)[61].

III.II.III. Predykcja kolejnego tokena

Ostatnim etapem działania transformera jest dokonanie predykcji kolejnego tokena w sekwencji. Istotną cechą architektury transformera jest jednak to, że cały kontekst (czyli wszystkie poprzednie tokeny) jest zakodowany w ostatnim wektorze (czyli w reprezentacji ostatniego tokena). Jest to efekt działania mechanizmu samouwagi. W związku z tym po dokonaniu opisanych powyżej operacji w ramach warstw transformera generowany jest nowy wektor. Sama predykcja z kolei polega na tym, że ten wynikowy wektor jest dopasowywany do słownika LLM (podobnego do tego,

56 Między innymi z wymienionych tutaj względów chatboty bardzo dobrze sprawdzają się przy analizie dokumentów, w szczególności do wykrywania sprzeczności w postanowieniach umów czy nieścisłości terminologicznych. Mechanizm samouwagi w połączeniu z możliwością działania na dużych zbiorach danych (wynikającą z rozmiaru okna kontekstowego) pozwala na odkrywanie zależności pomiędzy fragmentami dokumentów, a to z kolei umożliwia ich identyfikację przez LLM.

57 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 802. Obecnie w ramach tej warstwy stosuje się również architekturę „połączenia ekspertów" (ang. *mixture of experts*), które polega na tym, że do predykcji każdego kolejnego tokena wykorzystywana jest tylko wydzielona część tej sieci neuronowej, tzw. ekspert. Sieć *feedforward* jest podzielona na wielu ekspertów, gdzie każdy z nich „specjalizuje" się w czymś innym. W niektórych ujęciach w każdym kolejnym przebiegu modelu ma być aktywny tylko jeden ekspert, w innych może współpracować kilku. Ogólna myśl stojąca za takim rozwiązaniem polega na tym, że aktywowanie tylko części sieci *feedforward* przyczynia się do osiągania znacznych oszczędności w zakresie mocy obliczeniowej przy zachowaniu satysfakcjonujących wyników – por. W. Cai et al., *A Survey on Mixture of Experts in Large Language Models*, 2025, arXiv:2407.06204, s. 1 (dostęp: 16.09.2025 r.); N. Dikkala et al., *On the Benefits of Learning to Route in Mixture-of-Experts Models*, „Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing" 2023, s. 1.

58 K. Thakur, H. Barker, i A.-S. Pathan, *Artificial Intelligence...*, s. 160; L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 70; S. Russell, P. Norvig, *Artificial Intelligence...*, s. 920.

59 L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 70; M. Geva, R. Schuster, J. Berant, O. Levy, *Transformer Feed-Forward Layers Are Key-Value Memories*, 2020, arXiv:2012.14913, s. 1.

60 S. Russell, P. Norvig, *Artificial Intelligence...*, s. 920; K. Thakur, H. Barker, i A.-S. Pathan, *Artificial Intelligence...*, s. 159–160.

61 O. Skean et al., *Layer by Layer: Uncovering Hidden Representations in Language Models*, „Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing" 2024, s. 7.

który służy do osadzenia wektorowego) w celu określenia, jaki wektor jest najbliżej rezultatu. Efektem końcowym jest tutaj ostatecznie rozkład prawdopodobieństwa dla poszczególnych tokenów. Model z kolei wybiera ten token, którego prawdopodobieństwo jest największe. Token jest następnie zamieniany na słowo (lub jego część), które jest potem wyświetlane użytkownikowi. Następnie LLM powtarza te wszystkie operacje dla zwiększonej już sekwencji tokenów. W ten sposób model generuje słowo po słowie, aż zostanie uznane, że należy zakończyć odpowiedź np. poprzez predykcję tokena końca sekwencji[62].

Ujmując powyższe rozważania od innej strony, można powiedzieć, że celem działania systemów opartych na architekturze transformera jest generowanie najbardziej prawdopodobnych nowych tokenów na podstawie dostarczonych mu (a w dalszej kolejności również już tych wygenerowanych) tokenów.

IV. Trening chatbota – czyli jak zdobywana jest wiedza o świecie

Sposób generowania kolejnych słów (a tak naprawdę tokenów) powoduje, że pojawia się szereg pytań. Skąd LLM wie, jakie powinno być to kolejne słowo? Jak informacje są zakodowane w modelu? Jak wiedza o świecie (hipoteza świata) jest w nim „umieszczona"?

Tak jak zostało już wspomniane, uważa się, że „wiedza" jest w większości zawarta w parametrach sieci *feedforward*. Jest to tym bardziej uzasadnione, że około 2/3 parametrów całego LLM jest umieszczone właśnie w ramach tych warstw. W związku z tym za magazynowanie informacji o prawidłowych wzorcach odpowiedzi odpowiadają liczby (parametry), które regulują zasadę działania tej sieci neuronowej.

Żeby LLM mógł prawidłowo generować odpowiedzi na zapytanie użytkownika (czyli generować kolejne relewantne tokeny), konieczne jest jego odpowiednie wytrenowanie. W taki sposób, żeby odpowiedzi były prawdziwe, wartościowe i zrozumiałe dla człowieka. W tym celu stosuje się różne techniki uczenia, które w gruncie rzeczy polegają na „pokazaniu" modelowi przykładów prawidłowych wypowiedzi, co umożliwia mu dostosowanie swoich wszystkich parametrów. Ujmując to nieco inaczej – LLM dostaje dane wejściowe i prawidłowy efekt działania, jaki powinien zostać osiągnięty[63]. Następnie model musi dopasować swoje parametry w taki sposób, żeby dla tych danych wejściowych otrzymać właśnie taki wynik. Ten proces jest jednak niezwykle czasochłonny i kosztowny[64].

62 L. Tunstall, L. von Werra, T. Wolf, *Natural Language...*, s. 58 i 75; S. Russell, P. Norvig, *Artificial Intelligence...*, s. 922.

63 Warto jednak zauważyć, że nie jest to jedyna metoda uczenia LLM.

64 Dla przykładu koszt wytrenowania GPT-4 jest szacowany na ponad 100 milionów dolarów – https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over/ (dostęp: 16.09.2025 r.).

 DOI: 10.54383/0031-0344.2026.08.8

Proces uczenia w dużej mierze opiera się na danych zgromadzonych w Internecie[65]. W związku z tym jedną z konsekwencji jest to, że siłą rzeczy lepiej zapamiętane zostaną te informacje, które częściej występują, np. z którego roku jest polski kodeks karny. Dane mniej powszechne, np. liczba sędziów w Sądzie Okręgowym w Krakowie, niekoniecznie będą dostępne przez sam model[66].

Drugą z konsekwencji z tego wynikającą jest problem z aktualnością wiedzy zgromadzonej w tym modelu. Nawet jeżeli mielibyśmy do dyspozycji LLM, który został dzień wcześniej wytrenowany[67], stan jego wiedzy (czyli wartości parametrów) byłby już w gruncie rzeczy nieaktualny, a z każdą kolejną dobą ten problem by się powiększał.

IV.I. Interpretowalność parametrów

Koniecznym elementem skutecznego działania chatbota są parametry. W związku z tym może się pojawić pytanie, czy te wartości zapisane w pamięci modelu przedstawiają dla człowieka jakiś sens. Odpowiedź musi być jednak przecząca. Konkretne wartości parametrów używanych np. w ramach mechanizmu samouwagi i sieci *feedforward* nie są w żaden sposób możliwe do zinterpretowania. Innymi słowy – to, czy w danym miejscu jest wartość x zamiast y, nie jest możliwe do wyjaśnienia w tym sensie, że nie jest to powiązane z wynikiem w wytłumaczalny dla człowieka sposób. Parametry są dobierane przez model w taki sposób, żeby jak najlepiej generować odpowiedzi. Dodatkowo liczba parametrów (zmiennych, które można dostosowywać) LLM jest ogromna. Dla przykładu GPT-4 (model wykorzystywany w chatbocie ChatGPT) ma 1,76 biliona parametrów[68]. W efekcie również szczegółowe wskazanie, jakie role pełnią poszczególne komponenty transformera, jest co najmniej trudne do zrealizowania[69].

Co ciekawe, z perspektywy matematycznej operacje wykonywane w ramach transformera od strony jednostkowej są tak naprawdę mało skomplikowane. Obliczenia

65 Co ciekawe, ten ogrom wiedzy zgromadzony w ramach LLM może zostać wykorzystany przez użytkownika do wygenerowania nowych, nieznanych mu dotąd rozwiązań problemów. Ta baza informacji obejmuje przecież doświadczenia różnych osób (w tym prawników) i użycie chatbota pozwala na „podzielenie się" nimi z użytkownikiem. Dzięki temu ta technologia może zostać wykorzystana do przygotowania propozycji rozwiązania problemów, z którymi mierzą się na co dzień prawnicy. Użytkownik przy pomocy tych informacji ma szansę wyjść poza utarte schematy i odkryć możliwości, które w normalnym toku czynności nie byłyby możliwe do poznania. W tym sensie można powiedzieć, że LLM jest kreatywny.

66 Dodatkowo warto pamiętać, że polskie dane treningowe w najbardziej popularnych modelach stanowią jedynie niewielki ułamek całego zbioru, na którym uczy się model. Z tego względu bardzo szczegółowe zagadnienia prawnicze mogą niejednokrotnie być błędnie analizowane przez LLM.

67 Co jest samo w sobie mało prawdopodobne, ponieważ dla przykładu trenowanie GPT-4 zajęło 3 miesiące – https://klu.ai/blog/gpt-4-llm (dostęp: 16.09.2025 r.).

68 L. Wang et al., *Parameter-efficient fine-tuning in large language models: a survey of methodologies*, „Artificial Intelligence Review" 2025/58, s. 227.

69 S. Sonkar, R. Baraniuk, *Investigating the Role of Feed-Forward Networks in Transformers Using Parallel Attention and Feed-Forward Net Design*, 2023, arXiv:2305.13297, s. 1.

te opierają się na prostych działaniach na wektorach takich jak mnożenie, sumowanie itp. To, co sprawia, że całościowo ten mechanizm jest złożony, to właśnie rozmiar (wielość parametrów) LLM.

IV.II. Zewnętrzne bazy danych i narzędzia

W celu znalezienia rozwiązania na te oraz inne problemy związane z dostępem do informacji przez LLM umożliwiono mu korzystanie z różnych narzędzi, jednym z nich jest możliwość wyszukiwania informacji w Internecie. Korzystanie z wyszukiwarek internetowych przez chatboty od strony technicznej zazwyczaj opiera się na jednej z dwóch możliwości. Po pierwsze, model może zwrócić się z zapytaniem do zewnętrznego modułu RAG (ang. *retrieval-augmented generation*), który sam wykonuje wyszukiwanie, a następnie zwraca mu znalezione informacje. Po drugie, można nauczyć LLM samodzielnego korzystania z wyszukiwarek internetowych. W jednym i drugim przypadku końcowym efektem jest dołączenie uzyskanych danych do początkowego okna kontekstowego modelu. Dzięki temu generacja tokenów uwzględnia te informacje już na starcie – w szczególności dzięki wykorzystaniu mechanizmu samouwagi[70]. W efekcie model już bazowo ma zaktualizowane informacje i to na nich w głównej mierze opiera swoje dalsze predykcje.

W tym miejscu warto skupić się również na wspomnianym module RAG, ponieważ rola tego komponentu nie jest ograniczona do korzystania z wyszukiwarek internetowych. Ogólnym założeniem stojącym za tego typu rozwiązaniem jest chęć dołączenia zewnętrznej względem LLM bazy danych, w taki sposób, żeby informacje tam zawarte miały wpływ na odpowiedzi generowane przez model. Takie podejście ma wiele korzyści. Umożliwia uwzględnienie wiedzy specjalistycznej (np. z zakresu prawa karnego) czy ograniczenie halucynacji modelu, o czym w dalszej części.

Wyszukiwanie i zewnętrzne bazy danych nie wyczerpują przy tym katalogu narzędzi. LLM może również dla przykładu wykonywać obliczenia przy użyciu kalkulatora albo uruchamiać kod (najczęściej w języku Python). Rezultat nie jest wówczas efektem predykcji kolejnych tokenów, lecz działania odrębnego programu, którego wynik zostaje dołączony do okna kontekstowego. Ma to znaczenie dla zadań, o których w dalszej części.

V. Ograniczenia chatbotów, ryzyka i konsekwencje dla praktyki

Sposób generowania odpowiedzi przez chatboty nie jest pozbawiony wad. Zdawanie sobie sprawy z ograniczeń LLM jest o tyle istotne, że pozwala na świadome

70 B. Jin et al., *Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning*, 2025, arXiv:2503.09516, s. 1–2.

 DOI: 10.54383/0031-0344.2026.08.8

i prawidłowe korzystanie z tej technologii. Nieznajomość tego mechanizmu prowadzi natomiast do przypisywania chatbotom właściwości, których nie mają, a w konsekwencji do pomijania weryfikacji tam, gdzie jest ona konieczna.

Sam fakt, że mowa tutaj o modelach **językowych**, powinien uzmysławiać, że jednym z docelowych zadań tej technologii jest właśnie operowanie na języku. Dlatego też mogą okazać się nieocenione w zakresie streszczania tekstów[71], jak również rozbudowywania czy tłumaczenia ich fragmentów. Co więcej, zakres kontekstu, jaki człowiek wykorzystuje w analizie danej sytuacji, czasami jest ograniczony. Z tego względu warto pamiętać, że interpretacja postanowień umowy, decyzji organów czy innych dokumentów może być dokonywana przy pomocy chatbota. Niejednokrotnie może on „wyłapać" dodatkowy kontekst, który mógł umknąć człowiekowi. Należy jednak pamiętać, że działa to również w drugą stronę. Niektóre elementy mogą być pomijane przez tę technologię, a zauważone dopiero przez człowieka. Trzeba sobie również zdawać sprawę, że odpowiedź chatbota stanowi wypowiedź skierowaną do użytkownika, a nie fragment pisma czy uzasadnienia. Jej wykorzystanie wymaga zatem usunięcia elementów zwrotu do adresata – pytań o dalsze oczekiwania, propozycji zmiany stylu wypowiedzi i podobnych uwag – których pozostawienie w treści dokumentu zdarza się w praktyce.

Ważnym elementem mechanizmu generowania odpowiedzi jest tokenizacja. Powoduje ona, że LLM postrzega ciągi znaków odmiennie niż człowiek. Tokeny nie są zazwyczaj ani pojedynczymi literami, ani pełnymi słowami. To, co dla użytkownika może wydawać się oczywiste, dla LLM niekoniecznie jest. Z tego względu wszelkie operacje na słowach niejednokrotnie mogą stanowić wyzwanie dla modelu. Dotyczy to w szczególności liczb i dat, które są rozpatrywane jako ciągi znaków, a nie jako wartości. Wynika z tego, że wszystkie zadania związane z obliczaniem terminów, kwot itp. niosą ze sobą zwiększone ryzyko pomyłki. Dodatkowo warto pamiętać, że sformułowanie zapytania w taki sposób, który wymusza na modelu znalezienie prawidłowej odpowiedzi w niewielkiej liczbie tokenów, jest niepożądaną praktyką[72]. Ograniczenie to nie dotyczy natomiast sytuacji, w których model korzysta z zewnętrznych narzędzi

71 Streszczenie pozostaje jednak wypowiedzią wygenerowaną, a nie wyodrębnioną z dokumentu, więc może zawierać zdania, których w nim nie było.

72 Co wynika z faktu, że LLM ma ograniczone możliwości obliczeniowe w ramach jednego tokena. Prosty przykład. Jeżeli nakażemy chatbotowi rozwiązać skomplikowane zadanie matematyczne i zastrzeżemy, że musi to zrobić w jednym tokenie, to najprawdopodobniej odpowiedź będzie niepoprawna. Natomiast jeżeli LLM może „rozbić" te obliczenia na kilka iteracji, to powinniśmy otrzymać prawidłowy wynik. W celu umożliwienia LLM generowania odpowiedzi „na raty" stosuje się tzw. „rozumowanie" lub „myślenie", co najczęściej obejmuje tzw. łańcuch myśli (*ang. Chain of Thought, CoT*), polegający na tym, że model ma za zadanie rozbić rozwiązanie na kilka kroków – por. J. Wei et al., *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*, 2023, arXiv:2201.11903, s. 2 (dostęp: 16.09.2025 r.). Ta technika może być niezwykle przydatna dla prawnika w tych przypadkach, gdy sprawa jest skomplikowana, a dokonywanych jest wiele wzajemnie powiązanych wnioskowań. Pozwala to skontrolować poprawność poszczególnych kroków i w razie potrzeby interweniować. Może to również zostać wykorzystane do krytycznej analizy własnego rozumowania użytkownika. Z punktu widzenia działania modelu czym innym jest polecenie „zaproponuj rozwiązanie problemu" a czym innym „oceń proponowane rozwiązanie". Nie zmienia to jednak faktu, że zarówno w jednym, jak i drugim przypadku wkład chatbota może być nieoceniony.

DOI: 10.54383/0031-0344.2026.08.8

– wynik obliczenia nie jest wówczas efektem predykcji tokenów, lecz działania odrębnego komponentu (narzędzia), o którym była mowa powyżej. Skorzystanie z niego można polecić chatbotowi wprost.

Przy obliczeniach – terminach, kwotach, odsetkach – warto polecić chatbotowi wykonanie ich przy użyciu kalkulatora lub kodu albo co najmniej rozbicie na kolejne kroki. W pierwszym wypadku wynik nie pochodzi z generowania tekstu, w drugim – daje się skontrolować krok po kroku

Dodatkowo zapytanie użytkownika jest w gruncie rzeczy liniowym ciągiem symboli – tokenów, a generowanie odpowiedzi przebiega od lewej do prawej. Z tego względu warto formułować je tak, aby informacje pojawiały się po kolei i w uporządkowany sposób, co pozwoli na dokonanie relewantnej analizy przez model. Konsekwencją tego jest również to, że jeżeli model ustalił pewien wniosek jeszcze przed przeprowadzeniem właściwego rozumowania, to ten wniosek staje się automatycznie kontekstem dla dalszej części wypowiedzi oraz kolejnych zapytań. W efekcie może on w istotny sposób wpływać na poprawność ostatecznej analizy. W praktyce najlepiej, żeby sama konkluzja była efektem analizy, a nie pierwszym krokiem działania chatbota.

Jednym z najważniejszych wyzwań związanych z korzystaniem z chatbotów są halucynacje[73], czyli takie odpowiedzi LLM, które są bezsensowne lub niezgodne z dostarczonymi danymi źródłowymi, gdzie niejednokrotnie są one równocześnie niezwykle przekonujące[74]. Zasadnicze niebezpieczeństwo polega na tym, że model nie sygnalizuje własnej niepewności. Halucynacje nie różnią się formą od odpowiedzi prawidłowych – są równie stanowcze i równie precyzyjne. Inaczej niż w wypowiedzi człowieka, brak wiedzy nie pozostawia tu zwykle żadnego śladu. Jeśli jednak wziąć pod uwagę sposób generowania danych wyjściowych przez model, ich występowanie nie powinno dziwić. Każdy kolejny token jest jedynie najbardziej prawdopodobnym tokenem występującym w danym kontekście. Oznacza to w szczególności, że jeżeli jakaś informacja rzadko występuje w danych treningowych (czyli w niewielkim stopniu została zapamiętana przez model), to istnieje duża szansa, że model „nie będzie pewny" odpowiedzi. W związku z tym, zamiast informować o braku wiedzy, będzie generował dane, które są najbardziej prawdopodobne – czyli niekoniecznie prawdziwe. Trening może zwiększyć częstotliwość sygnalizowania niepewności, samego zjawiska jednak nie usuwa[75].

Problem halucynacji objawia się w szczególności przy wykorzystywaniu chatbotów do wyszukiwania orzecznictwa. Niejednokrotnie zdarza się, że orzeczenie

73 Y. Gao et al., *Retrieval-Augmented Generation for Large Language Models: A Survey*, 2023, arXiv:2312.10997, s. 1.

74 L. Huang et al., *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*, „ACM Transactions on Information Systems" 2025/43, s. 1–2.

75 Obecnie uważa się, że halucynacje są w szczególności efektem sposobów trenowania modeli, które premiują „pewność siebie" zamiast przyznania się do niewiedzy – por. A. Kalai, O. Nachum, E. Zhang, S. Vempala, *Why Language Models Hallucinate*, 2025, arXiv:2509.04664v1, s. 2–4 (dostęp: 16.09.2025 r.).

 DOI: 10.54383/0031-0344.2026.08.8

o podanej sygnaturze nie występuje albo co prawda istnieje, ale jego treść jest zupełnie inna. W drugim wypadku błąd przechodzi kontrolę polegającą na sprawdzeniu, czy sygnatura występuje w bazie – weryfikacji podlega zatem teza względem orzeczenia, a nie sygnatura względem bazy. Jeżeli nie ma wyroku, który odpowiada treściowo zapytaniu użytkownika, model może bez problemu go wygenerować. Mechanizm ten nie zapewnia przy tym wiernego odtworzenia treści dokumentu – informacje znalezione w wyszukiwarce zasilają jedynie kontekst, na którym model opiera dalszą predykcję, m.in. w ramach mechanizmu samouwagi. Cytat może więc zostać po prostu wygenerowany. Sposobami na ograniczanie halucynacji pozostają wykorzystanie RAG oraz wprost nakazywanie chatbotowi weryfikacji orzeczeń[76]. Nie dają one jednak gwarancji skuteczności, ponieważ problem tkwi w samej istocie działania modelu. Wykrycie halucynacji pozostaje zatem ostatecznie po stronie prawnika. To samo dotyczy przeglądu literatury, gdzie wygenerowane mogą zostać zarówno treści, tytuły, nazwiska autorów czy numery stron.

Kolejna kwestia to sam sposób konstruowania wypowiedzi. Model nie generuje jej jako spójnej całości w tym sensie, że każdy kolejny token jest jedynie wynikiem obliczeń matematycznych mających na celu określenie prawdopodobieństwa jego wystąpienia w danej sekwencji. Innymi słowy – nawet jeżeli jakaś informacja jest dla użytkownika oczywista ze względu na poprzednio wprowadzone słowa i kontekst, w jakim pracuje, to niekoniecznie tak jest w przypadku LLM[77]. Z tego względu okoliczności, które użytkownik uznaje za oczywiste z uwagi na kontekst swojej pracy – dziedzinę prawa, stan sprawy, cel analizy – powinny zostać wprowadzone do zapytania wprost. Jeżeli użytkownik tego nie zrobi, to model samodzielnie uzupełni tę lukę na podstawie najbardziej prawdopodobnych kontekstów, jednak najczęściej nie poinformuje o przyjętych założeniach.

Okoliczności istotne dla analizy – w szczególności gałąź prawa i jej cel – najlepiej wskazać wprost, także wtedy, gdy w kontekście pracy wydają się oczywiste. Ich pominięcie może nie zostać zasygnalizowane, a model uzupełni lukę samodzielnie, przyjmując najbardziej prawdopodobne założenie

Warto również zwrócić uwagę, że pamięć krótkotrwała modelu (okno kontekstowe) jest ograniczona (mimo że obecnie może być dosyć duża[78]). Jeżeli jakieś informacje są szczególnie istotne, to warto je wprowadzić właśnie do okna kontekstowego. Równocześnie nie można zapominać, że może to pozostawać w kolizji z obowiązkiem

76 Zmieniają one dane wejściowe modelu, a nie zasadę wyboru tokenów.

77 Eksperyment, który można tutaj przeprowadzić, mógłby wyglądać następująco. Nasze zapytanie wprowadzone do chatbota brzmi: „Mówimy o prawie karnym. Dokończ następujące zdanie w dwóch słowach: Nie można chcieć i…" Dla prawnika, który ukończył kurs prawa karnego, w oczywisty sposób nasuwa się skojarzenie „godzić się". LLM natomiast udziela takich odpowiedzi, jakie wynikają z jego statystycznych wyliczeń, czyli np. „nie działać" czy „nie obejmować".

78 To z kolei sprzyja analizie dużych zbiorów dokumentów, która przebiega znacznie szybciej niż ta dokonywana przez człowieka.

ochrony informacji, co ma istotne znaczenie w pracy prawnika. Szczegółowe rozważania w tym zakresie wykraczają jednak poza ramy niniejszego opracowania.

Warto wspomnieć tutaj o zjawisku sykofancji (ang. *sycophancy*), które przejawia się tym, że LLM mają tendencję do zgadzania się z użytkownikiem, nawet jeżeli „wiedzą", że nie ma on racji[79]. W związku z tym tworzenie zapytań w postaci „czy mam rację, twierdząc, że…" może być obarczone większym ryzykiem błędu. Jeżeli celem ma być krytyczna analiza, to lepszym rozwiązaniem jest takie przedstawienie problemu, które nie sugeruje poprawnej odpowiedzi (zdaniem użytkownika). Innymi słowy – „zachęcanie" modelu do krytycyzmu albo do przedstawienia argumentów za i przeciw może przynieść dużo lepsze rezultaty, ponieważ nie będzie on związany punktem widzenia użytkownika. Umożliwia to testowanie własnej argumentacji pod kątem jej słabych stron.

Zapytanie sugerujące oczekiwaną odpowiedź zwiększa ryzyko błędu. Jeżeli celem jest krytyczna ocena, problem warto przedstawić bez wskazywania własnego stanowiska

Modele językowe w założeniu są w stanie wykonywać szereg różnych czynności, do których niekoniecznie zostały wprost zaprogramowane. Natomiast jeżeli dana czynność jest rzadko spotykana (w danych, na których model się uczył) albo jeżeli użytkownikowi zależy na konkretnym formacie odpowiedzi, to warto skorzystać z praktyki polegającej na „pokazywaniu" chatbotowi, jak wygląda przykładowa prawidłowa odpowiedź na zadane pytanie. W ten sposób model będzie się na bieżąco dostosowywał do zadanego schematu[80]. Jest to szczególnie przydatne wtedy, gdy pismo ma odpowiadać określonemu formatowi, stylowi, a nawet przyjętemu sposobowi konstruowania argumentacji. Równocześnie warto pamiętać, że może to replikować błędy zawarte we wzorcu. Efektem ubocznym może być też nadmierne dopasowanie odpowiedzi do wzorca. Przy zadaniach takich jak przygotowywanie pytań do świadków może to prowadzić do pominięcia elementów istotnych w danej sprawie, jako nieobecnych w przedstawionym przykładzie.

Warto mieć również na względzie, że zazwyczaj lepsze rezultaty działania są osiągane, jeżeli informacje, na jakich ma bazować LLM, są bezpośrednio dołączane zamiast odwoływania się do wiedzy zgromadzonej w parametrach modelu. Innymi słowy, jeżeli celem ma być analiza jakiegoś konkretnego przepisu Kodeksu karnego, to lepiej bezpośrednio umieścić treść tego przepisu w oknie kontekstowym lub załączyć tekst ustawy. Co więcej, wiedza zgromadzona w parametrach modelu pochodzi

79 M. Sharma et al., *Towards Understanding Sycophancy in Language Models*, 2023, arXiv:2310.13548, s. 1; L. Malmqvist, *Sycophancy in Large Language Models: Causes and Mitigations*, 2024, arXiv:2411.15287, s. 1 (dostęp: 16.09.2025 r.).

80 Ostateczny efekt jest podobny do działania RAG, z tą różnicą, że nie ma konieczności budowania odrębnego modułu. W praktyce może to polegać na dostarczaniu modelowi dokumentów jako załączników w oknie kontekstowym.

 DOI: 10.54383/0031-0344.2026.08.8

z momentu treningu, a jej nieaktualność może nie zostać zasygnalizowana – ze względu na częstotliwość nowelizacji należy zatem załączać aktualne brzmienie przepisów.

Akt prawny, umowę czy orzeczenie najlepiej załączyć w aktualnym brzmieniu. Nieaktualność wiedzy zgromadzonej w parametrach modelu może nie zostać zasygnalizowana

Jedną z konsekwencji wykorzystania koncepcji okna kontekstowego jest to, że każde kolejne zapytanie osadzone w danej konwersacji z chatbotem jest rozpatrywane w kontekście poprzednich. W związku z tym, jeżeli w pierwszej kolejności zapytamy o znaczenie jednego przepisu Kodeksu karnego, to kolejne zapytanie o inny przepis, chociażby nie wskazywało *explicite*, że mowa o ustawie karnej, i tak najprawdopodobniej tak zostanie odczytane. Oznacza to, że jeżeli chcemy „wyzerować" pamięć konwersacji, to warto rozpocząć nowy czat.

Warto również pamiętać, że skoro chatbot w gruncie rzeczy „chce" jedynie przewidzieć najbardziej prawdopodobne kolejne słowo (a tak naprawdę token), to nie ma żadnych gwarancji, że będzie ono miało sens. Jest to tylko wynik działania złożonej funkcji matematycznej, która ze względu na ogromną ilość parametrów w praktyce zdaje się dobrze odwzorowywać wzorce językowe. Oczywiście wielość parametrów jest związana z wielością informacji nauczonych przez taki model, które to informacje są reprezentacją pewnej bazy wiedzy. Ujmując to w skrócie – to właśnie ogromna liczba zmiennych sprawia, że odpowiedzi generowane przez model wydają się sensowne i bardzo dobrze naśladują ludzki sposób konstruowania zdań. Są to jednak nadal wyłącznie liczby, które odpowiednio wykorzystane po prostu okazały się świetnie działać. Mechanizm ten zapewnia poprawność językową wypowiedzi, nie zaś jej trafność merytoryczną.

VI. Podsumowanie

Powyższa analiza jednoznacznie przesądza, że chatboty to w gruncie rzeczy bardzo zaawansowane predyktory tekstu, które, dzięki ekspozycji na ogromne ilości danych, są w stanie przekonująco przewidywać całe wypowiedzi. Sam mechanizm działania jest bez wątpienia skuteczny, jednak nie można tracić z pola widzenia sposobu, w jaki każdy kolejny token jest wybierany. Podstawą są nieskomplikowane operacje matematyczne, które bazują na koncepcji wektorowego przedstawienia tokenów. Ich osadzenie w wielowymiarowej przestrzeni pozwala odwzorować zależności semantyczne i syntaktyczne, choć same wartości parametrów, na których opierają się dalsze obliczenia, pozostają dla człowieka nieinterpretowalne. Nie ulega wątpliwości, że tworzenie tekstu jest mocną stroną tej technologii, jednak w istocie za tym wszystkim kryje się prawdopodobieństwo. Wniosek z tego płynący wydaje się następujący. Chatboty w rękach prawników powinny być jedynie nadzorowanym narzędziem, ponieważ to,

DOI: 10.54383/0031-0344.2026.08.8

co tworzą, nawet jeżeli zawiera kontekstowe odniesienie, to nie wyraża wprost sensu czy abstrakcji. Mechanizm tworzenia odpowiedzi przez LLM jest czysto obliczeniowy.

Pozwala to odpowiedzieć na postawione we wstępie pytanie badawcze. Część ograniczeń da się usunąć sposobem pracy z chatbotem – nieaktualność wiedzy zgromadzonej w parametrach przez załączenie aktualnego brzmienia przepisu, a zawodność obliczeń przez zlecenie ich zewnętrznemu narzędziu. Nie da się natomiast usunąć ograniczenia wynikającego z samej zasady generowania odpowiedzi, ponieważ model dobiera kolejne tokeny według prawdopodobieństwa i nie sygnalizuje własnej niepewności. Odpowiednie formułowanie zapytań i wymuszanie kolejności rozumowania zmniejszają ryzyko błędu, lecz zasady tej nie zmieniają. Weryfikacja rezultatu pozostaje więc elementem nieusuwalnym, a użyteczność tej technologii kończy się tam, gdzie koszt tej weryfikacji przekracza koszt samodzielnego wykonania zadania. Pierwszy człon tytułu, stanowiący hipotezę opracowania, nie znajduje zatem potwierdzenia.

W efekcie o zmianie paradygmatu podmiotowego w czynnościach zawodowych prawników można mówić w tym sensie, że nie dotyczy ona samego podmiotu, lecz sposobu wykonywania tych czynności[81]. Prawnik w mniejszym stopniu jest twórcą treści, a w większym podmiotem inicjującym jej generowanie oraz weryfikującym rezultat. Weryfikacja ta różni się od kontroli pracy powierzonej człowiekowi tym, że sprawdzeniu podlegają jedynie przejawy działania tej technologii, a nie sposób dojścia do wyniku. W tym świetle wymaga rewizji przywołane we wstępie spostrzeżenie, że zadanie właściwego pytania jest już całością rozwiązania problemu. Pozostaje ono istotną częścią rozwiązania, lecz jedynie częścią – drugą jest sprawdzenie uzyskanej odpowiedzi, którego nie da się przenieść na narzędzie.

mgr inż. Jakub Kisiel

Doktorant w Szkole Doktorskiej Nauk Społecznych Uniwersytetu Jagiellońskiego.

PhD student at a Doctoral School in the Social Sciences, Jagiellonian University.

ORCID: 0000-0002-5474-6954

81 Por. § 23e Zbioru Zasad Etyki Adwokackiej i Godności Zawodu (Kodeks Etyki Adwokackiej) dodany uchwałą nr 41/2026 Naczelnej Rady Adwokackiej z dnia 12.06.2026 r.

DOI: 10.54383/0031-0344.2026.08.8

Abstract

Keywords: *lawyer, chatbot, LLM, language model, prediction, AI, artificial intelligence, transformer*

LLM is all a lawyer needs – how chatbots work

Chatbots such as ChatGPT, Gemini, Copilot, Claude, Llama and Perplexity AI are becoming an indispensable tool in the work of lawyers. For some, their capabilities seem almost limitless, which – coupled with the belief that this technology is infallible – can give rise to many concerns. In order to have a meaningful discussion on this topic, it is necessary to understand how these systems work and what their limitations are. The purpose of this article is to discuss the principles of operation of this type of technologies, with particular emphasis on how responses are generated within large language models (LLMs) based on transformer architecture. This will allow us to specify to what extent and subject to what limitations chatbots can be used to assist lawyers. However, this analysis also aims to identify the reasons for these obstacles, which will make it possible to determine the strengths and weaknesses of the technology in question. Knowledge of these weaknesses is useful, if only because they can be mitigated to some extent by using appropriate techniques.

Bibliografia

1. **Ainslie Joshua et al.,** *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*, 2023, arXiv:2305.13245 (dostęp: 16.09.2025 r.)
2. **Brown Tom et al.,** *Language Models are Few-Shot Learners*, „Advances in Neural Information Processing Systems" 2020/33
3. **Bryłkowski Arkadiusz, Klikowski Jakub,** *Large Language Models in Legislative Content Analysis: A Dataset from the Polish Parliament*, 2025, arXiv:2503.12100 (dostęp: 16.09.2025 r.)
4. **Cai Weilin et al.,** *A Survey on Mixture of Experts in Large Language Models*, 2025, arXiv:2407.06204 (dostęp: 16.09.2025 r.)
5. **Carl Gustav Jung,** *The Archetypes and the Collective Unconscious*, USA 1959
6. **Cormen Thomas, Leiserson Charles, Rivest Ronald, Stein Clifford,** *Introduction to algorithms*, Cambridge 2009
7. **Dikkala Nishanth et al.**, *On the Benefits of Learning to Route in Mixture-of-Experts Models*, „Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing" 2023

DOI: 10.54383/0031-0344.2026.08.8

8. **Dourish Paul,** *Algorithms and their others: Algorithmic culture in context*, „Big Data & Society" 2016/2
9. **Floridi Luciano,** *AI as Agency without Intelligence: On Artifcial Intelligence as a New Form of Artifcial Agency and the Multiple Realisability of Agency Thesis*, „Philosophy & Technology" 2025/38
10. **Franchitti Jean-Claude,** *Introduction to Computer Science*, Houston 2024
11. **Gao Yunfan et al.,** *Retrieval-Augmented Generation for Large Language Models: A Survey*, 2023, arXiv:2312.10997
12. **Geva Mor, Schuster Roei, Berant Jonathan, Levy Omar,** *Transformer Feed-Forward Layers Are Key-Value Memories*, 2020, arXiv:2012.14913 2021
13. **Huang Lei et al.,** *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*, „ACM Transactions on Information Systems" 2025/43
14. **Jin Bowen et al.,** *Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning*, 2025, arXiv:2503.09516
15. **Kaczmarek-Templin Berenika,** *Sztuczna inteligencja (AI) i perspektywy jej wykorzystania w postępowaniu przed sądem cywilnym*, „Studia Prawnicze: Rozprawy i Materiały" 2022/2
16. **Kalai Adam, Nachum Ofir, Zhang Edwin, Vempala Santosh,** *Why Language Models Hallucinate*, 2025, arXiv:2509.04664v1
17. **Malmqvist Lars,** *Sycophancy in Large Language Models: Causes and Mitigations*, 2024, arXiv:2411.15287
18. **Murphy Kevin,** *Machine Learning. A Probabilistic Perspective*, Massachusetts 2012
19. **Naveed Humza et al.,** *A Comprehensive Overview of Large Language Models*, „ACM Transactions on Intelligent Systems and Technology" 2025
20. **Raeini Mohammad,** *The Evolution of Language Models: From N-Grams to LLMs, and Beyond*, https://ssrn.com/abstract=4625356 (dostęp: 16.09.2025 r.)
21. **Ren Jie et al.,** *Identifying Semantic Induction Heads to Understand In-Context Learning* (w:) *Findings of the Association for Computational Linguistics ACL 2024*, Bangkok 2024
22. **Russell Stuart, Norvig Peter,** *Artificial Intelligence: A Modern Approach*, Harlow 2021
23. **Shao Minghao, Basit Abdul, Karri Ramesh, Shafique Muhammad,** *Survey of Different Large Language Model Architectures: Trends, Benchmarks, and Challenges*, „IEEE Access" 2024/12
24. **Sharma Mrinank et al.,** *Towards Understanding Sycophancy in Language Models*, 2023, arXiv:2310.13548
25. **Skean Oscar et al.,** *Layer by Layer: Uncovering Hidden Representations in Language Models*, „Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing" 2024

 DOI: 10.54383/0031-0344.2026.08.8

26. **Sonkar Shashank, Baraniuk Richard,** *Investigating the Role of Feed-Forward Networks in Transformers Using Parallel Attention and Feed-Forward Net Design*, 2023, arXiv:2305.13297
27. **Świerczyński Marek,** *Sztuczna inteligencja w prawie prywatnym międzynarodowym – wstępne rozważania*, „Problemy Prawa Prywatnego Międzynarodowego" 2019/25
28. **Thakur Kutub, Barker Helen, Pathan Al-Sakib,** *Artificial Intelligence and Large Language Models: An Introduction to the Technological Future*, Boca Raton 2024
29. **Tunstall Lewis, von Werra Leandro, Wolf Thomas,** *Natural Language Processing with Transformers*, Sebastopol 2022
30. **Vaswani Ashish et al.,** *Attention Is All You Need*, 2017, arXiv:1706.03762v7
31. **Wang Luping et al.,** *Parameter-efficient fine-tuning in large language models: a survey of methodologies*, „Artificial Intelligence Review" 2025/58
32. **Wei Chengwei, Wang Yun-Cheng, Wang Bin, Kuo C.-C.,** *An Overview of Language Models: Recent Developments and Outlook*, 2023, arXiv:2303.05759
33. **Wei Jason et al.,** *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*, 2023, arXiv:2201.11903
34. **Yu Fangyi, Quartey Lee, Schilder Frank,** *Exploring the Effectiveness of Prompt Engineering for Legal Reasoning Tasks* (w:) *Findings of the Association for Computational Linguistics: ACL 2023*, red. A. Rogers, J. Boyd-Graber, N. Okazaki, Kanada 2023
35. **Zhang Shucong, Parcollet Titouan, van Dalen Rogier, Bhattacharya Sourav,** *Benchmarking Rotary Position Embeddings for Automatic Speech Recognition*, 2025, arXiv:2501.06051v1