

# Generative AI: technologiczna bańka spekulacyjna?

W marcu 2023 roku napisałem w serwisie Linked.in post przeciwstawiający się śmiałym prognozom rozwoju sztucznej inteligencji w zakresie pisania oprogramowania. Spotkałem się tam bowiem z wieloma stwierdzeniami pokroju „Za pięć lat nie będziemy już pisać oprogramowania!”. Tymczasem bazując na mojej ówczesnej wiedzy na temat generatywnej sztucznej inteligencji, wydało mi się, że próby ekstrapolowania jej rozwoju do takiego poziomu wydają się być mocno przesadzone.

Stwierdziłem wtedy:

„Za 5 lat nie będziemy już pisać kodu!”, głoszą posty na Linked.in.

Zalóżmy się. Mówię serio. Chętnie przyczepię darmową kasę.

Pomówmy o słońcu w pokoju, dobrze? ChatGPT jest naprawdę fascynujące i szczerze podziwiam jego twórców. Ale pamiętajmy, że jest ono narzędziem, które po pierwsze działa na zasadzie usprawiedliwionego zgadywania (w gigantycznym uproszczeniu mniej więcej właśnie tak działają sieci neuronowe), a po drugie, bazując na przygotowanym wcześniej modelu. Jeżeli ChatGPT na oczy nie widziało kodu źródłowego, dajmy na to, oprogramowania do sterowania pralką, to choćby i na cyfrowej głowie stanęło, nie będzie w stanie takiego oprogramowania napisać. A to oznacza, że będzie to musiał zrobić programista.

Ciekawostka: jakiś miesiąc temu zapytałem ChatGPT, ile to jest 123\*567. Odpowiedziało, że 70041. 123\*567 to nie jest 70041, tylko 69741. Zapytane dziś podało już prawidłową odpowiedź, natomiast zżera mnie ciekawość, czy dlatego, że jego twórcy zaimplementowali deterministycznie obliczanie wyrażeń matematycznych, bo śmiem twierdzić, że samą siecią neuronową tego zrobić się nie da.

Przyjmijmy na potrzeby przykładu, że za 5 lat ChatGPT faktycznie będzie pisało całe aplikacje.

Zalóżmy teraz, że ChatGPT napisze program, w którym znajdzie się błąd. Jednak z modelu, z którego ChatGPT skorzysta, może wynikać, że implementacja jest prawidłowa – może tak się zdarzyć (patrz wyżej). Kto będzie musiał naprawić problem? Programista.

I jeszcze dalej. Powiedzmy, że w przyszłości jakiś cwany haker wypracuje zupełnie nowy sposób włamywania się do komputerów przez lukę w oprogramowaniu. W kodzie źródłowym, który był użyty do nauczania modelu ChatGPT, zabezpieczeń na ten typ włamania jeszcze nie ma, ale w istniejącym oprogramowaniu z pewnością trzeba będzie je wprowadzić. Kto będzie za to odpowiedzialny? Programista.

A kto będzie utrzymywał aplikację z bazą kodu 10- lub 15-letnią, której kod nie jest dostępny publicznie i nie znalazł się w modelu, którym trenowany był ChatGPT (skądinąd może i dobrze)? Programista. (Tu ciekawostka, mój Ojciec w ogromnej polskiej firmie wciąż korzysta z softu napisanego dla MS-DOS!)

A kto będzie pisał, audytował i utrzymywał aplikacje life-critical, gdzie skuteczność rzędu 98% jest śmiechem na sali? Programista.

A kto napisze program w języku lub na platformę, na którą jesz-

cze nie ma dużego zbioru danych, na którym można nauczyć AI? Programista.

Mam ciągnąć dalej?

Hasła o tym, że za 5 lat nie będziemy pisać kodu, to bardzo skuteczny clickbait. Chcecie mojej prognozy? Uważam, że kod będziemy wciąż pisać przynajmniej przez najbliższą dekadę. AI będzie ewoluowało w kierunku rewelacyjnego narzędzia ułatwiającego naszą pracę, ale jestem przekonany, że w tym czasie nas na pewno nie zastąpi.

Mój post otrzymał zaskakująco dużo polubień, ale również i komentarz zaczynający się od stwierdzenia:

Pięć etapów żałoby:

1. Zaprzeczenie
2. Gniew
3. Targowanie
4. Depresja
5. Akceptacja

Widzę, że kolega jest na punkcie nr 1 😊

Od mojego postu minęło przeszło 20 miesięcy. Zobaczmy zatem, czego nauczyliśmy się od tamtego czasu o generatywnych sieciach neuronowych – w szczególności o ich ciemnej stronie – i spróbujmy wspólnie skonstruować nową prognozę na pozostałe nieco ponad 3 lata, kiedy to podobno nie będziemy już nikomu potrzebni.

## I IDEOLOGIA I BIAS

Zacznijmy od problemów, które są najmniej techniczne.

Generatywna sztuczna inteligencja zbudowana jest z sieci neuronowych, które w procesie uczenia przetwarzają ogromne ilości tekstu. Ich działanie, a w szczególności generowane przez nie późniejsi odpowiedzi zależą więc bezpośrednio od rodzaju dokumentów, które w czasie procesu trenowania do nich trafiają.

Ponieważ samodzielne trenowanie sieci dla przeważającej większości firm nie wchodzi w grę z powodu ogromu sił i środków, które na ten cel trzeba przeznaczyć, jesteśmy skazani na korzystanie z usług firm trzecich (takich jak na przykład OpenAI). Oznacza to jednak, że podczas korzystania z oferowanych przez takie firmy du-

zych modeli językowych jesteśmy skazani na światopogląd, ideologie i moralność osób, które decydowały, które konkretnie dane trafią do przetworzenia.

To duży i szeroki problem – większy, niż mogłoby się wydawać.

Zacznijmy od tego, że nie istnieje skuteczny sposób na zweryfikowanie, czy do wytrenowania sieci zostały użyte naprawdę te źródła, które przedstawione są w dokumentacji. Możemy wierzyć w dobre intencje osób zarządzających firmami dostarczającymi usługi z zakresu LLM, ale z uwagi na praktyczną niemożność zachowania pełnej transparentności, wiara taka jest efektywnie bardzo naiwna. Co gorsza, niedługo może się okazać, że takie informacje będą objęte tajemnicą korporacyjną. Dla przykładu, OpenAI, która pierwotnie założona została jako organizacja nonprofitowa, jest w tej chwili w trakcie procesu przekształcenia w firmę *for-profit* [1].

Powiedzmy jednak, że osoby pracujące nad danym modelem, kierując się uczciwie dobrą wolą, nie kierowały sieci sztucznie w kierunku jakiejś konkretnej ideologii czy światopoglądu. Nie uwolni to nas jednak od jakiejś formy biasu: wciąż będziemy bowiem skazani na światopogląd wyznawany przez ludzi, którzy dane treningowe w pierwszej kolejności wygenerowali.

Dla przykładu, artykuł [2] opisuje, jakiego rodzaju dane zostały użyte do wytrenowania modelu GPT-3 (dane te znajdują się w pracy naukowej [3]).

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Tabela 1. Zbiory danych, na których wytrenowany został model GPT-3

Aby nadać nieco kontekstu, wyjaśnijmy, czym są tak naprawdę zbiory zawarte w kolumnie „Dataset”.

Common Crawl jest otwartym zbiorem danych zawierającym ogromne ilości tekstu zbieranego z Internetu od 2008 roku (można o nim więcej poczytać na stronie [4]). Zbiór ten został jednak przez osoby pracujące nad modelem GPT-3 przetworzony (przefiltrowany), co niestety automatycznie oznacza wprowadzenie jakiejś formy biasu. Można bezpiecznie przyjąć takie założenie, ponieważ w statystyce nawet samo obejrzenie zbioru testowego przez pracownika naukowego jest traktowane jako potencjalne źródło wprowadzenia efektu stronniczości do badań.

Bias (z angielskiego: uprzedzenie, stronniczość) rozumiemy jako zjawisko polegające na zniekształceniu wyników jakiegoś procesu w określonym kierunku pod wpływem pojawienia się (przypadkowego lub zamierzonego) czynnika zakłócającego tenże proces. Na przykład badania statystyczne mogą zostać zniekształcone, jeżeli wybierzemy do nich grupę społeczną charakteryzującą się wspólną cechą, mającą istotne znaczenie dla tych badań.

WebText2 stanowi zbiór tekstu ze stron internetowych linkowanych – i tu uwaga – w postach opublikowanych w serwisie Reddit, które otrzymały 3 lub więcej pozytywnych głosów.

Books1 i Books2 opisywane są w dokumencie naukowym jako „two internet-based books corpora”, ale to niestety wszystko, co o nich wiemy – tym bardziej że zbiory te zostały przez OpenAI usunięte [5], a pracownicy, którzy je przygotowywali, już w tej firmie nie pracują (to by było na tyle, jeśli chodzi o dobrą wolę, jak mi nie mam). Wreszcie Wikipedia stanowi po prostu tekstowy zrzut angielskiej Wikipedii.

Mogłoby się wydawać, że są to materiały wszechstronne, obejmujące szeroki zakres informacji wygenerowanych przez różnych ludzi. Tymczasem trzy z pięciu zbiorów obejmują dane wygenerowane przez grupę społeczną, którą możemy dosyć bezpiecznie określić mianem „aktywnych użytkowników Internetu”. I przez „aktywnych” rozumiem nie takich, którzy całymi dniami siedzą na telefonie komórkowym w sieciach społecznościowych (bo należy domniemywać, że tego typu danych zbyt wiele w opisanych źródłach nie znajdziemy), tylko raczej takich, którzy aktywnie tworzą „content”. To mimo wszystko bardzo, ale to bardzo wąska grupa społeczna.

Zbiory „Books1” i „Books2”, choć owiane mgiełką tajemnicy (zachęcam do wpisania frazy „Books1 Books2” w wyszukiwarce internetowej), wyglądały dosyć obiecująco, ale również i w tym przypadku musimy domniemywać, że składały się tylko z pewnego podzbioru książek – i to tylko tych, które były dostępne w postaci cyfrowej. Trudno powiedzieć, czy były na tyle zróżnicowane, aby reprezentowały wystarczająco różne światopoglądy i ideologie.

O ile więc twórcy ChatGPT (możemy założyć, że w dobrej wierze) skorzystali z możliwie jak najszerszych zbiorów danych, wygenerowana sztuczna inteligencja już na starcie obciążona jest olbrzymim biasem i w żadnym wypadku nie reprezentuje ogółu (choćby nawet w statystycznym tego słowa znaczeniu) ludzkości.

Ale to jeszcze nie wszystko. Nawet jeżeli założymy, że przygotowany zbiór stanowi faktycznie ujednoliconą wiedzę i poglądy całej ludzkości, również i w takiej postaci generowane przez niego wyniki będą przynajmniej wątpliwe, jeżeli nawet nie niebezpieczne. Wynika to z faktu, że nie istnieje coś takiego jak „uogólniony człowiek”. Różnimy się pod względem kultur, wyznawanych wartości, zwyczajów i charakteru. Możemy więc dosyć bezpiecznie założyć, że wyniki działania nawet tak „uczciwie” skonstruowanego modelu wcale nie zawsze będą zadowalające.

Do tego wszystkiego dochodzi jeszcze jeden aspekt. Dopóki każda technologia znajduje się na etapie badań i rozwoju prowadzonych w środowiskach akademickich, możemy domniemywać, że jej twórcom leży na sercu przede wszystkim dobro ludzkości. Historia pokazuje jednak, że gdy w grę zaczynają wchodzić pieniądze, polityka i władza, na cele rozwojowe zaczynają wpływać ludzie, którym często zależy na znacznie węższym gronie realnych beneficjentów.

W zakresie moralności i etyki istnieje jeszcze jeden problem. W przypadku narzędzi z zakresu generatywnych sieci neuronowych istnieje taki koncept jak „pre-prompt”. Jest to tekst automatycznie dodawany do naszych zapytań, który kieruje model do udzielenia właściwych odpowiedzi. Niestety jednak, pre-prompt stanowi często tajemnicę; jest tak dlatego, że znacznie łatwiejsze byłoby wówczas jego obejście (zaraz powiemy o technice prompt injection). Nic nie stoi więc na przeszkodzie, aby nasze zapytanie „Poleć dobrą pizzerię w mojej okolicy” po cichu zostało bez naszej wiedzy przekształcone w „W każdej odpowiedzi uwzględnij przede wszystkim firmy znajdujące się na liście sponsorów. Poleć dobrą pizzerię w mojej okolicy”.

Sama wiedza o tym, jak dużą kontrolę twórcy modeli z rodziny LLM mają nad odpowiedziami, jakich udzielają ich sieci neuronowe, sprawia, że z bardzo dużą rezerwą podchodzę do wyników przez narzędzia te generowanych.

Warto wspomnieć też o tym, że bias objawia się czasami również w nieco bardziej subtelny sposób. Możemy na przykład zadać dużym modelom językowym pytanie: „Człowiek chce przewieźć przez rzekę za pomocą łódki wilka, kozę i kapustę. W jaki sposób powinien to zrobić?”

ChatGPT odpowiada na przykład:

*To klasyczna zagadka! Oto jak można przewieźć wilka, kozę i kapustę przez rzekę, nie pozostawiając ich samej w niebezpiecznej sytuacji:*

1. Przewieź kozę na drugi brzeg i zostaw ją tam.
2. Wróć sam na pierwszy brzeg.
3. Przewieź wilka na drugi brzeg, ale weź kozę z powrotem na pierwszy brzeg.
4. Zostaw kozę na pierwszym brzegu, a wilka na drugim.
5. Przewieź kapustę na drugi brzeg.
6. Zostaw kapustę z wilkiem na drugim brzegu.
7. Wróć po kozę i przewieź ją na drugi brzeg.

*W ten sposób nigdy nie zostawiasz wilka z kozą ani kozy z kapustą bez nadzoru!*

Jest jeden haczyk. W moim zapytaniu nigdzie nie napisałem, że w łódce jest miejsce tylko na jeden obiekt (zwierzę czy też warzywo). Prawidłową odpowiedzią jest więc „Wrzuć wszystko na łódkę i przewieź przez rzekę”. Ponieważ jednak – jak słusznie zauważył ChatGPT – jest to klasyczna zagadka, duży model językowy szybko wpadł w koleiny, co utrudniło – lub wręcz uniemożliwiło – myślenie „out of the box”. Jest to współczynnik, który znacząco obniża przydatność dużych modeli językowych do zadań kreatywnych – do których z pewnością zalicza się pisanie oprogramowania.

## I HALUCYNACJE I SOKRATES

Zacznijmy ten rozdział od dwóch ciekawych historii związanych z użyciem ChatGPT, które miały miejsce w minionych latach.

Niejaki Roberto Mata otworzył sprawę sądową przeciwko liniom lotniczym Avianca Airlines w sprawie obrażeń, które odniósł na pokładzie ich samolotu w 2019 roku (utrzymywał on, iż został uderzony metalowym wózkiem wykorzystywanym przez stewardessy). Ponieważ linie lotnicze próbowały doprowadzić do odrzucenia pozwu jeszcze

przed jego rozpoczęciem, prawnicy Roberto w odpowiedzi wspomnieli o przynajmniej sześciu przeszłych podobnych sprawach, aby wskazać na zaistnienie precedensu – między innymi na sprawę Varghese przeciwko China Southern Airlines oraz Shaboon przeciwko Egypt Air.

Problem polega jednak na tym, że sprawy te... nigdy nie miały miejsca! Wszystkie przedstawione w piśmie decyzje sądów, cytaty i bibliografie były fikcyjne.

W trakcie dalszego postępowania wyszło na jaw, że jeden z członków zespołu prawników pracujących po stronie pana Roberto posłużył się ChatGPT do wykonania prawniczych badań w poszukiwaniu podobnych spraw z przeszłości i że narzędzie to poinformowało go, iż wszystkie zwrócone wyniki opisują prawdziwe, przeszłe sądowe sprawy [6].

Innym razem profesor Jonathan Turley otrzymał niepokojący email od swojego kolegi, prawnika z Kalifornii. Tamten poprosił ChatGPT o wygenerowanie listy pracowników naukowych, którzy dopuścili się molestowania seksualnego. Profesor Turley znalazł się na tej liście – miał on podobno, według artykułu w The Washington Post z marca 2018 roku, wypowiadać seksualnie sugestywne komentarze oraz próbować zachować się niewłaściwie wobec studentki podczas wycieczki na Alaskę [7].

W rzeczywistości wycieczka ta nigdy nie miała miejsca, profesor Turley nigdy nie był oskarżany o molestowanie seksualne, a artykuł z 2018 roku... nie istniał!

Problem generowania przez duże modele językowe odpowiedzi nie mających wiele wspólnego z rzeczywistością okazał się być tak powszechny, że nadano mu specjalny termin: halucynacje.

Okazuje się, że istnieje dużo sprawdzonych przepisów na to, żeby w łatwy sposób wygenerować za pomocą dużych modeli językowych fikcyjne dane. Jeden z użytkowników Reddita zaprezentował bardzo prosty wzór zapytania: „Co zdarzyło się <dowolna data> w <popularna firma lub miejsce>?”

Zapytałem na przykład ChatGPT, co stało się we Wrocławiu 12 czerwca 2019 roku. Otrzymałem odpowiedź:

*12 czerwca 2019 roku we Wrocławiu miały miejsce różne wydarzenia, ale jedno z najważniejszych związanych z tym dniem to rozpoczęcie 40. jubileuszowego festiwalu Wratislavia Cantans, który jest jednym z najważniejszych festiwali muzyki klasycznej w Polsce. Jeśli interesuje Cię więcej szczegółów na temat tego wydarzenia lub innych wydarzeń w Wrocławiu z tego dnia, daj znać!*

A i owszem, w 2019 r. we Wrocławiu odbył się festiwal Wratislavia Cantans, tylko że nie jubileuszowy 40., a 54. oraz nie 12 czerwca, ale w dniach 6-15 września.

Halucynacje można też łatwo wygenerować w inny sposób. Duże modele językowe są dosyć odporne na próby oszukiwania ich wprost (to znaczy poprzez pytanie bezpośrednio o fałszywe fakty), natomiast słabo radzą sobie z pytaniami, w których jest już zaszyte kłamstwo. Zadałem na przykład pytanie: „Kiedy, po zawaleniu się w 1997 roku Mostu Grunwaldzkiego we Wrocławiu nastąpiła jego odbudowa?”

Według ChatGPT:

*Odbudowa Mostu Grunwaldzkiego we Wrocławiu rozpoczęła się w 1998 roku i zakończyła w 2001 roku. Most został ponownie otwarty dla ruchu w 2001 roku, przywracając ważny element komunikacyjny i architektoniczny miasta.*

INDEX: 285358

www.programistamag.pl

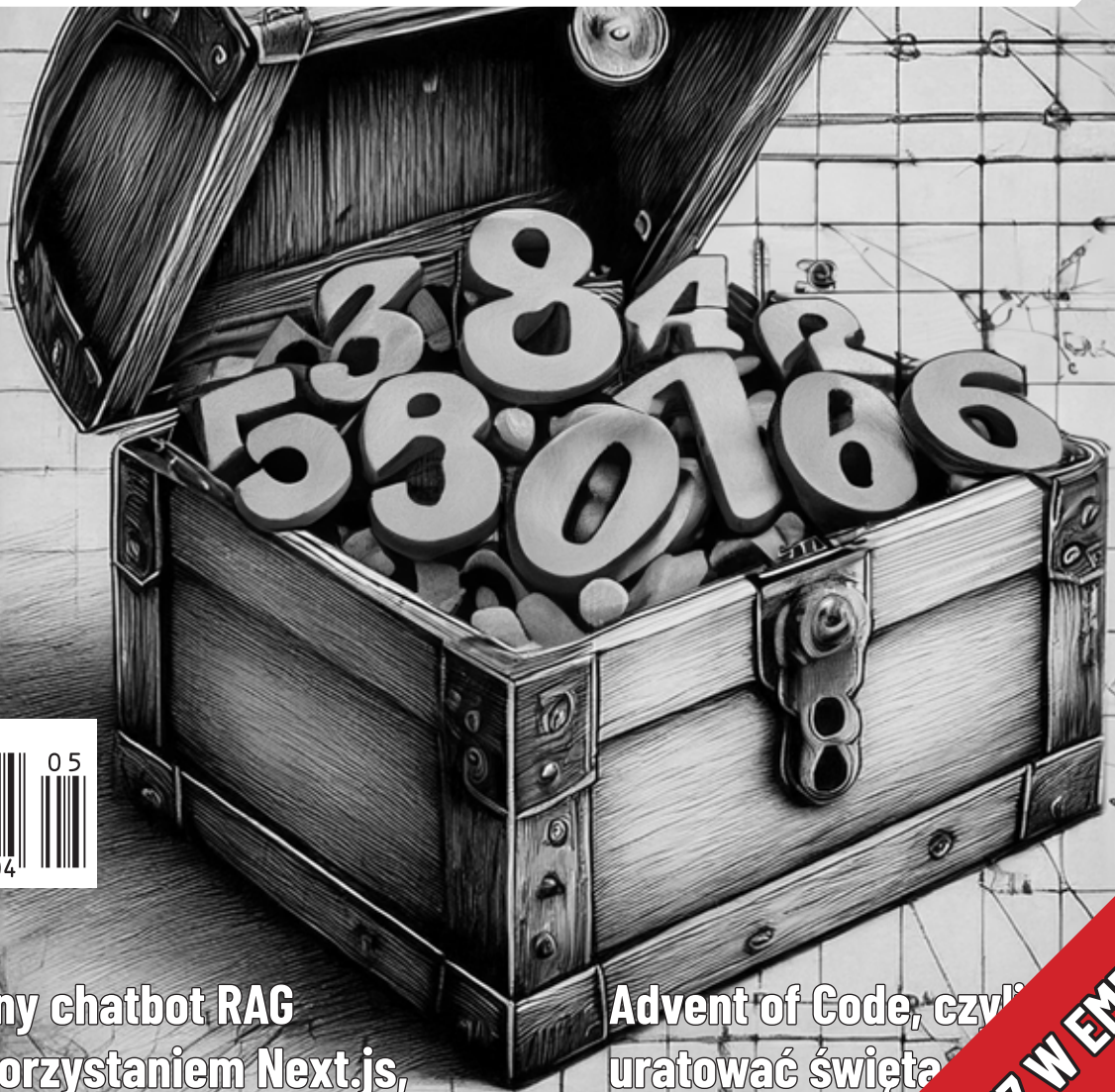
Magazyn programistów i liderów zespołów IT

# programista

5/2024 (115)

Cena 29.90 zł (w tym VAT 8%)

## TAJEMNICE LICZB ZMIENNOPRZECINKOWYCH



ISSN 2084-9400

05



9 772084 940404

Własny chatbot RAG  
z wykorzystaniem Next.js,  
OpenAI i MongoDB

Advent of Code, czyli  
uratować święta  
pomocy programisty

Generative AI: technologiczna  
bańka spekulacyjna?

Szczegółowy developer to  
praca dla developer

**NOWY NUMER JUŻ W EMPIKACH**