Od ELIZY do agentów: krótka historia rozmów z maszyną

Poziom: podstawy **Dotyczy: modele językowe, czaty i agenty AI** **Czas: ~15 min**

Ta strona otwiera książkę o sztucznej inteligencji: od pierwszego programu udającego rozmówcę, przez dekady prób nauczenia maszyny rozumienia tekstu, do dzisiejszych rozmów z modelem i programów, które sięgają po nasze narzędzia same.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
AI	Artificial Intelligence: sztuczna inteligencja, czyli programy naśladujące ludzkie rozumowanie
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
MIT	MIT License: bardzo liberalna licencja wolnego oprogramowania, pozwala na dowolne użycie i modyfikację kodu

Skąd się wzięło

W 1966 roku Joseph Weizenbaum, informatyk z MIT, opublikował program o nazwie ELIZA. Najsłynniejszy skrypt tego programu, DOCTOR, udawał terapeutę nurtu rogeriańskiego: odbijał słowa rozmówcy z powrotem w formie pytania, więc na zdanie „mam problem z szefem" odpowiadał „dlaczego mówisz, że masz problem z szefem?". Pod spodem nie było żadnego rozumienia, tylko dopasowywanie wzorców tekstu i podstawianie fragmentów zdania w gotowy szablon. Weizenbaum był zaskoczony, jak łatwo ludzie zaczynali traktować program poważnie: jego własna sekretarka poprosiła go w pewnym momencie o zostawienie jej samej z ELIZĄ, żeby móc „naprawdę porozmawiać".

Zjawisko przypisywania maszynie zrozumienia, którego ona nie ma, dziś nazywa się efektem ELIZY, a samo pytanie, czy maszyna w ogóle może „rozumieć", zadał kilkanaście lat wcześniej Alan Turing. W artykule z 1950 roku, opublikowanym w czasopiśmie Mind, zaproponował grę imitacyjną: jeżeli człowiek rozmawiający przez tekst nie potrafi odróżnić maszyny od drugiego człowieka, mamy dobry powód, żeby mówić o inteligencji maszyny bez sporu o to, czym inteligencja jest naprawdę. ELIZA, mimo prostoty, oszukiwała ludzi dokładnie w duchu tej gry, choć sam Weizenbaum uważał to raczej za ostrzeżenie niż sukces.

Równolegle rozwijał się zupełnie inny nurt: sieci neuronowe, matematyczne modele luźno inspirowane budową neuronów w mózgu. Pierwszy taki model, perceptron, zbudował w 1958 roku Frank Rosenblatt, ale przez kolejne dekady sieci neuronowe pozostawały ciekawostką akademicką, zbyt małe i zbyt wolno trenowane, żeby przebić prostsze metody. Przełom przyszedł w 2012 roku, gdy sieć AlexNet wygrała konkurs rozpoznawania obrazów ImageNet z przewagą, jakiej nikt się nie spodziewał, pokazując, że duża sieć trenowana na dużych zbiorach danych i mocnych kartach graficznych radzi sobie lepiej niż ręcznie projektowane algorytmy.

Ostatni element układanki dołożyło w 2017 roku ośmioro badaczy z Google w pracy „Attention Is All You Need". Opisali w niej architekturę transformera, która zamiast czytać tekst słowo po słowie po kolei, jak wcześniejsze sieci rekurencyjne, patrzy na wszystkie słowa zdania naraz i uczy się, które z nich są dla siebie ważne. Ta zmiana pozwoliła trenować dużo większe modele dużo szybciej, i to właśnie transformer stoi dziś pod maską każdego dużego modelu językowego.

> **Pojęcie: model językowy.** Program, który na podstawie ogromnej ilości przeczytanego wcześniej tekstu przewiduje, jakie słowo powinno paść jako następne, i w ten sposób, słowo po słowie, buduje całe odpowiedzi. Nie ma wbudowanej bazy faktów ani rozumienia w ludzkim sensie, tylko wyuczone podobieństwo do tekstów, na których się uczył.

Modele językowe rosły z każdym rokiem, ale do listopada 2022 roku zostawały głównie w rękach programistów piszących do nich zapytania przez kod. Wtedy firma OpenAI udostępniła szerokiej publiczności ChatGPT: to samo podejście modelu językowego, tylko owinięte w prostą rozmowę w przeglądarce, bez żadnej wiedzy technicznej po stronie użytkownika.

> **Pojęcie: czat.** Rozmowa z modelem językowym w formie kolejnych pytań i odpowiedzi, w której model tylko odpowiada tekstem, nie wykonuje żadnej akcji poza samą odpowiedzią.

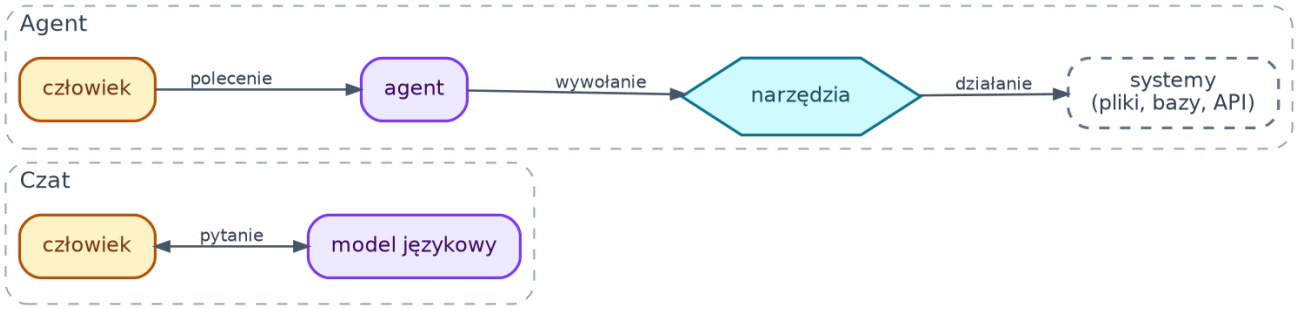
Ta prostota decyzji „zapytaj i dostań tekst" sprawiła, że w ciągu miesięcy z rozmów z modelem korzystały setki milionów ludzi, którzy nigdy wcześniej nie napisali linijki kodu.

Ostatni krok w tej historii to danie modelowi możliwości działania, nie tylko odpowiadania.

> **Pojęcie: agent.** Model językowy, któremu dano dostęp do zestawu narzędzi: funkcji, które potrafią przeczytać plik, wykonać zapytanie do bazy albo wywołać inny system, i który sam decyduje, kiedy i jak z nich skorzystać, żeby zrealizować polecenie.

W listopadzie 2024 roku firma Anthropic opisała otwarty standard Model Context Protocol, ułatwiający podłączanie takich narzędzi do modelu w jednolity sposób, niezależnie od tego, kto model dostarcza. To właśnie ta zmiana, z odpowiadania na działanie, dzieli dzisiejsze „AI" na dwa zupełnie różne narzędzia o tej samej nazwie.

Jak to działa w jednym obrazku



Czat: człowiek pyta model językowy i dostaje odpowiedź tekstem. Agent: człowiek wydaje polecenie, agent wywołuje narzędzia, a te działają na systemach zewnętrznych (kliknięcie otwiera pełny rozmiar)

Różnica sprowadza się do jednej strzałki na tym obrazku: w czacie rozmowa kończy się na odpowiedzi, a w agencie odpowiedź jest dopiero początkiem, bo prowadzi do wywołania narzędzia, które faktycznie coś zmienia w jakimś systemie. Ten sam model językowy potrafi grać obie role, w zależności od tego, czy ktoś dał mu dostęp do narzędzi, czy nie.

Dlaczego Zielona

> **Z ŻYCIA WZIĘTE**
>
> Pani Halina pyta zespół IT, czy „ten czat AI", z którym programiści bawią się na open space, mógłby sam sprawdzić zaległe faktury w Optimie i wysłać przypomnienie do klienta. Zespół tłumaczy, że zwykły czat tylko odpowiada na pytania i nie ma dostępu do żadnego systemu Zielonej. Dopiero agent podłączony do konkretnych narzędzi mógłby coś takiego zrobić, i to tylko w zakresie, na jaki mu wyraźnie pozwolimy.

W Zielonej czat i agent trafiają do zupełnie różnych szuflad. Czat pomaga napisać e-mail do klienta, wyjaśnić komunikat błędu albo zaproponować pierwszą wersję skryptu, i nic poza tym rozmówcą się nie dzieje. Agent z dostępem do narzędzi to już inna kategoria ryzyka, bo potrafi faktycznie coś zrobić w systemach firmy, więc zanim go komukolwiek udostępnimy, musimy wiedzieć dokładnie, do czego ma dostęp i co może zepsuć. Reszta tej książki uczy rozpoznawać tę granicę i świadomie po niej stąpać.

Mapa tej książki

> Zaraz po tej stronie w książce stoi strona autora Różnica pomiędzy czatem a agentem, jedyna dotąd istniejąca strona tej półki: to ona pierwsza rozwija granicę zarysowaną wyżej, zanim reszta rozdziałów zejdzie do szczegółów.

Strona	Na jakie pytanie odpowiada
Mapa pojęć: model, token, kontekst, prompt, temperatura	Co znaczą podstawowe słowa używane w reszcie książki bez tłumaczenia
Czat w praktyce: jak rozmawiać, żeby dostać robotę	Co realnie poprawia jakość odpowiedzi czatu, a co jest tylko przyzwyczajeniem z rozmowy z człowiekiem
Kontekst i pamięć: co model wie, a czego nie	Czym różnią się trzy rodzaje „pamięci" modelu i gdzie każda z nich zawodzi
Narzędzia i MCP: jak agent dotyka świata	Jak wygląda samo wywołanie narzędzia i co standaryzuje protokół MCP
Uprawnienia, prompt injection i zasięg rażenia	Jakie uprawnienia dziedziczy agent i dlaczego treść, którą przeczyta, bywa niebezpieczna
Kiedy agent, kiedy workflow, kiedy zwykły skrypt	Kiedy do zadania lepiej użyć agenta, gotowego workflow, a kiedy zwykłego skryptu
Jak sprawdzić, czy to w ogóle działa	Jak zbudować powtarzalny sposób sprawdzania, czy agent albo prompt robi to, co ma robić
Modele lokalne: AI na własnym sprzęcie	Kiedy model uruchamiamy u siebie zamiast w chmurze, ile pamięci karty graficznej potrzebuje i jak go postawić
RAG w praktyce: czat, który zna naszą bazę wiedzy	Jak podłączyć czat do naszych dokumentów i jak pilnować aktualności, uprawnień i halucynacji
Asystenci kodujący i agenci w terminalu: jak pracować bezpiecznie	Jak zamknąć agenta piszącego kod w piaskownicy, z wąskimi uprawnieniami i przeglądem zmian
Polityka użycia AI w firmie: wzór dla Zielonej	Co wolno wklejać do czatu, jakie narzędzia zatwierdzamy i co mówią RODO oraz AI Act
Projektowanie guardraili: jak ograniczyć, co model i agent mogą zrobić	Jakie warstwy ograniczeń stawiamy wokół modelu i agenta, od filtrów wejścia po zatwierdzanie przez człowieka
Bezpieczeństwo AI w infrastrukturze: modele, agenci i MCP w sieci firmy	Jak odseparować gpu01, pilnować sekretów, serwerów MCP i pochodzenia modeli oraz co robić po incydencie

Powiązane: Różnica pomiędzy czatem a agentem · Mapa pojęć: model, token, kontekst, prompt, temperatura

Źródła: en.wikipedia.org/wiki/ELIZA · historyofinformation.com, „Joseph Weizenbaum Writes ELIZA" · smithsonianmag.com, „Why Joseph Weizenbaum Invented the Eliza Chatbot" ·

plato.stanford.edu/entries/turing-test · A. M. Turing, „Computing Machinery and Intelligence", Mind, 1950 · historyofinformation.com, „Rosenblatt's Perceptron" · dejan.ai, „AlexNet: The Deep Learning Breakthrough" · arxiv.org/abs/1706.03762, „Attention Is All You Need" · en.wikipedia.org/wiki/ChatGPT · anthropic.com/news/model-context-protocol

---

Revision #30
Created 2026-09-28 05:52:42 UTC by Tomasz Konieczny
Updated 2026-10-26 14:10:07 UTC by Tomasz Konieczny