O'REILLY®

Helion

Inżynieria AI

Tworzenie aplikacji z wykorzystaniem modeli bazowych

Chip Huyen

Tytuł oryginału: AI Engineering: Building Applications with Foundation Models

Tłumaczenie: Jacek Janusz

ISBN: 978-83-289-2628-8

Authorized Polish translation of the English edition of *AI Engineering*
ISBN 9781098166304 © 2025 Developer Experience Advisory LLC.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

Wstęp **11**

1. Wprowadzenie do tworzenia aplikacji AI przy użyciu modeli podstawowych **21**

Rozwój inżynierii AI 22
- Od modeli językowych do dużych modeli językowych 22
- Od dużych modeli językowych do modeli podstawowych 28
- Od modeli podstawowych do inżynierii AI 31

Przykłady zastosowań modeli podstawowych 34
- Programowanie 39
- Tworzenie obrazów i wideo 41
- Generowanie tekstów 41
- Edukacja 43
- Boty konwersacyjne 44
- Agregacja informacji 45
- Organizowanie danych 46
- Automatyzacja przepływu danych 46

Planowanie aplikacji AI 47
- Ocena przypadków użycia 47
- Ustalenie oczekiwań 51
- Zaplanowanie kamieni milowych 52
- Konserwacja i utrzymanie 52

Stos technologiczny inżynierii AI 54
- Trzy warstwy stosu AI 55
- Inżynieria AI a inżynieria ML 57
- Inżynieria AI a inżynieria pełnowymiarowa 64

Podsumowanie 65

2. Zrozumienie modeli podstawowych 67
Dane treningowe 68
Modele wielojęzyczne 69
Modele specyficzne dla danej dziedziny 73
Modelowanie 75
Architektura modelu 75
Rozmiar modelu 84
Post-trening 93
Dostrajanie nadzorowane 96
Dostrajanie preferencji 99
Próbkowanie 103
Podstawy próbkowania 103
Strategie próbkowania 105
Efektywność obliczeniowa czasu testowania 110
Dane ustrukturyzowane 113
Probabilistyczna natura sztucznej inteligencji 119
Podsumowanie 125

3. Metodyka ewaluacji 127
Wyzwania związane z ewaluacją modeli podstawowych 128
Zrozumienie wskaźników dotyczących modelowania języka 132
Entropia 133
Entropia krzyżowa 134
Wskaźniki bits-per-character i bits-per-byte 134
Nieokreśloność 135
Interpretacja nieokreśloności i jej zastosowania 136
Ewaluacja dokładna 138
Poprawność funkcjonalna 139
Pomiar poziomu podobieństwa względem danych referencyjnych 140
Wprowadzenie do osadzania 146
AI jako sędzia 148
Dlaczego „AI jako sędzia”? 149
Jak używać metody „AI jako sędzia”? 150
Ograniczenia metody „AI jako sędzia” 153
Jakie modele mogą być sędziami? 156
Ranking modeli wynikający z ewaluacji porównawczej 159
Wyzwania ewaluacji porównawczej 163
Przyszłość ewaluacji porównawczej 166
Podsumowanie 167

4. Ewaluacja modeli AI 168
Kryteria ewaluacji 168
Zdolności specyficzne dla danej dziedziny 170
Zdolności generacyjne 172
Zdolność do podążania za instrukcjami 180
Koszty i opóźnienia 185
Wybór modelu 187
Proces wyboru modelu 187
Budowa czy zakup modelu? 189
Zestawy testów dostępne publicznie 199
Projektowanie procesu ewaluacji 208
Krok 1. Ewaluacja wszystkich komponentów systemu 208
Krok 2. Utworzenie wytycznych do ewaluacji 210
Krok 3. Określenie metod ewaluacji i danych 212
Podsumowanie 216

5. Inżynieria promptów 218
Wprowadzenie do tworzenia promptów 219
Uczenie w kontekście: zero-shot i few-shot 220
Prompt systemowy a prompt użytkownika 222
Długość i efektywność kontekstu 224
Najlepsze zasady inżynierii promptów 226
Twórz jasno i precyzyjnie sformułowane instrukcje 227
Dostarcz niezbędny kontekst 229
Podziel zadania złożone na prostsze podzadania 231
Daj modelowi czas na myślenie 233
Doskonal prompty w procesie iteracyjnym 234
Oceniaj narzędzia do inżynierii promptów 235
Porządkuj prompty i zarządzaj ich wersjami 238
Strategia zabezpieczania promptów 240
Prompty zastrzeżone i inżynieria odwrotna promptów 242
Omijanie zabezpieczeń i wstrzykiwanie promptów 243
Ekstrakcja informacji 248
Obrona przed atakami na prompty 252
Podsumowanie 256

6. Generowanie wspomagane wyszukiwaniem i agenty 257
Generowanie wspomagane wyszukiwaniem 258
Architektura generowania wspomaganego wyszukiwaniem 260
Algorytmy wyszukiwania 261
Optymalizacja wyszukiwania 272
Generowanie wspomagane wyszukiwaniem a inne modalności 277

Agenty 280
Przegląd agentów 281
Narzędzia 283
Planowanie 286
Tryby błędów agenta i sposoby ich oceny 301
Pamięć 304
Podsumowanie 308

7. Dostrajanie 310
Wprowadzenie do dostrajania 311
Kiedy należy dostrajać? 314
Powody, by dostrajać 314
Powody, by nie dostrajać 315
Dostrajanie a generowanie wspomagane wyszukiwaniem (RAG) 319
Ograniczenia pamięciowe 322
Propagacja wsteczna i parametry trenowane 323
Obliczenia dotyczące pamięci 325
Reprezentacje numeryczne 328
Kwantyzacja 330
Techniki dostrajania 334
Dostrajanie efektywne parametrowo 335
Scalanie modeli i dostrajanie wielozadaniowe 348
Taktyki dostrajania 357
Podsumowanie 361

8. Inżynieria zbiorów danych 363
Przygotowanie danych 365
Jakość danych 367
Pokrycie danych 369
Ilość danych 371
Pozyskiwanie i etykietowanie danych 376
Synteza i generowanie sztucznych danych 379
Po co stosować syntezę danych? 379
Tradycyjne metody syntezy danych 381
Synteza danych wspierana przez AI 385
Destylacja modelu 393
Przetwarzanie danych 394
Inspekcja danych 395
Deduplikacja danych 396
Czyszczenie i filtrowanie danych 398
Formatowanie danych 399
Podsumowanie 400

9. Optymalizacja wnioskowania 402
Zrozumienie optymalizacji wnioskowania 403
Podstawy wnioskowania 403
Wskaźniki związane z wydajnością wnioskowania 409
Akceleratory AI 415
Optymalizacja wnioskowania 422
Optymalizacja modelu 422
Optymalizacja usługi wnioskowania 436
Podsumowanie 443

10. Architektura systemów AI i informacje zwrotne od użytkowników 445
Architektura systemów AI 445
Krok 1. Rozszerzenie kontekstu 446
Krok 2. Wprowadzenie zabezpieczeń 447
Krok 3. Wprowadzenie routingu i bramki dostępowej 451
Krok 4. Zmniejszenie opóźnień za pomocą mechanizmów buforowania 456
Krok 5. Dodanie wzorców agentowych 459
Monitorowanie systemu i przejrzystość jego działania 461
Orkiestracja potoku AI 467
Informacja zwrotna od użytkowników 469
Pozyskiwanie informacji zwrotnej z rozmów 469
Projektowanie systemu gromadzenia informacji zwrotnych 475
Ograniczenia systemu gromadzenia informacji zwrotnych 482
Podsumowanie 485

Epilog 487

Skorowidz 488

ROZDZIAŁ 1.

Wprowadzenie do tworzenia aplikacji AI przy użyciu modeli podstawowych

Gdybym mogła użyć tylko jednego słowa do opisania sztucznej inteligencji po 2020 roku, byłaby to *skala*. Modele sztucznej inteligencji stojące za aplikacjami takimi jak ChatGPT, Google Gemini i Midjourney mają taką skalę, że zużywają nietrywialną część (*https://oreil.ly/J0IyO*) światowej energii elektrycznej, a oprócz tego do ich trenowania zaczyna brakować publicznie dostępnych danych internetowych (*https://arxiv.org/abs/2211.04325*).

Skalowanie modeli sztucznej inteligencji ma dwie główne konsekwencje. Po pierwsze, modele AI stają się coraz potężniejsze i zdolne do wykonywania większej liczby zadań, a dzięki temu mają więcej zastosowań. Coraz więcej osób i zespołów wykorzystuje sztuczną inteligencję do zwiększania produktywności, tworzenia wartości ekonomicznej i poprawy jakości życia.

Po drugie, trenowanie dużych modeli językowych (ang. *large language model*, w skrócie LLM) wymaga danych, zasobów obliczeniowych i specjalistycznego talentu, na które może sobie pozwolić tylko kilka organizacji. Doprowadziło to do pojawienia się *modelu jako usługi*. Oznacza to, że modele opracowane przez niewielką liczbę organizacji są udostępniane innym do wykorzystania jako usługa. Każdy, kto chce wykorzystać sztuczną inteligencję do tworzenia aplikacji, może od razu użyć tych modeli i nie jest zmuszony do inwestowania z góry w ich budowę.

Krótko mówiąc, popyt na aplikacje AI wzrósł, podczas gdy poziom wymagań potrzebnych do tworzenia aplikacji AI zmniejszył się. To sprawiło, że *inżynieria AI* — proces tworzenia aplikacji na podstawie łatwo dostępnych modeli — stała się jedną z najszybciej rozwijających się dyscyplin inżynieryjnych.

Tworzenie aplikacji w oparciu o modele uczenia maszynowego (ML) nie jest niczym nowym. Już długo przed tym, jak modele uczenia maszynowego stały się popularne, sztuczna inteligencja zasilała wiele aplikacji, w tym rekomendacje produktów, wykrywanie oszustw i przewidywanie rezygnacji. Co prawda wiele zasad tworzenia aplikacji AI pozostaje takich samych, jednak nowa generacja łatwo dostępnych, wielkoskalowych modeli udostępnia nowe możliwości i nowe wyzwania, na których koncentruje się ta książka.

Rozdział ten rozpoczyna się od przeglądu modeli podstawowych, kluczowego katalizatora eksplozji inżynierii AI. Następnie zostanie przeanalizowany szereg udanych wdrożeń sztucznej inteligencji, z których każdy ilustruje, w czym sztuczna inteligencja jest dobra, a w czym jeszcze nie.

W miarę jak możliwości sztucznej inteligencji rosną z dnia na dzień, przewidywanie jej przyszłych zdolności staje się coraz większym wyzwaniem. Istniejące wzorce zastosowań mogą jednak pomóc odkryć obecne możliwości i zaoferować wskazówki dotyczące tego, w jaki sposób sztuczna inteligencja może być nadal wykorzystywana w przyszłości.

Na zakończenie rozdziału zostanie zaprezentowany nowy stos sztucznej inteligencji. Wyjaśnię, co się zmieniło w modelach podstawowych, co pozostaje takie samo i czym dzisiejsza rola inżyniera sztucznej inteligencji różni się od roli inżyniera tradycyjnego uczenia maszynowego[1].

Rozwój inżynierii AI

Podstawowe modele wyłoniły się z dużych modeli językowych, które z kolei powstały jako modele językowe. Może się wydawać, że aplikacje takie jak ChatGPT i Copilot GitHuba pojawiły się znikąd, jednak są one wynikiem działania dziesięcioleci postępu technologicznego — pierwsze modele językowe pojawiły się w latach 50. XX wieku. W tym podrozdziale przedstawiono kluczowe przełomy, które umożliwiły ewolucję od modeli językowych do inżynierii sztucznej inteligencji.

Od modeli językowych do dużych modeli językowych

Chociaż modele językowe istnieją już od jakiegoś czasu, to dopiero dzięki *samonadzorowaniu* mogły się one rozwinąć do takiej skali jak obecnie. Ten punkt zawiera krótką analizę objaśniającą, czym są modele językowe i samonadzorowanie. Jeśli jesteś już zaznajomiony z tymi pojęciami, możesz go pominąć.

Modele językowe

Model językowy koduje informacje statystyczne na temat jednego języka lub większej ich liczby. Intuicyjnie rzecz ujmując, informacje te mówią nam, jak prawdopodobne jest pojawienie się słowa w danym kontekście. Na przykład biorąc pod uwagę kontekst „Mój ulubiony kolor to __", model językowy kodujący język polski powinien częściej przewidywać słowo „niebieski" niż „samochód".

Statystyczna natura języków została odkryta wieki temu. W opowiadaniu z 1905 roku *Tańczące sylwetki* (*https://pl.wikipedia.org/wiki/Ta%C5%84cz%C4%85ce_sylwetki*) Sherlock Holmes wykorzystał proste informacje statystyczne o języku angielskim do rozszyfrowania sekwencji tajemniczych figur. Ponieważ najczęściej występującą literą w języku angielskim jest *E*, Holmes wydedukował, że najczęściej występujący rysunek musi również oznaczać *E*.

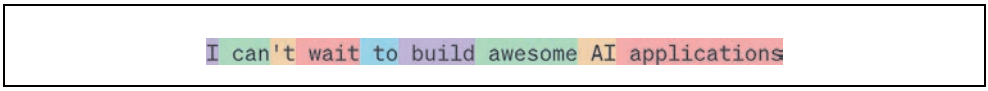
Później, podczas II wojny światowej, Claude Shannon wykorzystał bardziej wyrafinowane statystyki do rozszyfrowania wiadomości wroga. Jego praca dotycząca modelowania języka angielskiego została opublikowana w 1951 roku w przełomowym artykule *Prediction and Entropy*

[1] W tej książce używam sformułowania „tradycyjne uczenie maszynowe" w odniesieniu do wszystkich modeli ML, które istniały przed pojawieniem się modeli podstawowych.

of Printed English (*https://oreil.ly/G_HBp*). Wiele pojęć wprowadzonych w tym artykule, w tym entropia, jest nadal wykorzystywanych do modelowania języka.

Na początku model językowy obejmował tylko jeden język. Obecnie jednak model językowy może zawierać wiele języków.

Podstawową jednostką modelu językowego jest *token*. W zależności od modelu token może być znakiem, słowem lub częścią słowa (na przykład „-izm”)[2]. Przykładowo GPT-4, model stojący za ChatGPT, dzieli angielskie zdanie „I can't wait to build AI applications” na dziewięć tokenów, jak pokazano na rysunku 1.1. Należy zauważyć, że w tym przykładzie słowo „can't” zostało podzielone na dwa tokeny: *can* i *'t*. Na stronie OpenAI możesz sprawdzić, w jaki sposób różne modele OpenAI tokenizują tekst (*https://oreil.ly/0QI91*).



Rysunek 1.1. Przykład tokenizacji zdania wykonanej przez GPT-4

Proces dzielenia oryginalnego tekstu na tokeny jest nazywany *tokenizacją*. W przypadku GPT-4 przeciętny token to około ¾ długości słowa (*https://oreil.ly/EYccr*). Tak więc 100 tokenów to około 75 słów.

Zbiór wszystkich tokenów, z którymi model może pracować, jest zwany *słownikiem*. Do skonstruowania dużej liczby oddzielnych słów można użyć niewielkiej liczby tokenów. Jest to podobne do użycia kilku liter alfabetu w celu utworzenia wielu słów. Model Mixtral 8x7B (*https://oreil.ly/bxMcW*) ma słownik o rozmiarze 32 000. Rozmiar słownika GPT-4 wynosi 100 256 (*https://github.com/openai/tiktoken/blob/main/tiktoken/model.py*). Metoda tokenizacji i rozmiar słownika są ustalane przez twórców modelu.

Dlaczego modele językowe używają jako jednostki *tokena* zamiast *słowa* lub *znaku*? Istnieją trzy główne powody:

1. W przeciwieństwie do znaków tokeny pozwalają modelowi rozbijać słowa na znaczące komponenty. Na przykład słowo „cooking” (gotowanie) można podzielić na „cook” i „ing”, przy czym oba składniki wpływają na znaczenie oryginalnego słowa.
2. Ponieważ istnieje mniej unikatowych tokenów niż unikatowych słów, zmniejsza to rozmiar słownika modelu i czyni go bardziej wydajnym (jak omówiono w rozdziale 2.).
3. Tokeny również pomagają modelowi przetwarzać słowa nieznane. Na przykład wymyślone słowo, takie jak „chatgpting”, można podzielić na „chatgpt” i „ing”, a przez to pomóc modelowi zrozumieć jego strukturę. Tokeny są krótsze od słów, ale zawierają więcej informacji niż pojedyncze znaki.

[2] W przypadku języków innych niż angielski pojedynczy znak Unicode może być czasami reprezentowany jako wiele tokenów.

Istnieją dwa główne typy modeli językowych: *maskowane modele językowe* i *autoregresyjne modele językowe*. Różnią się one w zależności od tego, jakich informacji mogą użyć do wstępnego prognozowania tokena:

Maskowany model językowy

Maskowany model językowy jest trenowany w celu prognozowania brakujących tokenów w dowolnym miejscu sekwencji *przy użyciu kontekstu zarówno przed brakującymi tokenami, jak i po nich*. Ten model językowy jest przede wszystkim trenowany po to, by móc wypełnić puste miejsca. Na przykład biorąc pod uwagę kontekst „Mój ulubiony __ to niebieski", maskowany model językowy powinien przewidzieć, że puste miejsce to prawdopodobnie słowo „kolor". Dobrze znany przykład maskowanego modelu językowego to dwukierunkowe reprezentacje kodera z modeli transformer lub BERT (Devlin i in., 2018 (*https://arxiv.org/abs/1810.04805*)).

W chwili obecnej maskowane modele językowe są powszechnie stosowane w zadaniach niegeneratywnych, takich jak analiza nastrojów i klasyfikacja tekstów. Są one również przydatne w zadaniach wymagających zrozumienia ogólnego kontekstu, takich jak debugowanie kodu, w przypadku których model musi zrozumieć zarówno poprzedni, jak i następny kod, aby zidentyfikować błędy.

Autoregresyjny model językowy

Autoregresyjny model językowy jest trenowany w celu prognozowania następnego tokena w sekwencji, a *używa tylko wcześniejszych tokenów*. Przewiduje on, jaki wyraz pojawi się na końcu zdania „Mój ulubiony kolor to __"[3]. Model autoregresyjny może stale generować jeden token po drugim. Obecnie istniejące autoregresyjne modele językowe są modelami wybieranymi do generowania tekstu i z tego powodu są znacznie bardziej popularne niż maskowane modele językowe[4].

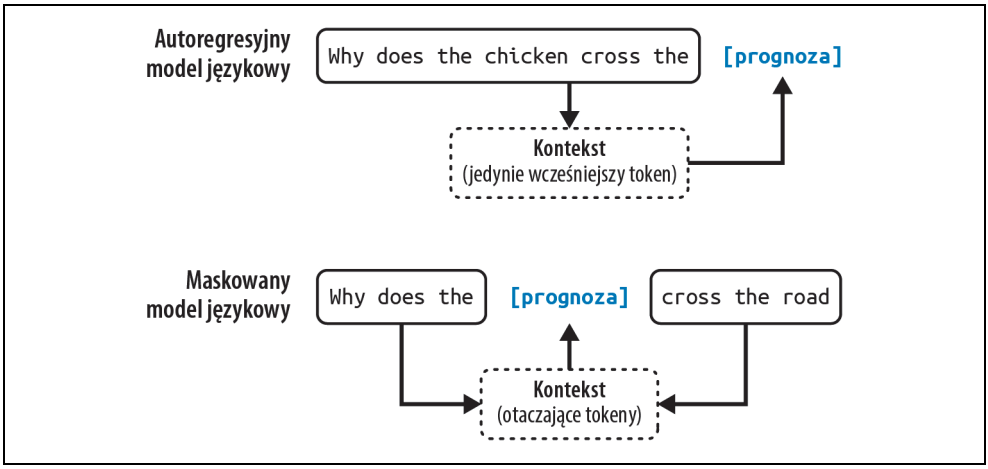
Na rysunku 1.2 pokazano dwa powyżej wspominane modele językowe.

W tej książce, o ile nie zostanie to wyraźnie określone, fraza „model językowy" będzie oznaczała model autoregresyjny.

Wyniki uzyskane z modeli językowych mają charakter otwarty. Model językowy może wykorzystywać swoje stałe, skończone słownictwo do konstruowania nieskończonej liczby możliwych wyników. Model, który generuje wyniki otwarte, jest nazywany *generatywnym*, stąd termin *generatywna sztuczna inteligencja*.

[3] Autoregresyjne modele językowe są czasami nazywane przyczynowymi modelami językowymi (*https://oreil.ly/h0Y8x*).

[4] Technicznie rzecz ujmując, jeśli bardzo się postarasz, maskowany model językowy, taki jak BERT, może być używany również do generowania tekstu.



Rysunek 1.2. Autoregresyjny model językowy i maskowany model językowy

Model językowy możesz sobie wyobrazić jako maszynę do uzupełniania — biorąc pod uwagę tekst (prompt), próbuje ona go uzupełnić. Oto przykład:

Prompt (od użytkownika): `„Być albo nie być"`

Uzupełnienie (z modelu językowego): `„ – oto jest pytanie".`

Ważne jest, by pamiętać, że uzupełnienia są prognozami opartymi na prawdopodobieństwie i nie gwarantują poprawności. Ta probabilistyczna natura modeli językowych sprawia, że są one zarówno ekscytujące, jak i frustrujące w użyciu. Omówimy to dokładniej w rozdziale 2.

Choć brzmi prosto, uzupełnianie jest niezwykle potężne. Wiele zadań, w tym tłumaczenia, streszczanie, kodowanie i rozwiązywanie problemów matematycznych, można ująć w ramy zadań uzupełniania. Na przykład biorąc pod uwagę prompt „»Jak się masz« po francusku to…", model językowy może być w stanie uzupełnić je o „*Comment ça va*", skutecznie tłumacząc z jednego języka na drugi.

Innym przykładem może być następujący prompt:

Pytanie: `Czy ta wiadomość mailowa jest być może spamem? Oto ona: <treść e-maila>`

Odpowiedź:

Model językowy może być w stanie uzupełnić je o „Prawdopodobny spam", co zmienia ten model językowy w klasyfikator spamu.

Jak stwierdziliśmy, uzupełnianie jest potężnym narzędziem, jednak nie jest tym samym, co angażowanie się w konwersację. Na przykład jeśli zadasz maszynie uzupełniającej pytanie, może ona uzupełnić to, co napisałeś, dodając kolejne pytanie, zamiast odpowiadać na nie. W podrozdziale „Post-trening" zostanie wyjaśnione, jak sprawić, by model odpowiednio reagował na żądanie użytkownika.

Samonadzorowanie

Modelowanie języka to tylko jeden z wielu algorytmów ML. Istnieją również modele służące do wykrywania obiektów, modelowania tematów, systemów rekomendacji, prognozowania pogody, przewidywania cen akcji itp. Co jest takiego szczególnego w modelach językowych, że stały się one najważniejsze, co z kolei sprawiło, iż powstał ChatGPT?

Odpowiedź brzmi: modele językowe mogą być trenowane przy użyciu *samonadzorowania*, podczas gdy wiele innych modeli wymaga *nadzorowania*. Nadzorowanie oznacza proces trenowania algorytmów uczenia maszynowego przy użyciu etykietowanych danych, których uzyskanie może być kosztowne i powolne. Samonadzorowanie pomaga przezwyciężyć wąskie gardło etykietowania danych poprzez tworzenie większych zbiorów danych, na podstawie których modele mogą się uczyć i z powodzeniem się rozwijać. Oto jak należy to zrobić.

Nadzorowanie polega na etykietowaniu przykładów w celu zaprezentowania zachowań, których model ma się nauczyć, a następnie trenowaniu na nich. Po wytrenowaniu można użyć modelu z nowymi danymi. Na przykład aby wytrenować model wykrywania oszustw, należy użyć przykładów transakcji, z których każda jest oznaczona jako „oszustwo" lub „brak oszustwa". Gdy model nauczy się na tych przykładach, można go użyć do przewidywania, czy transakcja jest nieuczciwa.

Sukces modeli sztucznej inteligencji w 2010 roku polegał na nadzorowaniu. Model, który zapoczątkował rewolucję w uczeniu głębokim, czyli AlexNet (Krizhevsky i in., 2012 (*https://oreil.ly/WEQFj*)), był nadzorowany. Został wytrenowany, aby się nauczyć klasyfikować ponad 1 milion obrazów w zbiorze danych ImageNet. Przypisywał każdy obraz do jednej z 1000 kategorii, takich jak „samochód", „balon" lub „małpa".

Wada nadzorowania to to, że etykietowanie danych jest kosztowne i czasochłonne. Jeśli oznaczenie jednego obrazu przez jedną osobę kosztuje 5 centów, oznaczenie miliona obrazów dla ImageNet kosztowałoby 50 tysięcy dolarów[5]. Jeśli chciałbyś, aby dwie różne osoby etykietowały każdy z obrazów (by można było sprawdzić jakość etykiet), kosztowałoby to dwa razy więcej. Ponieważ świat zawiera znacznie więcej niż 1000 typów obiektów, to aby rozszerzyć możliwości modeli i umożliwić ich obsługę, należy zaetykietować większą liczbę kategorii. W celu uzyskania 1 miliona kategorii sam koszt etykietowania będzie musiał wzrosnąć do 50 milionów dolarów.

Etykietowanie przedmiotów codziennego użytku jest czymś, co większość ludzi może zrobić bez wcześniejszego trenowania. Można to więc wykonać stosunkowo tanio. Jednak nie wszystkie zadania etykietowania są tak proste. Generowanie tłumaczeń łacińskich dla modelu angielsko-łacińskiego jest już droższe. Etykietowanie, czy skan tomografii komputerowej wykazuje oznaki raka, byłoby astronomicznie kosztowne.

[5] Rzeczywisty koszt etykietowania danych różni się w zależności od kilku czynników, w tym złożoności zadania, skali (większe zbiory danych zazwyczaj skutkują niższymi kosztami za próbkę) oraz dostawcy usług etykietowania. Na przykład od września 2024 r. Amazon SageMaker Ground Truth (*https://oreil.ly/EVXJl*) pobiera centów za obraz w przypadku etykietowania mniej niż 50 tysięcy obrazów, ale tylko 2 centy za obraz, gdy etykietowanych jest ponad 1 milion obrazów.

Samonadzorowanie pomaga przezwyciężyć wąskie gardło etykietowania danych. W takim przypadku model nie wymaga jawnych etykiet, ale może sam je tworzyć na podstawie danych wejściowych. Modelowanie języka jest operacją samonadzorującą, ponieważ każda sekwencja wejściowa dostarcza zarówno etykiet (prognozowanych tokenów), jak i kontekstów, które model może wykorzystać do przewidywania tych etykiet. Na przykład zdanie „I love street food" zwraca sześć próbek treningowych, jak pokazano w tabeli 1.1.

Tabela 1.1. Próbki treningowe ze zdania „I love street food" wykorzystywane do modelowania języka

Dane wejściowe (kontekst)	Dane wyjściowe (następny token)
`<BOS>`	`I`
`<BOS>, I`	`love`
`<BOS>, I, love`	`street`
`<BOS>, I, love, street`	`food`
`<BOS>, I, love, street, food`	`.`
`<BOS>, I, love, street, food, .`	`<EOS>`

W tabeli 1.1 znaczniki `<BOS>` i `<EOS>` oznaczają początek i koniec sekwencji. Znaczniki te są niezbędne, aby model języka mógł pracować z wieloma sekwencjami. Każdy znacznik jest zwykle traktowany przez model jako jeden specjalny token. Znacznik końca sekwencji jest szczególnie ważny, ponieważ pozwala modelom językowym rozpoznać, kiedy należy zakończyć generowaną odpowiedź[6].

Samonadzorowanie różni się od nienadzorowania. W samonadzorowaniu etykiety są wnioskowane na podstawie danych wejściowych. W uczeniu bez nadzoru etykiety w ogóle nie są potrzebne.

Samonadzorowanie oznacza, że modele językowe mogą się uczyć na podstawie sekwencji tekstowych bez konieczności ich etykietowania. Ponieważ sekwencje tekstowe są wszędzie — w książkach, postach na blogach, artykułach i komentarzach na Reddicie — możliwe jest skonstruowanie ogromnej ilości danych treningowych, co pozwala modelom językowym rozwijać się do wersji LLM.

LLM (duży model językowy) nie jest jednak terminem naukowym. Jak duży musi być model językowy, aby można go było uznać za *duży*? To, co dziś jest duże, jutro może być uważane za małe. Rozmiar modelu jest zazwyczaj mierzony liczbą jego parametrów. *Parametr* to zmienna w modelu ML, która jest aktualizowana w procesie trenowania[7]. Ogólnie rzecz ujmując (choć nie zawsze jest to prawda), im więcej parametrów ma model, tym większa jest jego zdolność do uczenia się pożądanych zachowań.

[6] To trochę jak z ludźmi — ważne jest, aby wiedzieli, kiedy powinni przestać mówić.

[7] Na studiach uczono mnie, że parametry modelu obejmują zarówno wagi modelu, jak i jego odchylenia. Obecnie jednak zazwyczaj używamy wag modeli w odniesieniu do wszystkich parametrów.

Gdy pierwszy generatywny, wstępnie wytrenowany model typu transformer (GPT) firmy OpenAI pojawił się w czerwcu 2018 roku, miał 117 milionów parametrów i był uważany za duży. W lutym 2019 roku, gdy OpenAI wprowadziło GPT-2 z 1,5 miliarda parametrów, 117 milionów zostało zdegradowanych do uznania za małe. W chwili pisania tej książki model o 100 miliardach parametrów jest uważany za duży. Być może pewnego dnia rozmiar ten zostanie uznany za mały.

Zanim przejdziemy do następnego punktu, chciałabym poruszyć kwestię, która jest zwykle uważana za oczywistą: *dlaczego większe modele potrzebują większej ilości danych?* Większe modele mają większą zdolność uczenia się, a zatem potrzebują więcej danych treningowych, aby zmaksymalizować swoją wydajność[8]. Można również trenować duży model na małym zbiorze danych, ale byłoby to marnotrawstwo obliczeń. Podobne lub lepsze wyniki da się osiągnąć na tym samym zbiorze danych przy użyciu mniejszych modeli.

Od dużych modeli językowych do modeli podstawowych

Chociaż modele językowe są zdolne do wykonywania niesamowitych zadań, są one ograniczone do tekstu. My, ludzie, postrzegamy świat nie tylko poprzez język, ale także przez wzrok, słuch, dotyk i wiele innych sposobów. Możliwość przetwarzania danych wykraczających poza tekst jest niezbędna, aby sztuczna inteligencja mogła działać w świecie rzeczywistym.

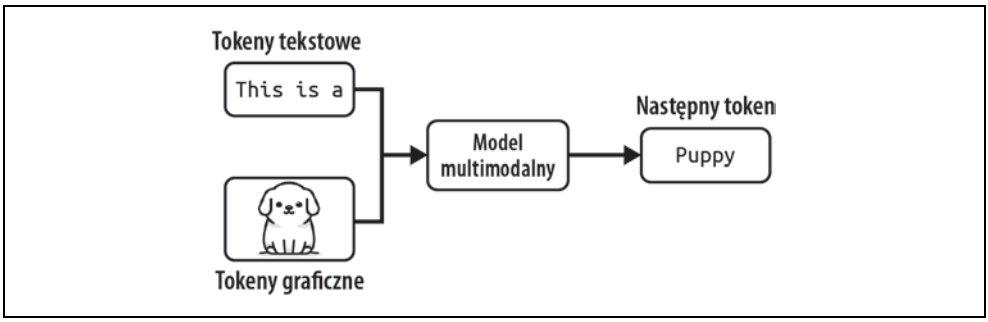
Z tego powodu modele językowe są rozszerzane, aby uwzględnić więcej rodzajów danych. GPT-4V i Claude 3 mogą rozumieć obrazy i teksty. Niektóre modele rozumieją nawet filmy, zasoby 3D, struktury białek itp. Włączenie większej liczby rodzajów danych do modeli językowych czyni je jeszcze potężniejszymi. Firma OpenAI zanotowała w 2023 roku w karcie systemu GPT-4V (*https://oreil.ly/NoGX7*), że „dodanie dodatkowych opcji (takich jak dane wejściowe w postaci obrazów) do LLM jest postrzegane przez niektórych jako kluczowa granica w badaniach i rozwoju sztucznej inteligencji".

Mimo że wiele osób nadal nazywa Gemini i GPT-4V modelami typu LLM, bardziej pasują one do *modeli podstawowych* (*https://arxiv.org/abs/2108.07258*). Słowo „podstawowy" obejmuje zarówno znaczenie tych modeli w zastosowaniach sztucznej inteligencji, jak i fakt, że można je budować w celu rozwiązywania różnych zadań.

Modele podstawowe stanowią przełom w stosunku do tradycyjnej struktury badań nad sztuczną inteligencją. Przez długi czas badania nad sztuczną inteligencją były podzielone według typów danych. Przetwarzanie języka naturalnego (NLP) zajmuje się tylko tekstem. Widzenie komputerowe zajmuje się tylko widzeniem. Modele tekstowe mogą być wykorzystywane do zadań takich jak tłumaczenie i wykrywanie spamu. Modele oparte wyłącznie na obrazach mogą być wykorzystywane do wykrywania obiektów i klasyfikacji obrazów. Modele dźwiękowe mogą obsługiwać rozpoznawanie mowy (zamiana mowy na tekst) i syntezę mowy (zamiana tekstu na mowę).

[8] To wydaje się sprzeczne z intuicją, że większe modele wymagają większej ilości danych treningowych. Skoro model jest bardziej wydajny, czyż nie powinien wymagać mniejszej liczby przykładów, by się skutecznie uczyć? Jednak nie chcemy, aby duży model dorównywał wydajności małego modelu przy użyciu tych samych danych. Staramy się zmaksymalizować wydajność modelu.

Model, który może pracować z więcej niż jednym typem danych, jest nazywany *modelem multimodalnym*. Generatywny model multimodalny jest również nazywany dużym modelem multimodalnym (ang. *large multimodal model*, w skrócie LMM). Jeśli model językowy generuje następny token uwarunkowany tylko tokenami tekstowymi, model multimodalny generuje następny token uwarunkowany zarówno tokenami tekstowymi, jak i graficznymi lub dowolnymi innymi, które są przez niego obsługiwane, co pokazano na rysunku 1.3.



Rysunek 1.3. Model multimodalny może wygenerować następny token przy użyciu informacji z tokenów tekstowych i graficznych

Podobnie jak modele językowe, również modele multimodalne potrzebują danych w celu rozwijania się. Tu też można zastosować samonadzorowanie. Na przykład firma OpenAI wykorzystała wariant samonadzorowania, zwany *nadzorem języka naturalnego*, do trenowania swojego modelu językowo-obrazowego CLIP (OpenAI, 2021 (*https://oreil.ly/zcqdu*)). Zamiast ręcznie generować etykiety dla każdego obrazu, znaleziono pary (obraz, tekst), które współwystępowały w internecie. Można więc było wygenerować (bez kosztów ręcznego etykietowania) zbiór danych zawierający 400 milionów par (obraz, tekst), który był 400 razy większy niż ImageNet. Ten zbiór danych pozwolił CLIP stać się pierwszym modelem, który mógł wykonywać wiele różnych zadań związanych z klasyfikowaniem obrazów bez konieczności dodatkowego trenowania.

W niniejszej książce używamy terminu „modele podstawowe” w odniesieniu zarówno do dużych modeli językowych, jak i dużych modeli multimodalnych.

Należy zauważyć, że CLIP nie jest modelem generatywnym — nie został wyszkolony do generowania wyników otwartych. CLIP jest *modelem osadzania*, wytrenowanym do tworzenia wspólnych osadzeń zarówno tekstów, jak i obrazów. W punkcie „Wprowadzenie do osadzania”, dostępnym w dalszej części książki, dokładniej wyjaśniono, czym jest osadzanie. Na razie możesz traktować osadzenia jako wektory, które mają na celu uchwycenie znaczenia oryginalnych danych. Modele osadzania multimodalnego, takie jak CLIP, są podstawą generatywnych modeli multimodalnych, na przykład Flamingo, LLaVA i Gemini (a wcześniej Bard).

Modele podstawowe oznaczają również przejście od modeli o specyficznych zastosowaniach do modeli ogólnego przeznaczenia. Wcześniej modele były często opracowywane dla konkretnych zadań, takich jak analiza nastrojów lub tłumaczenie. Model wytrenowany do analizy nastrojów nie byłby w stanie wykonać tłumaczenia (i odwrotnie).

Modele podstawowe dzięki swojej skali i sposobowi, w jaki są trenowane, mogą wykonywać szeroki zakres zadań. Modele ogólnego przeznaczenia mogą od razu działać stosunkowo dobrze podczas rozwiązywania wielu różnych problemów. LLM wykonuje zarówno analizę opinii, jak i tłumaczenie. Często jednak można odpowiednio dostroić model ogólnego przeznaczenia, aby zmaksymalizować jego wydajność podczas wykonywania określonego zadania.

Na rysunku 1.4 przedstawiono zadania wykorzystywane przez test Super-NaturalInstructions do oceny modeli podstawowych (Wang i in., 2022 (*https://arxiv.org/abs/2204.07705*)), by dać wyobrażenie o typach zadań, które mogą one realizować.

Rysunek 1.4. Zakres zadań w teście Super-NaturalInstructions (Wang i in., 2022)

Wyobraź sobie, że pracujesz ze sprzedawcą detalicznym nad stworzeniem aplikacji do generowania opisów produktów na jego stronie internetowej. Gotowy model będzie w stanie wygenerować dokładne opisy, ale może nie uchwycić głosu marki lub nie podkreślić jej przekazu. Wygenerowane opisy mogą nawet być pełne marketingowej mowy i frazesów.

Istnieje wiele metod, których możesz użyć, aby model wygenerował to, co chcesz. Można na przykład opracować szczegółowe instrukcje z przykładami pożądanych opisów produktów. Takie podejście to *inżynieria promptów*. Łączysz model z bazą danych o opiniach klientów, którą model wykorzystuje do generowania lepszych opisów. Korzystanie z bazy danych w celu uzupełnienia instrukcji nazywa się *generowaniem wspomaganym wyszukiwaniem* (RAG). Można również *dostroić* (czyli dodatkowo wytrenować) model na zbiorze danych z wysokiej jakości opisami produktów.

Inżynieria promptów, RAG i dostrajanie to trzy bardzo powszechne techniki inżynierii sztucznej inteligencji, których można użyć do dostosowania modelu do własnych potrzeb. Zostaną one szczegółowo omówione w dalszej części książki.

Dostosowanie istniejącego, potężnego modelu do danego zadania jest na ogół znacznie łatwiejsze niż zbudowanie dedykowanego modelu od podstaw — mówimy o 10 przykładach i 1 weekendzie w porównaniu z 1 milionem przykładów i 6 miesiącami. Modele podstawowe sprawiają, że tworzenie aplikacji AI jest tańsze, a czas wdrażania produktów się skraca. Dokładna ilość danych potrzebnych do dostosowania modelu zależy od używanej techniki. W tej książce poruszymy również tę kwestię podczas omawiania każdej z metod. Niemniej jednak modele specyficzne dla danego zadania nadal mają wiele zalet, na przykład mogą być znacznie mniejsze, a dzięki temu szybsze i tańsze w użyciu.

Czy warto stworzyć własny model, czy skorzystać z istniejącego, to klasyczne pytanie typu „kupić czy zbudować", na które zespoły muszą samodzielnie odpowiedzieć. Analizy zawarte w tej książce mogą pomóc w podjęciu decyzji.

Od modeli podstawowych do inżynierii AI

Inżynieria AI dotyczy procesu tworzenia aplikacji przy użyciu modeli podstawowych. Aplikacje AI są budowane już od ponad dekady — proces ten jest często znany jako inżynieria ML lub MLOps (skrót od angielskich słów *ML operations* — działania związane z uczeniem maszynowym). Dlaczego jednak obecnie koncentrujemy się na inżynierii AI?

Tradycyjna inżynieria ML polega na opracowywaniu modeli uczenia maszynowego, natomiast inżynieria AI wykorzystuje już istniejące modele. Istnienie potężnych modeli podstawowych i łatwy dostęp do nich prowadzi do trzech czynników, które razem tworzą idealne warunki do szybkiego rozwoju inżynierii AI jako dyscypliny:

Czynnik 1. Możliwości AI ogólnego przeznaczenia

> Modele podstawowe są potężne nie tylko dlatego, że mogą lepiej wykonywać istniejące zadania. Są również potężne, ponieważ mogą wykonywać ich więcej. Aplikacje, które wcześniej uważano za niemożliwe do zaimplementowania, są teraz dostępne, a te, o których wcześniej nie myślano, zaczynają być wdrażane. Nawet to, co dziś wydaje się niemożliwe, jutro może stać się rzeczywistością. Sprawia to, że AI znajduje zastosowanie w coraz większej liczbie dziedzin, co prowadzi do wzrostu zarówno liczby użytkowników, jak i zapotrzebowania na aplikacje sztucznej inteligencji.

Ponieważ sztuczna inteligencja potrafi teraz tworzyć teksty równie dobrze jak ludzie, a czasem nawet lepiej, może zautomatyzować częściowo lub całkowicie każde zadanie wymagające komunikacji, czyli prawie wszystko. AI jest używana do pisania maili, odpowiadania na zapytania klientów i wyjaśniania złożonych umów. Każdy posiadacz komputera ma dostęp do narzędzi umożliwiających natychmiastowe generowanie spersonalizowanych, wysokiej jakości obrazów i filmów, które mogą wspierać tworzenie materiałów marketingowych, edycję zdjęć biznesowych, wizualizację koncepcji artystycznych, ilustrowanie książek itd. Sztuczna inteligencja może być nawet wykorzystywana do syntezy danych szkoleniowych, opracowywania algorytmów i pisania kodu, co z kolei pomoże w trenowaniu jeszcze bardziej wydajnych modeli w przyszłości.

Czynnik 2. Zwiększone inwestycje w sztuczną inteligencję

Sukces ChatGPT spowodował gwałtowny wzrost inwestycji w sztuczną inteligencję, zarówno ze strony funduszy venture capital, jak i przedsiębiorstw. W miarę jak aplikacje AI stają się coraz tańsze do zbudowania i szybciej trafiają na rynek, zwroty z inwestycji w AI stają się bardziej atrakcyjne. Firmy spieszą się z włączaniem sztucznej inteligencji do swoich produktów i procesów. Matt Ross, starszy menedżer ds. badań stosowanych w Scribd, poinformował mnie, że szacunkowy koszt związany z wykorzystaniem sztucznej inteligencji w jego projektach spadł stukrotnie (o dwa rzędy wielkości) między kwietniem 2022 a kwietniem 2023 roku.

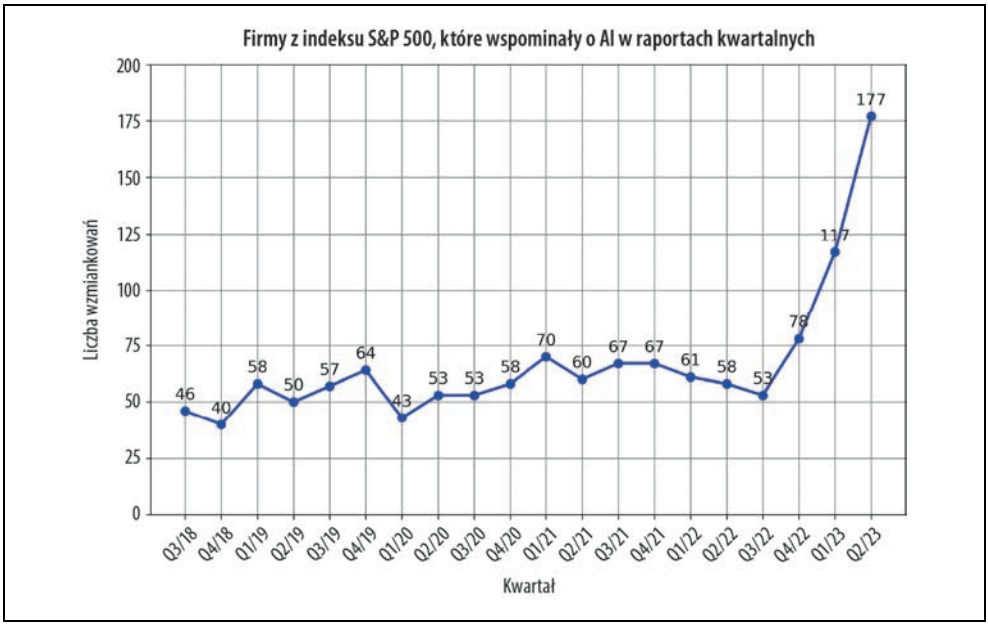
Firma Goldman Sachs Research (*https://oreil.ly/okMw6*) oszacowała, że do 2025 roku inwestycje w sztuczną inteligencję mogą zbliżyć się do 100 miliardów dolarów w USA i 200 miliardów dolarów na całym świecie[9]. Sztuczna inteligencja jest często wymieniana jako przewaga konkurencyjna. Firma FactSet (*https://oreil.ly/tgm-a*) stwierdziła, że jedna na trzy spółki z indeksu S&P 500 wspominała o sztucznej inteligencji w swoich komunikatach o wynikach za drugi kwartał 2023 roku, czyli trzy razy więcej niż rok wcześniej. Rysunek 1.5 pokazuje liczbę firm z indeksu S&P 500, które wspominały o AI w swoich raportach kwartalnych w latach 2018 – 2023.

Według danych WallStreetZen firmy, które wspomniały o sztucznej inteligencji w swoich raportach kwartalnych, zanotowały wyższy wzrost cen akcji (średnio o 4,6% w porównaniu do 2,4% (*https://oreil.ly/fK5uh*)) niż te, które tego nie zrobiły. Nie jest jednak jasne, czy to zależność przyczynowa (AI sprawia, że te firmy odnoszą większy sukces), czy korelacja (firmy odnoszą sukces, ponieważ szybko adaptują się do nowych technologii).

Czynnik 3. Łatwiejsze tworzenie aplikacji AI

Model usługowy, spopularyzowany przez OpenAI i innych dostawców modeli, upraszcza wykorzystanie sztucznej inteligencji do tworzenia aplikacji. Modele są udostępniane poprzez interfejsy API, które przyjmują zapytania użytkowników i zwracają wyniki. Korzystanie z modelu AI bez interfejsu API wymaga odpowiedniej infrastruktury do jego przechowywania i obsługi. Interfejs API zapewnia dostęp do zaawansowanych modeli za pomocą prostych wywołań funkcji.

[9] Dla porównania, całkowite wydatki Stanów Zjednoczonych na publiczne szkoły podstawowe i średnie wynoszą około 900 miliardów dolarów, co stanowi tylko dziewięciokrotność inwestycji w sztuczną inteligencję.

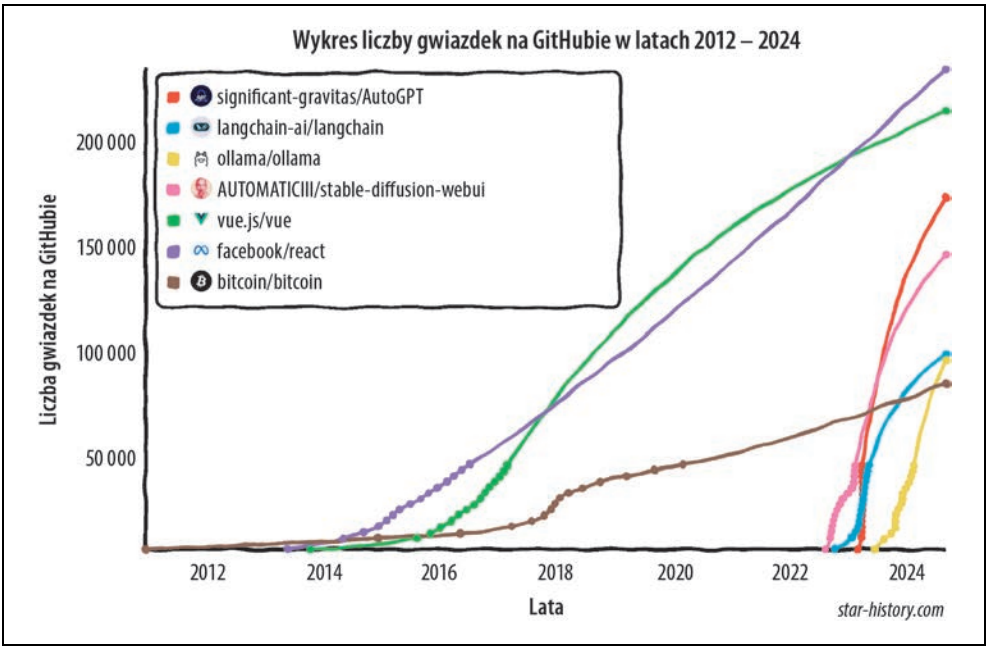


Rysunek 1.5. Liczba firm wchodzących w skład indeksu S&P 500, które wspomniały o sztucznej inteligencji w swoich raportach kwartalnych, osiągnęła rekordowy poziom w 2023 roku. Dane pochodzą z FactSet

> Co więcej, AI umożliwia budowanie aplikacji przy minimalnym programowaniu. Po pierwsze, AI może pisać kod za Ciebie, co umożliwia osobom bez wykształcenia inżynierskiego szybkie przekształcenie pomysłów w kod i zaprezentowanie ich użytkownikom. Po drugie, możesz współpracować z tymi modelami, używając prostego języka naturalnego zamiast języka programowania. *Teraz każdy, i mam na myśli rzeczywiście każdego człowieka, może tworzyć aplikacje AI.*

Ze względu na zasoby potrzebne do opracowania modeli podstawowych proces ten jest możliwy tylko dla dużych korporacji (Google, Meta, Microsoft, Baidu, Tencent), rządów (Japonia (*https://oreil.ly/r86Qz*), ZEA (*https://oreil.ly/IUcVg*)) oraz ambitnych, dobrze finansowanych startupów (OpenAI, Anthropic, Mistral). W wywiadzie z września 2022 roku Sam Altman, dyrektor generalny OpenAI (*https://oreil.ly/D9QBM*), powiedział, że najlepszym rozwiązaniem będzie adaptacja tych modeli do specyficznych zastosowań.

Świat dostrzega tę szansę. Inżynieria AI prędko stała się jedną z najszybciej rozwijających się dziedzin inżynierii. Bardzo prawdopodobne jest nawet, że rozwija się najszybciej ze wszystkich. Narzędzia wykorzystywane w inżynierii AI zyskują popularność szybciej niż jakiekolwiek wcześniejsze narzędzia inżynierii oprogramowania. W ciągu zaledwie dwóch lat cztery narzędzia o otwartym oprogramowaniu, używane w inżynierii AI (AutoGPT, Stable Diffusion eb UI, LangChain, Ollama), zdobyły już więcej gwiazdek na GitHubie niż bitcoin. Są na dobrej drodze, by pod względem liczby gwiazdek wyprzedzić nawet najpopularniejsze biblioteki do tworzenia aplikacji webowych, w tym React i Vue. Rysunek 1.6 przedstawia wzrost liczby gwiazdek na GitHubie dla narzędzi służących do inżynierii AI w porównaniu z bitcoinem, Vue i Reactem.



Rysunek 1.6. Narzędzia o otwartym oprogramowaniu używane w inżynierii AI zyskują na popularności szybciej niż jakiekolwiek inne narzędzia inżynierii oprogramowania (na podstawie liczby gwiazdek na GitHubie)

Z badania przeprowadzonego w sierpniu 2023 roku przez firmę LinkedIn wynika, że liczba profesjonalistów dodających do swoich profili określenia takie jak „AI generatywna", „ChatGPT", „Inżynieria promptów" i „Projektowanie promptów" rosła średnio o 75% w każdym kolejnym miesiącu (*https://oreil.ly/m8SvB*). Serwis ComputerWorld (*https://oreil.ly/47sGE*) stwierdził, że „nauczanie AI odpowiednich zachowań to najszybciej rozwijająca się umiejętność zawodowa".

Dynamicznie rozwijająca się społeczność inżynierów AI wykazała się ogromną kreatywnością, tworząc szeroki wachlarz fascynujących aplikacji. W następnym podrozdziale przyjrzymy się niektórym z najpopularniejszych wzorców zastosowań.

Przykłady zastosowań modeli podstawowych

Jeśli jeszcze nie tworzysz aplikacji AI, mam nadzieję, że po przeczytaniu poprzedniego podrozdziału będziesz chciał to robić. Jeśli masz już pomysł na aplikację, możesz od razu przejść do podrozdziału „Planowanie aplikacji AI". Jeśli potrzebujesz inspiracji, w tym podrozdziale zapoznasz się z szerokim wachlarzem zastosowań, które zostały sprawdzone w branży i charakteryzują się ogromnym potencjałem.

Dlaczego używam terminu „inżynieria AI”?

Istnieje wiele terminów opisujących proces tworzenia aplikacji opartych na modelach podstawowych, takich jak inżynieria ML, MLOps, AIOps, LLMOps itp. Dlaczego w tej książce zdecydowałam się używać terminu „inżynieria AI”?

Nie wybrałam określenia „inżynieria ML”, ponieważ, jak wyjaśniono w dalszym punkcie „Inżynieria AI a inżynieria ML”, praca z modelami podstawowymi różni się od obsługi tradycyjnych modeli ML w kilku kluczowych aspektach. Użycie terminu „inżynieria ML” nie wystarczy, by uchwycić tę różnicę. Niemniej jednak jest to dobre określenie, które obejmuje oba procesy.

Nie zdecydowałam się na nazwy kończące się na „Ops”, ponieważ, choć proces ma komponenty operacyjne, główny nacisk kładzie się na dostosowywanie (inżynierię) modeli podstawowych, by realizowały nasze cele.

Przeprowadziłam także ankietę wśród 20 osób, które rozwijały aplikacje oparte na modelach podstawowych. Zapytałam, jakim określeniem nazwałyby swoją działalność. Większość z nich wybrała „inżynierię AI”. Zdecydowałam się zaufać tej opinii.

Liczba możliwych aplikacji, które można utworzyć z wykorzystaniem modeli podstawowych, wydaje się niemal nieskończona. Bez względu na to, jaki przypadek użycia przyjdzie Ci do głowy, jest duża szansa, że już istnieje odpowiednie rozwiązanie[10]. Wymienienie wszystkich możliwych przypadków użycia AI jest wręcz niemożliwe.

Nawet próba kategoryzacji tych przypadków użycia jest trudna, ponieważ różne badania stosują różne klasyfikacje. Na przykład firma Amazon Web Services (AWS) (*https://oreil.ly/-k_QX*) podzieliła przypadki użycia AI generatywnego w firmach na trzy grupy: doświadczenie klienta, produktywność pracowników i optymalizacja procesów. Badanie O’Reilly z 2024 roku sklasyfikowało przypadki użycia w ośmiu kategoriach: programowanie, analiza danych, obsługa klienta, teksty marketingowe, inne teksty, badania, projektowanie stron internetowych i sztuka.

Niektóre organizacje, takie jak Deloitte (*https://oreil.ly/T272_*) podzieliły przypadki użycia na podstawie wartości, jaką mogą przynieść, na przykład redukcję kosztów, poprawę efektywności procesów, wzrost czy przyspieszanie innowacji. Gartner (*https://oreil.ly/OyIUP*) ma kategorię *ciągłości biznesowej*, co oznacza, że firma może upaść, jeśli nie wdroży generatywnego AI. Z przeprowadzonego przez Gartnera badania w 2023 roku wynika, że 7% spośród 2500 ankietowanych menedżerów wskazało ciągłość biznesową jako powód przyjęcia generatywnego AI.

Eloundou i in. (2023) (*https://arxiv.org/abs/2303.10130*) przeprowadzili interesujące badania na temat stopnia, w jakim różne zawody są podatne na wpływ AI. Zdefiniowali zadanie jako podatne, jeśli AI i oprogramowanie oparte na sztucznej inteligencji mogą skrócić czas potrzebny do jego wykonania o przynajmniej 50%. Zawód z 80% podatnością oznacza, że 80% zadań w tym zawodzie jest podatnych na automatyzację przez AI. Według badania do zawodów z całkowitą lub

[10] Ciekawostka: na dzień 16 września 2024 roku strona theresanaiforthat.com wymienia 16 814 aplikacji AI używanych w 14 688 zadaniach i 4803 zastosowaniach.

niemal całkowitą podatnością należą tłumacze, podatkowe biura rachunkowe, projektanci stron internetowych oraz pisarze. Przykłady tych zawodów zostały przedstawione w tabeli 1.2. Nic dziwnego, że do zawodów, które wcale nie są podatne na AI, zalicza się kucharzy, kamieniarzy i sportowców. To badanie dostarcza cennych wskazówek dotyczących zastosowań sztucznej inteligencji.

Tabela 1.2. Zawody o najwyższej podatności na AI. Grupa α odnosi się do bezpośredniej podatności na modele AI, podczas gdy grupy β i ζ dotyczą podatności na oprogramowanie wspierane przez AI. Tabela pochodzi od Eloundou i in. (2023)

Grupa	Najbardziej podatne zawody	% podatności
α	tłumacze (pisemni i ustni)	76,5
	analitycy ankiet	75,0
	poeci, autorzy tekstów piosenek i pisarze kreatywni	68,8
	naukowcy zajmujący się zwierzętami	66,7
	specjaliści ds. public relations	66,7
β	analitycy ankiet	84,4
	pisarze i autorzy	82,5
	tłumacze (pisemni i ustni)	82,4
	specjaliści ds. public relations	80,6
	naukowcy zajmujący się zwierzętami	77,8
ζ	Matematycy	100,0
	Podatkowe biura rachunkowe	100,0
	Analitycy finansowi zajmujący się analizami ilościowymi	100,0
	Pisarze i autorzy	100,0
	Projektanci stron internetowych i interfejsów cyfrowych	100,0
	Ludzie zidentyfikowali 15 zawodów jako „w pełni podatne”	

Podczas analizy przypadków użycia przyjrzałam się zarówno aplikacjom dla firm, jak i konsumentów. Aby zrozumieć przypadki użycia w przedsiębiorstwach, przeprowadziłam wywiady z przedstawicielami 50 firm na temat ich strategii AI oraz zapoznałam się z ponad 100 studiami przypadków. By poznać aplikacje konsumenckie, zbadałam 205 aplikacji AI opartych na otwartych źródłach, które zdobyły co najmniej 500 gwiazdek na GitHubie[11]. Jak przedstawiono w tabeli 1.3, aplikacje zostały podzielone na osiem grup. Ta lista jest ograniczona, ale pełni rolę punktu odniesienia. W rozdziale 2. będziesz zgłębiać sposób tworzenia modeli podstawowych, a w rozdziale 3. ich ewaluacji, dlatego lepiej zrozumiesz, w jakich przypadkach użycia modele podstawowe mogą i powinny być wykorzystywane.

[11] Odkrywanie różnych zastosowań sztucznej inteligencji było jednym z moich ulubionych zajęć podczas pisania tej książki. Obserwowanie, co tworzą różni ludzie, jest niesamowicie wciągające. Ty również możesz się zapoznać z listą aplikacji AI opartych na otwartych źródłach, które śledzę (*https://huyenchip.com/llama-police*). Lista jest aktualizowana co 12 godzin.

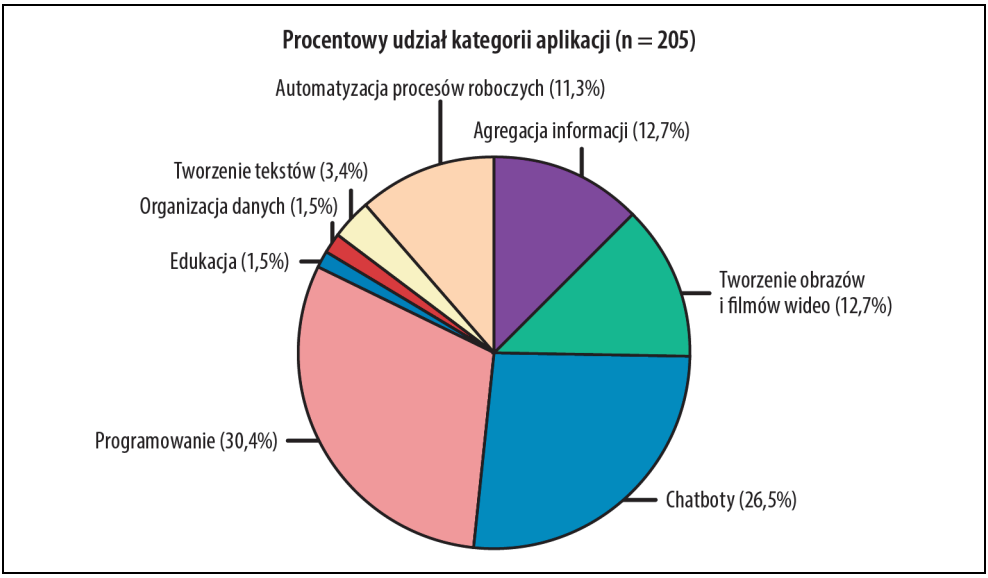
Tabela 1.3. Najczęstsze przypadki użycia generatywnej sztucznej inteligencji w aplikacjach konsumenckich i firmowych

Kategoria	Przykłady przypadków użycia w aplikacjach konsumenckich	Przykłady przypadków użycia w aplikacjach firmowych
programowanie	programowanie	programowanie
tworzenie obrazów i filmów wideo	edycja obrazów i plików wideo	prezentacje
	projektowanie	tworzenie reklam
tworzenie tekstów	e-maile	pisanie tekstów, optymalizacja pod kątem wyszukiwarek (seo)
	posty w mediach społecznościowych i blogach	raporty, notatki, dokumenty projektowe
edukacja	korepetycje	wprowadzenie pracowników
	ocena esejów	szkolenia podnoszące kwalifikacje pracowników
boty konwersacyjne	chatbot uniwersalny	wsparcie klienta
	wirtualny znajomy ai	asystenci produktowi
agregowanie informacji	podsumowywanie	podsumowywanie
	analiza dokumentów	badania rynku
organizacja danych	wyszukiwanie obrazów	zarządzanie wiedzą
	memex	przetwarzanie dokumentów
automatyzacja procesów	planowanie podróży	ekstrakcja, wprowadzanie i opisywanie danych
	planowanie wydarzeń	generowanie leadów

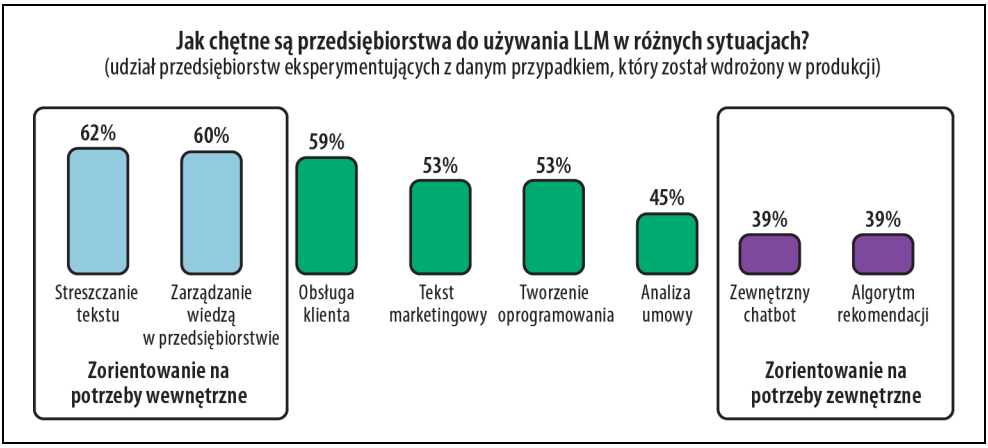
Ponieważ modele podstawowe są uniwersalne, oparte na nich aplikacje mogą rozwiązywać wiele różnych problemów. Oznacza to, że aplikacja może należeć do więcej niż jednej kategorii. Na przykład bot zapewnia towarzystwo i agreguje informacje. Aplikacja może pomóc w ekstrakcji danych strukturalnych z pliku PDF i odpowiadać na związane z nim pytania.

Na rysunku 1.7 pokazano rozkład tych przypadków użycia wśród 205 aplikacji o otwartym oprogramowaniu. Należy zauważyć, że niewielki procent przypadków użycia związanych z dziedzinami edukacji, organizowania danych i tworzenia tekstów nie oznacza, że są one niepopularne. Po prostu takie aplikacje nie zostały utworzone w oparciu o otwarte oprogramowanie. Ich twórcy mogli uznać je za bardziej odpowiednie dla zastosowań w przedsiębiorstwach.

Świat przedsiębiorstw zazwyczaj preferuje aplikacje o niższym ryzyku. Na przykład raport z 2024 roku wykonany przez firmę a16z (*https://oreil.ly/XWeDt*) wykazał, że przedsiębiorstwa szybciej wdrażają aplikacje skierowane do użytku wewnętrznego (zarządzanie wiedzą wewnętrzną) niż te, które są przeznaczone do użytku zewnętrznego (chatboty do obsługi klienta), co ilustruje rysunek 1.8. Aplikacje wewnętrzne pomagają firmom w rozwijaniu swoich kompetencji w zakresie inżynierii AI, a jednocześnie ograniczają ryzyko związane z prywatnością danych, zgodnością z przepisami oraz potencjalnymi awariami o katastrofalnych skutkach. Chociaż modele



Rysunek 1.7. Rozkład przypadków użycia dla 205 repozytoriów oprogramowania o otwartych źródłach na GitHubie



Rysunek 1.8. Firmy są bardziej skłonne do wdrażania aplikacji zorientowanych na potrzeby wewnętrzne

podstawowe są wszechstronne i mogą być wykorzystywane do różnych zadań, wiele opartych na nich aplikacji ma wciąż zamknięty charakter — na przykład klasyfikacja. Zadania związane z klasyfikacją są prostsze do oceny, co sprawia, że łatwiej oszacować związane z nimi ryzyko.

Mimo że widziałam już setki aplikacji AI, co tydzień wciąż napotykam nowe, które mnie zadziwiają. Na początku istnienia internetu mało kto przewidywał, że to media społecznościowe staną się dominującym przypadkiem użycia w sieci. Gdy nauczymy się w pełni wykorzystywać AI, przypadek użycia, który ostatecznie stanie się najważniejszy, może nas zaskoczyć. Mam nadzieję, że zaskoczenie to będzie pozytywne.

Programowanie

Programowanie jest zdecydowanie najpopularniejszym przypadkiem zastosowania w wielu badaniach dotyczących generatywnej sztucznej inteligencji. Narzędzia AI do programowania cieszą się popularnością względu na umiejętności sztucznej inteligencji, a także dlatego, że początkujący inżynierowie AI to programiści, którzy mają większe doświadczenie z wyzwaniami związanymi z kodowaniem.

Jednym z pierwszych realnych sukcesów modeli podstawowych było narzędzie do uzupełniania kodu GitHub Copilot, które osiągnęło roczne przychody przekraczające 100 milionów dolarów (*https://oreil.ly/Xamik*) jedynie dwa lata po jego udostępnieniu. W czasie pisania tego tekstu (w sierpniu 2024 roku) startupy związane z programistyczną dziedziną AI zarobiły setki milionów dolarów: firma Magic pozyskała 320 milionów dolarów (*https://oreil.ly/t0xDf*), a Anysphere 60 milionów dolarów (*https://oreil.ly/BW5Hk*). Oprogramowanie o otwartych źródłach, takie jak gpt-engineer (*https://github.com/gpt-engineer-org/gpt-engineer*) i screenshot-to-code (*https://github.com/abi/screenshot-to-code*), zdobyło w ciągu roku 50 tysięcy gwiazdek na GitHubie. Kolejne narzędzia pojawiają się w szybkim tempie.

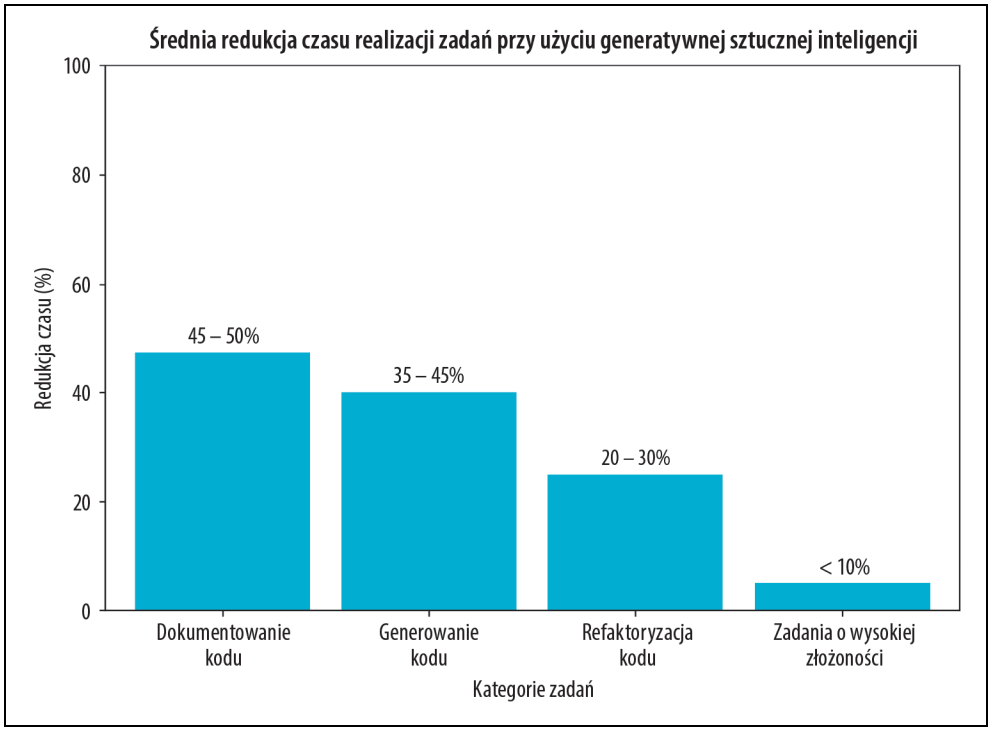
Oprócz ogólnych narzędzi do programowania istnieje wiele specjalistycznych, które koncentrują się na określonych zadaniach. Oto przykłady takich zadań:

- Wydobywanie danych strukturalnych ze stron internetowych i plików PDF (AgentGPT (*https://github.com/reworkd/AgentGPT*)).
- Konwertowanie języka angielskiego na kod (DB-GPT (*https://github.com/eosphoros-ai/DB-GPT*), SQL Chat (*https://github.com/sqlchat/sqlchat*), PandasAI (*https://github.com/Sinaptik-AI/pandas-ai*)).
- Generowanie kodu na podstawie projektu lub zrzutu ekranu, który z kolei utworzy stronę internetową wyglądającą jak podany obraz (screenshot-to-code, draw-a-ui (*https://github.com/sawyerhood/draw-a-ui*)).
- Tłumaczenie kodu z jednego języka programowania lub pakietu na inny (GPT-Migrate (*https://github.com/joshpxyne/gpt-migrate*), AI Code Translator (*https://github.com/mckay-wrigley/ai-code-translator*)).
- Tworzenie dokumentacji (Autodoc (*https://github.com/context-labs/autodoc*)).
- Tworzenie testów (PentestGPT (*https://github.com/GreyDGL/PentestGPT*)).
- Generowanie opisów dla commitów w Gicie (AI Commits (*https://github.com/Nutlope/aicommits*)).

To jasne, że sztuczna inteligencja może wykonać wiele zadań inżynierii oprogramowania. Pytanie brzmi, czy AI może zautomatyzować całą inżynierię oprogramowania. Z jednej strony Jensen Huang, dyrektor generalny firmy NVIDIA (*https://oreil.ly/zUpGu*), przewiduje, że AI zastąpi ludzkich inżynierów oprogramowania, a my powinniśmy przestać mówić, że dzieci mają się uczyć programowania. W ujawnionym nagraniu prezes AWS Matt Garman (*https://oreil.ly/Hz_3i*) podzielił się opinią, że w niedalekiej przyszłości większość programistów przestanie kodować. Nie oznacza to końca zawodu programisty — chodzi po prostu o to, że ich rola się zmieni.

Z drugiej strony wielu inżynierów oprogramowania jest przekonanych, że nigdy nie zostaną zastąpieni przez AI — zarówno z powodów technicznych, jak i emocjonalnych (ludzie niechętnie przyznają, że ich rola może zostać przejęta przez sztuczną inteligencję).

Inżynieria oprogramowania obejmuje wiele różnych zadań. Sztuczna inteligencja z niektórymi z nich radzi sobie lepiej, z innymi gorzej. Badania przeprowadzone przez firmę McKinsey (*https://oreil.ly/aqUmX*) wykazały, że AI może pomóc programistom osiągnąć dwukrotnie wyższą wydajność w tworzeniu dokumentacji oraz poprawić produktywność o 25 – 50% podczas generowania kodu i refaktoryzacji. Jak pokazano na rysunku 1.9, w przypadku bardziej złożonych zadań zaobserwowano minimalny wzrost wydajności. Gdy rozmawiam z twórcami narzędzi AI do kodowania, często słyszę, że sztuczna inteligencja radzi sobie znacznie lepiej z programowaniem warstwy wizualnej niż serwerowej.



Rysunek 1.9. AI może znacznie zwiększyć produktywność programistów, szczególnie w przypadku prostych zadań, ale ma mniejsze zastosowanie w zadaniach o wysokiej złożoności. Dane: McKinsey

Niezależnie od tego, czy AI zastąpi inżynierów oprogramowania, z pewnością może zwiększyć ich produktywność. Oznacza to, że firmy mogą teraz osiągać lepsze wyniki, zatrudniając mniejszą liczbę programistów. AI może również wpłynąć na branżę outsourcingu, ponieważ zadania zlecane są zazwyczaj prostsze i niezwiązane z kluczową działalnością firmy.

Tworzenie obrazów i wideo

Dzięki probabilistycznej naturze sztuczna inteligencja doskonale sprawdza się w zadaniach kreatywnych. Wiele z najczęściej odnoszących sukcesy startupów AI to aplikacje o charakterze kreatywnym, takie jak Midjourney służąca do generowania obrazów, Adobe Firefly do edycji zdjęć oraz Runway, Pika Labs i Sora do tworzenia wideo. Pod koniec 2023 roku istniejąca jedynie półtora roku aplikacja Midjourney (*https://oreil.ly/EAzCl*) osiągnęła już 200 milionów dolarów rocznych przychodów cyklicznych. W grudniu 2023 roku połowa z 10 najlepszych darmowych aplikacji w kategorii „Grafika i projektowanie", dostępnej na platformie App Store firmy Apple, zawierała w nazwie słowo „AI". Sądzę, że wkrótce aplikacje graficzne i projektowe będą automatycznie wykorzystywały sztuczną inteligencję, a słowo „AI" w ich nazwach stanie się zbędne. W rozdziale 2. szczegółowo przeanalizowano probabilistyczną naturę AI.

Z AI coraz powszechniej korzysta się podczas tworzenia zdjęć profilowych dla mediów społecznościowych — poczynając od LinkedIn, a kończąc na TikToku. Wiele osób ubiegających się o pracę uważa, że zdjęcia generowane przez AI mogą pomóc im lepiej się zaprezentować i zwiększyć szanse na zatrudnienie (*https://oreil.ly/fZLVg*). Postrzeganie zdjęć profilowych generowanych przez sztuczną inteligencję uległo znaczącej zmianie. W 2019 roku Facebook (*https://oreil.ly/WNqUw*) z powodów bezpieczeństwa zablokował konta używające zdjęć wygenerowanych przez AI. W 2023 roku wiele aplikacji społecznościowych umożliwia użytkownikom tworzenie zdjęć profilowych przy użyciu sztucznej inteligencji.

Działy reklamy i marketingu w wielu firmach szybko zaadaptowały AI[12]. Sztuczna inteligencja może być wykorzystywana do bezpośredniego generowania obrazów i klipów reklamowych. Może też wspierać proces twórczy, dostarczając wstępnych koncepcji, które później zostaną rozwinięte przez ekspertów. AI może być używana do tworzenia wielu wersji reklam, a następnie testowania, na którą z nich najlepiej reagują odbiorcy. Ponadto może również dostosować reklamy do pór roku i lokalizacji, na przykład poprzez zmianę kolorów liści w czasie jesieni lub umieszczanie śniegu na ziemi zimą.

Generowanie tekstów

Sztuczna inteligencja jest od dawna wykorzystywana do wspomagania pisania. Jeśli korzystasz ze smartfona, prawdopodobnie wiesz, co to autokorekta i autouzupełnianie. Obie te opcje są wspierane przez sztuczną inteligencję. Tworzenie tekstów to idealne zastosowanie dla sztucznej inteligencji, ponieważ tę czynność wykonujemy często, może być ona dość żmudna, a poza tym jesteśmy bardziej wyrozumiali wobec błędów popełnianych przez AI. Jeśli model sugeruje coś, co Ci się nie podoba, możesz to po prostu zignorować.

Nie jest zaskoczeniem, że modele LLM dobrze się spisują w takich sytuacjach, ponieważ są one po prostu trenowane do uzupełniania tekstów. Aby sprawdzić, jaki wpływ na pracę może mieć użycie modelu ChatGPT podczas pisania, uczelnia MIT przeprowadziła badanie (Noy

[12] Ponieważ przedsiębiorstwa zazwyczaj przeznaczają znaczne środki na reklamy i marketing, ich automatyzacja może przynieść ogromne oszczędności. Średnio 11% budżetu firmy jest wydawane na marketing. Patrz *Marketing Budgets Vary by Industry* (Christine Moorman, WSJ, 2017 (*https://oreil.ly/D0-yA*)).

i Zhang, 2023 (*https://oreil.ly/IzQ6F*)). Polegało ono na przydzieleniu pewnych zadań 453 profesjonalistom wykształconym na poziomie college'u, przy czym losowo wybranej połowie osób udostępniono ChatGPT. Wyniki pokazują, że w przypadku osób wykorzystujących ChatGPT średni czas tworzenia tekstów zmniejszył się o 40%, a jakość wyników wzrosła o 18%. ChatGPT pomaga w wyrównywaniu poziomu jakości pracy, co oznacza, że jest szczególnie pomocny dla tych, którzy nie mają naturalnych umiejętności w danej dziedzinie. Osoby, które miały dostęp do ChatGPT podczas przeprowadzania eksperymentu, dwa tygodnie później były dwukrotnie bardziej skłonne do korzystania z niego w swojej pracy, a 1,6 razy po dwóch miesiącach.

W przypadku konsumentów korzyści są oczywiste. Wiele osób używa sztucznej inteligencji, aby się lepiej komunikować. Kipiąc ze złości, możesz napisać obraźliwego maila, ale AI sprawi, że jego treść będzie uprzejma. Możesz podać kilka punktów, a AI przekształci je w pełne akapity. Niektórzy twierdzą, że nie wysyłają już ważnych maili bez wcześniejszego poproszenia sztucznej inteligencji o ich poprawienie.

Studenci wykorzystują AI do pisania wypracowań. Pisarze używają jej do tworzenia książek[13]. Wiele startupów już teraz korzysta ze sztucznej inteligencji do generowania książek dla dzieci, fan fiction, romansów i fantastyki. W przeciwieństwie do tradycyjnych książek te generowane przez AI mogą być interaktywne, ponieważ fabuła może się zmieniać w zależności od preferencji czytelnika. Oznacza to, że czytelnicy mogą aktywnie uczestniczyć w tworzeniu historii, którą czytają. Aplikacja do nauki czytania dla dzieci identyfikuje słowa, z którymi dziecko ma trudności, a następnie generuje opowieści skoncentrowane na nich.

Aplikacje do tworzenia notatek i maili, takie jak Google Docs, Notion czy Gmail, wykorzystują AI, aby pomagać użytkownikom w poprawie jakości ich tekstów. Aplikacja Grammarly (*https://arxiv.org/abs/2305.09857*), asystująca w pisaniu, dostraja model AI, aby teksty były bardziej płynne, spójne i przejrzyste.

Takie umiejętności sztucznej inteligencji mogą być jednak również wykorzystywane w niewłaściwy sposób. W 2023 roku „New York Times" (*https://oreil.ly/LB72P*) informował, że na Amazonie pojawiła się ogromna liczba przewodników turystycznych niskiej jakości, wygenerowanych przez AI, z fikcyjnymi biografiami autorów, stronami internetowymi i entuzjastycznymi recenzjami — wszystko to utworzone przez sztuczną inteligencję.

W przedsiębiorstwach AI jest powszechnie stosowana w sprzedaży, marketingu i ogólnej komunikacji zespołowej. Wielu menedżerów przyznało, że korzysta ze sztucznej inteligencji do tworzenia raportów oceniających pracę. AI może pomóc w tworzeniu skutecznych maili sprzedażowych, tekstów reklamowych i opisów produktów. Aplikacje do zarządzania relacjami z klientami (CRM), takie jak HubSpot i Salesforce, oferują narzędzia umożliwiające użytkownikom biznesowym generowanie treści umieszczanych na stronach internetowych oraz maili marketingowych.

[13] Sztuczna inteligencja stanowiła dla mnie ogromne wsparcie podczas pisania tej książki. Uważam, że może ona zautomatyzować wiele etapów tego procesu. Pisząc fikcję, często proszę AI o pomoc w wymyślaniu pomysłów dotyczących tego, co powinno się dalej wydarzyć lub w jaki sposób bohater ma zareagować na daną sytuację. Wciąż analizuję, które aspekty pisania da się zautomatyzować, a które pozostaną w rękach człowieka.

Sztuczna inteligencja wydaje się szczególnie skuteczna w zakresie SEO (ang. *Search Engine Optimization* — optymalizacja pod kątem wyszukiwarek), być może dlatego, że wiele modeli AI jest trenowanych na danych z internetu, który obfituje w teksty przystosowane do lepszego pozycjonowania. AI radzi sobie z SEO na tyle dobrze, że umożliwiła powstanie nowej generacji farm treści. Takie farmy zawierają strony o niskiej jakości i wypełniają je treściami generowanymi przez AI, aby osiągnąć wysokie pozycje w wyszukiwarce Google i przyciągnąć ruch. Następnie sprzedają przestrzeń reklamową za pośrednictwem giełd reklamowych. W czerwcu 2023 roku firma NewsGuard (*https://oreil.ly/mZKjr*) zidentyfikowała niemal 400 reklam od 141 popularnych marek na stronach z treściami generowanymi przez AI. Jedna z takich stron produkowała aż 1200 artykułów dziennie. Jeśli nie zostaną podjęte żadne działania ograniczające ten proceder, przyszłość treści internetowych będzie zdominowana przez AI. Jest to dość ponura wizja[14].

Edukacja

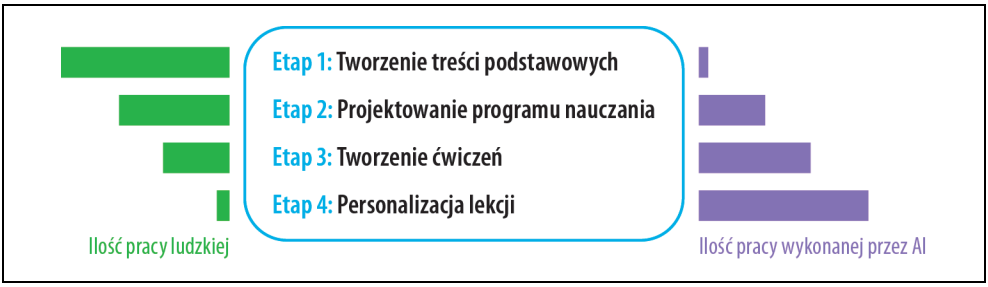
Za każdym razem, gdy model ChatGPT jest niedostępny, serwer Discord firmy OpenAI jest zasypywany skargami uczniów, którzy nie mogą ukończyć swoich prac domowych. Kilka rad pedagogicznych, w tym szkoły publiczne w Nowym Jorku oraz Zjednoczony Dystrykt Szkół Los Angeles, szybko zabroniło korzystania z usług oferowanych przez ChatGPT (*https://oreil.ly/pqI5z*) z obawy przed ich nadużywaniem, ale kilka miesięcy później zmieniło swoją decyzję (*https://oreil.ly/nxtzw*).

Zamiast zakazywać sztucznej inteligencji, szkoły mogłyby wprowadzić ją do procesu nauczania, aby pomóc uczniom szybciej zdobywać wiedzę. AI może streszczać podręczniki i tworzyć spersonalizowane plany wykładowe dla każdego ucznia. Reklamy od dawna są już spersonalizowane, ponieważ wiemy, że każda osoba jest inna — a edukacja wciąż nie jest. AI może pomóc dostosować materiały do formatu najlepiej odpowiadającego potrzebom każdego ucznia. Uczniowie preferujący naukę słuchową mogą poprosić sztuczną inteligencję, aby przeczytała im materiały na głos. Ci, którzy kochają zwierzęta, mogą poprosić AI o dostosowanie wizualizacji, by umieściła je na nich. Osoby, które łatwiej rozumieją kod niż równania matematyczne, mogą poprosić AI o przetłumaczenie tych równań na odpowiedni kod.

Sztuczna inteligencja jest szczególnie pomocna w nauce języków obcych, ponieważ można ją poprosić o tworzenie różnych scenariuszy do ćwiczeń. Pajak i Bicknell (Duolingo, 2022) (*https://oreil.ly/C8kmI*) stwierdzili, że spośród czterech etapów tworzenia kursu personalizacja lekcji jest tym, który może najbardziej skorzystać na zastosowaniu AI (patrz rysunek 1.10).

Sztuczna inteligencja może tworzyć quizy (zarówno wielokrotnego wyboru, jak i otwarte), a także oceniać odpowiedzi. Może również pełnić rolę partnera w debacie, ponieważ jest znacznie lepsza od przeciętnego człowieka w przedstawianiu różnych perspektyw dotyczących tego samego zagadnienia. Na przykład Khan Academy (*https://oreil.ly/tC7-g*) oferuje uczniom asystentów edukacyjnych wspomaganych przez AI, a nauczycielom asystentów kursów (*https://oreil.ly/_N1JR*). Innowacyjna metoda nauczania polega na wygenerowaniu przez sztuczną inteligencję wypracowań, a następnie przekazaniu ich uczniom, którzy mają za zadanie znaleźć w nich błędy oraz je poprawić.

[14] Uważam, że z czasem staniemy się tak sceptyczni wobec treści w internecie, iż będziemy sięgać jedynie po te tworzone przez osoby lub marki, którym naprawdę ufamy.



Rysunek 1.10. Sztuczna inteligencja może być wykorzystywana we wszystkich czterech etapach tworzenia kursu na portalu Duolingo, jednak najbardziej pomocna jest na etapie personalizacji. Rysunek pochodzi od Pajaka i Bicknella (Duolingo, 2022)

Wiele firm edukacyjnych z powodzeniem wprowadza sztuczną inteligencję w celu tworzenia lepszych produktów, jednak niektóre z nich doświadczyły negatywnych skutków, ponieważ AI zaczęło przejmować ich rolę. Przykładem jest firma Chegg, pomagająca uczniom w odrabianiu prac domowych, która doświadczyła spadku swojej wartości rynkowej — z 28 dolarów za akcję w listopadzie 2022 roku do 2 dolarów we wrześniu 2024 roku. Powodem było to, że uczniowie zaczęli powszechnie korzystać z AI.

Jeśli zagrożeniem jest sytuacja, gdy AI zastępuje wiele umiejętności, to szansą jest jej wykorzystanie jako narzędzia do ich nauki. Sztuczna inteligencja często może pomóc w szybkim przyswojeniu wiedzy, a następnie umożliwia dalszy rozwój, prowadzący nawet do prześcignięcia jej zdolności.

Boty konwersacyjne

Boty konwersacyjne są wyjątkowo uniwersalne. Potrafią pomóc w poszukiwaniach informacji, wyjaśnianiu pojęć oraz tworzeniu nowych pomysłów. Sztuczna inteligencja może pełnić rolę towarzysza, a także terapeuty, odwzorowując różne osobowości, co pozwala na prowadzenie rozmów z cyfrowym odpowiednikiem dowolnej osoby. Wirtualni partnerzy, zarówno w wersji kobiecej, jak i męskiej, zdobyli popularność w zaskakująco krótkim czasie. Coraz więcej osób spędza więcej czasu, rozmawiając z botami niż z ludźmi (patrz odpowiednie analizy: *https://oreil.ly/dZbym* oraz *https://oreil.ly/svWj8*). Istnieje także obawa, że AI zrewolucjonizuje (*https://oreil.ly/SNme7*) randkowanie (*https://oreil.ly/Jbt4R*).

W badaniach naukowych odkryto, że grupy botów konwersacyjnych mogą być wykorzystywane do symulowania społeczeństwa, co umożliwia badanie dynamiki społecznej (Park i in., 2023 (*https://arxiv.org/abs/2304.03442*)).

W biznesie najbardziej popularne są boty do obsługi klienta. Pomagają firmom zmniejszać koszty, a jednocześnie poprawiają doświadczenia użytkowników, ponieważ odpowiadają na pytania szybciej niż ludzie. Sztuczna inteligencja może także pełnić funkcję asystentów produktowych, prowadząc klientów przez trudne i złożone procesy, takie jak składanie wniosków ubezpieczeniowych, rozliczanie podatków czy przeglądanie polityk korporacyjnych.

Sukces modelu ChatGPT zapoczątkował rozwój wielu botów konwersacyjnych opartych na informacjach tekstowych. Jednak tekst to nie jedyny sposób interakcji z agentami konwersacyjnymi.

Asystenci głosowi, tacy jak Google Assistant, Siri czy Alexa, funkcjonują od lat[15]. Boty konwersacyjne 3D stają się coraz bardziej powszechne w grach oraz zyskują popularność w handlu detalicznym i marketingu.

Przykładami wykorzystania postaci 3D wspomaganych przez AI są inteligentne postacie niezależne (obejrzyj prezentacje platform Inworld (*https://oreil.ly/yn-DN*) i Convai (*https://oreil.ly/zAHwz*) firmy NVIDIA)[16]. W wielu grach stanowią one kluczowe elementy, wspierające rozwój fabuły. W przypadku gier tradycyjnych postacie niezależne są zazwyczaj sterowane skryptami i mogą wykonywać proste czynności, a ich dialogi są ograniczone. Dzięki AI postacie niezależne stają się jednak znacznie bardziej inteligentne. Może to znacząco wpłynąć na dynamikę gier takich jak *The Sims* czy *Skyrim*, a także umożliwić tworzenie nowych, bardziej innowacyjnych, które wcześniej były niemożliwe do zrealizowania.

Agregacja informacji

Wiele osób uważa, że sukces zależy od zdolności do filtrowania i przyswajania użytecznych informacji. Jednakże nadążanie za mailami, wiadomościami na Slacku i informacjami ze świata czasami bywa przytłaczające. Na szczęście z pomocą przyszła sztuczna inteligencja. AI udowodniła, że potrafi agregować informacje i je streszczać. Zgodnie z badaniem Generative AI Snapshot Research firmy Salesforce z 2023 roku (*https://oreil.ly/74soT*) 74% użytkowników sztucznej inteligencji wykorzystuje ją do upraszczania złożonych idei i podsumowywania informacji.

Konsumenci uzyskali dostęp do wielu aplikacji, które potrafią przetwarzać dokumenty, takie jak umowy, oświadczenia, artykuły. Dzięki temu uzyskują informacje w sposób konwersacyjny. Ten przypadek użycia nazywany jest również *talk-to-your-docs*. AI może pomóc w streszczaniu stron internetowych, dokumentów naukowych oraz tworzeniu raportów dotyczących wybranych przez użytkownika tematów. Podczas pisania tej książki sztuczna inteligencja okazała się pomocna w streszczaniu i porównywaniu artykułów.

Agregacja informacji i ich destylacja są kluczowe dla funkcjonowania przedsiębiorstw. Efektywniejsze agregowanie i przetwarzanie informacji może pomóc organizacjom stać się bardziej wydajnymi, ponieważ wspiera średni szczebel zarządzania. Gdy firma Instacart (*https://oreil.ly/Qq5-g*) uruchomiła wewnętrzny rynek promptów, odkryto, że jednym z najczęściej wybieranych szablonów jest „Fast Breakdown" (szybkie podsumowanie). Dzięki niemu sztuczna inteligencja streszcza notatki ze spotkań, maili i rozmów na Slacku, uwzględniając fakty, kwestie do dyskusji i zadania do wykonania. Wyniki mogą następnie zostać automatycznie wprowadzone do narzędzia służącego do śledzenia projektów i przypisane do odpowiednich właścicieli.

[15] To zaskakujące, że firmy Apple i Amazon zwlekają z wdrożeniem generatywnej sztucznej inteligencji w produktach Siri i Alexa. Mój przyjaciel sugeruje, że może to wynikać z wyższych standardów jakości i zgodności, które te firmy muszą spełniać, a także z faktu, że opracowanie interfejsów głosowych jest bardziej czasochłonne niż tekstowych.

[16] Uwaga: pełnię rolę doradcy w ramach platformy Convai.

AI może pomóc w ujawnianiu kluczowych informacji na temat potencjalnych klientów oraz przeprowadzać analizy firm konkurencyjnych. Im więcej informacji zbierasz, tym ważniejsze staje się ich uporządkowanie. Agregacja informacji idzie w parze z organizowaniem danych.

Organizowanie danych

Jednym z niepodważalnych faktów dotyczących przyszłości jest ten, że będziemy generować coraz większe ilości danych. Użytkownicy smartfonów będą nadal tworzyć zdjęcia i filmy. Firmy będą wciąż rejestrować szczegóły dotyczące swoich produktów, pracowników i klientów. Każdego roku powstają miliardy umów. Zdjęcia, filmy, dzienniki i pliki PDF to dane nieustrukturyzowane lub częściowo ustrukturyzowane. Kluczowym zagadnieniem staje się zorganizowanie tych danych w sposób umożliwiający ich efektywne przeszukiwanie w przyszłości.

Sztuczna inteligencja może w tym pomóc. AI jest w stanie automatycznie generować opisy tekstowe dla obrazów i filmów lub pomagać w dopasowywaniu zapytań tekstowych do odpowiadających im treści wizualnych. Usługi takie jak Google Photos już teraz wykorzystują AI do wyszukiwania obrazów pasujących do zapytań[17]. Wyszukiwarka obrazów Google idzie o krok dalej — jeśli nie istnieje obraz odpowiadający potrzebom użytkownika, może go wygenerować.

Sztuczna inteligencja doskonale sprawdza się w analizie danych. Może tworzyć programy generujące wizualizacje danych, identyfikować anomalie oraz opracowywać prognozy, takie jak przewidywanie przychodów[18].

Przedsiębiorstwa mogą wykorzystywać AI do ekstrakcji ustrukturyzowanych informacji z danych nieustrukturyzowanych, co pozwala na lepszą organizację i wyszukiwanie danych. Proste przypadki użycia obejmują automatyczne wyodrębnianie informacji z kart kredytowych, praw jazdy, paragonów, biletów, danych kontaktowych istniejących w podpisach maili itp. Bardziej złożone przypadki dotyczą ekstrakcji danych z umów, raportów, wykresów i innych dokumentów. Szacuje się, że branża inteligentnego przetwarzania danych (ang. *Intelligent Data Processing*, w skrócie IDP) osiągnie wartość 12,81 miliarda dolarów do 2030 roku (*https://oreil.ly/vnDNK*), wzrastając w tempie 32,9% rocznie.

Automatyzacja przepływu danych

Sztuczna inteligencja powinna zautomatyzować jak najwięcej procesów. W przypadku użytkowników końcowych automatyzacja może pomóc w wykonywaniu nudnych, codziennych zadań, takich jak rezerwowanie miejsc w restauracji, składanie wniosków o zwrot pieniędzy, planowanie podróży czy wypełnianie formularzy.

[17] Obecnie przechowuję ponad 40 tysięcy zdjęć i filmów w usłudze Google Photos. Bez wykorzystania sztucznej inteligencji przeszukiwanie tej kolekcji w celu znalezienia określonych zdjęć byłoby niezwykle trudne, a może nawet niemożliwe.

[18] Osobiście uważam również, że sztuczna inteligencja świetnie radzi sobie z objaśnianiem danych i wykresów. Gdy natrafiam na skomplikowany wykres zawierający zbyt wiele informacji, proszę model ChatGPT, aby pomógł mi go przeanalizować i uprościć.

Biorąc pod uwagę przedsiębiorstwa, AI może automatyzować powtarzalne zadania, takie jak zarządzanie leadami, wystawianie faktur, rozliczenia, obsługa próśb klientów, wprowadzanie danych i wiele innych. Szczególnie interesujące zastosowanie to wykorzystanie modeli AI do syntezy danych, które następnie mogą posłużyć do ich doskonalenia. Można użyć sztucznej inteligencji do oznaczania danych, angażując ludzi jedynie do ulepszania etykiet. Temat syntezy danych zostanie przeanalizowany w rozdziale 8.

Aby wykonać wiele zadań, niezbędny jest dostęp do zewnętrznych narzędzi. Na przykład by zarezerwować stolik w restauracji, aplikacja może potrzebować uprawnień do uruchomienia wyszukiwarki w celu znalezienia numeru telefonu, korzystania z telefonu do wykonania połączenia oraz dodania wydarzenia do kalendarza. Systemy AI, które potrafią planować i korzystać z narzędzi, są nazywane *agentami*. Zainteresowanie agentami AI graniczy z obsesją, ale nie jest całkowicie nieuzasadnione. Agenty sztucznej inteligencji mają potencjał, który pozwala znacząco zwiększać produktywność i generować ogromną wartość ekonomiczną. Agenty zostaną przeanalizowane w rozdziale 6.

Badanie różnych zastosowań sztucznej inteligencji to niezwykle fascynujące zajęcie. Jeden z moich ulubionych tematów do rozmyślań to różnorodne aplikacje, które mogłabym utworzyć. Nie wszystkie aplikacje powinny być jednak budowane. W następnym podrozdziale zostanie przeanalizowane, co należy wziąć pod uwagę przed utworzeniem aplikacji opartej na sztucznej inteligencji.

Planowanie aplikacji AI

Biorąc pod uwagę niemal nieograniczony potencjał sztucznej inteligencji, kuszące staje się, by od razu zabrać się za tworzenie aplikacji. Jeśli po prostu chcesz się czegoś nauczyć i dobrze bawić, nie zwlekaj — zacznij działać. Tworzenie to jeden z najlepszych sposobów uczenia się. W początkowych latach rozwoju modeli podstawowych wielu liderów AI sugerowało swoim zespołom eksperymentowanie z aplikacjami sztucznej inteligencji w celu podniesienia kompetencji.

Jeśli jednak robisz to zawodowo, warto się zatrzymać na chwilę i zastanowić, dlaczego chcesz zbudować daną aplikację, a także jak należy do tego podejść. Łatwo jest stworzyć efektowną wersję demonstracyjną z wykorzystaniem modeli podstawowych. Znacznie trudniej opracować produkt, który będzie przynosił zyski.

Ocena przypadków użycia

Przede wszystkim warto się zastanowić, dlaczego chcesz stworzyć daną aplikację. Podobnie jak ma to miejsce w przypadku wielu decyzji biznesowych również tworzenie aplikacji AI często wynika z analizy ryzyka i możliwości. Oto przykłady różnych poziomów ryzyka, uporządkowanych od najwyższego do najniższego:

1. *Jeśli tego nie zrobisz, konkurenci, którzy wykorzystują AI, mogą Cię wyprzedzić.* W przypadku gdy sztuczna inteligencja stanowi poważne zagrożenie dla Twojego biznesu, włączenie jej do działalności powinno być priorytetem. W badaniu Gartnera z 2023 roku (*https://oreil.ly/gqi3d*) 7% respondentów wskazało ciągłość działalności jako powód implementacji AI.

Jest to szczególnie ważne w branżach zajmujących się zarządzaniem dokumentami i agregowaniem danych, takich jak analizy finansowe, ubezpieczenia czy przetwarzanie danych. AI ma również duże znaczenie w pracy twórczej, jak reklama, projektowanie stron internetowych czy tworzenie obrazów. By sprawdzić, jak różne branże są narażone na wpływ AI, można się zapoznać z badaniem *GPTs are GPTs* firmy OpenAI z 2023 roku (Eloundou i in., 2023 (*https://arxiv.org/abs/2303.10130*)).

2. *Jeśli tego nie zrobisz, zmarnujesz okazję, by zwiększyć zyski i produktywność.* Większość firm wdraża AI ze względu na możliwości, jakie ona niesie. Sztuczna inteligencja może wspierać prawie każdą dziedzinę działalności. Potrafi na przykład obniżyć koszty pozyskiwania klientów, generując bardziej skuteczne teksty reklamowe, opisy produktów i materiały promujące. Może też poprawić poziom lojalności klientów, ulepszając obsługę oraz personalizując doświadczenia użytkowników. Pomaga również w generowaniu leadów sprzedażowych, badaniach rynkowych, analizie konkurencji i komunikacji wewnętrznej.
3. *Możesz nie być jeszcze pewny, w jaki sposób AI wpisuje się w Twój model biznesowy, ale nie chcesz zostać w tyle.* Choć firma nie powinna bezrefleksyjnie podążać za każdą modą, wiele z tych, które opóźniły decyzję o wdrożeniu innowacji, zniknęło z rynku (przykładami są Kodak, Blockbuster czy BlackBerry). Zainwestowanie w zrozumienie wpływu rewolucyjnej technologii na biznes to dobra decyzja, o ile firma może sobie na to pozwolić. W większych organizacjach może to być część działań badawczo-rozwojowych[19].

Po znalezieniu solidnego uzasadnienia dla danego przypadku użycia warto rozważyć, czy musisz budować aplikację we własnym zakresie. Jeśli AI stanowi zagrożenie egzystencjalne dla Twojego biznesu, rozważ realizację projektu wewnętrznie, zamiast zlecać to firmom konkurencyjnym.

Jeśli natomiast wykorzystujesz sztuczną inteligencję do zwiększenia zysków i produktywności, możesz mieć do dyspozycji wiele rozwiązań gotowych do wdrożenia, które pozwolą zaoszczędzić czas i pieniądze, przy czym oferują lepszą wydajność.

Role sztucznej inteligencji i ludzi w tworzeniu aplikacji

Zadania wykonywane przez sztuczną inteligencję w opartym na niej produkcie wpływają na jego rozwój oraz wymagania. Firma Apple (*https://oreil.ly/Dz1HE*) utworzyła doskonały dokument wyjaśniający różne sposoby wykorzystania AI w produkcie. Oto trzy kluczowe punkty, istotne w kontekście bieżącej analizy:

Rola krytyczna lub uzupełniająca

Jeśli aplikacja może funkcjonować bez sztucznej inteligencji, to AI jest tylko jej uzupełnieniem. Na przykład funkcja Face ID nie działałaby bez rozpoznawania twarzy wspomaganego przez AI, podczas gdy usługa Gmail nadal działałaby bez Smart Compose.

Im bardziej sztuczna inteligencja jest krytyczna dla aplikacji, tym dokładniejszy i bardziej niezawodny musi być obszar, w którym się ją wykorzystuje. Użytkownicy są bardziej wyrozumiali wobec błędów, gdy AI nie stanowi kluczowego elementu aplikacji.

[19] Mniejsze startupy mogą być zmuszone do koncentrowania się na rozwoju produktu i nie stać ich na to, by nawet jedna osoba zajmowała się analizowaniem obszaru biznesowego.

Rola reaktywna lub proaktywna

Funkcja reaktywna reaguje na prośby lub określone działania użytkownika, podczas gdy proaktywna jest uruchamiana, kiedy pojawia się odpowiednia okazja. Na przykład chatbot jest funkcją reaktywną, a powiadomienia o ruchu drogowym w Google Maps proaktywne.

Ponieważ funkcje reaktywne są generowane w odpowiedzi na wydarzenia, zazwyczaj (choć nie zawsze) muszą być wykonywane szybko. Z drugiej strony funkcje proaktywne mogą być przetwarzane wcześniej i wyświetlane w odpowiednim momencie, dlatego opóźnienie nie jest tak istotne.

Ponieważ użytkownicy nie proszą o funkcje proaktywne, mogą je uznać za natrętne lub irytujące, jeśli ich jakość jest niska. Prognozy i treści generowane proaktywnie charakteryzują się więc zwykle wyższym poziomem jakości.

Rola dynamiczna lub statyczna

Funkcje dynamiczne są ciągle aktualizowane na podstawie opinii użytkowników, podczas gdy funkcje statyczne są aktualizowane okresowo. Na przykład funkcja Face ID musi być aktualizowana, gdy wygląd twarzy użytkownika się zmienia. Z drugiej strony wykrywanie obiektów w usłudze Google Photos prawdopodobnie jest aktualizowane tylko wtedy, gdy aktualizowana jest cała aplikacja.

W przypadku sztucznej inteligencji użycie funkcji dynamicznych może oznaczać, że każdy użytkownik ma swój model, który jest na bieżąco dostosowywany do jego danych. Mogą istnieć też inne mechanizmy personalizacji, takie jak funkcja pamięci w modelu ChatGPT, która pozwala zapamiętywać preferencje użytkowników. Funkcje statyczne mogą wykorzystywać jeden model przydzielony grupie użytkowników. W takim przypadku funkcje te są modyfikowane tylko wtedy, gdy wspólny model zostaje uaktualniony.

Ważne jest również określenie roli ludzi w aplikacji. Czy sztuczna inteligencja będzie pełnić jedynie drugoplanową funkcję wsparcia, podejmować decyzje bezpośrednio, czy też jedno i drugie? Na przykład w przypadku chatbota obsługi klienta odpowiedzi AI mogą być wykorzystywane na różne sposoby:

- AI pokazuje kilka propozycji odpowiedzi, które agenty mogą wykorzystać, aby szybciej przygotować swoje treści.
- AI odpowiada tylko na proste zapytania, a bardziej złożone przekierowuje do ludzi.
- AI odpowiada na wszystkie zapytania bezpośrednio, bez udziału człowieka.

Zaangażowanie ludzi w procesy decyzyjne AI jest określane terminem *human-in-the-loop* (człowiek w pętli).

W 2023 roku firma Microsoft zaproponowała zestaw reguł stopniowego zwiększania automatyzacji AI w produktach, który nazwała *Crawl-Walk-Run* (*https://oreil.ly/JW4_A*):

1. *Crawl* (czołganie się) oznacza, że udział człowieka jest obowiązkowy.
2. *Walk* (chodzenie) oznacza, że AI może bezpośrednio współdziałać z pracownikami wewnętrznymi.
3. *Run* (bieganie) oznacza zwiększoną automatyzację, potencjalnie obejmującą bezpośrednie interakcje AI z użytkownikami zewnętrznymi.

W miarę wzrostu jakości systemu AI rola człowieka może się zmieniać. Na przykład na początku, gdy wciąż oceniasz możliwości AI, możesz jej używać do generowania sugestii dla pracowników obsługi. Jeśli wskaźnik akceptacji sugestii AI przez konsultantów jest wysoki (na przykład 95% odpowiedzi sugerowanych przez AI w prostych zapytaniach jest bez zmian wykorzystywanych przez pracowników), możesz w takich przypadkach pozwolić klientom na bezpośrednią interakcję ze sztuczną inteligencją.

Zabezpieczenie produktu AI przed konkurencją

Jeśli sprzedajesz aplikacje AI jako samodzielne produkty, ważne jest, aby wziąć pod uwagę ich trwałość przewagi konkurencyjnej. Łatwość rozpoczęcia działalności w tej branży to zarówno błogosławieństwo, jak i przekleństwo. Jeśli coś jest łatwe do utworzenia dla Ciebie, będzie również proste dla Twoich konkurentów. Jak więc możesz zabezpieczyć swój produkt?

Budowanie aplikacji na podstawie modeli podstawowych oznacza w pewnym sensie dodanie kolejnej warstwy funkcjonalności[20]. Jeśli jednak takie modele rozszerzą swoje możliwości, warstwa, którą dostarczasz, może zostać przez nie wchłonięta, co uczyni Twoją aplikację przestarzałą. Wyobraź sobie utworzenie aplikacji do analizy plików PDF opierającej się na założeniu, że ChatGPT nie radzi sobie dobrze z takimi zadaniami lub nie obsługuje ich na dużą skalę. Jeśli to założenie przestanie być prawdziwe, Twój poziom konkurencyjności się obniży. Niemniej jednak taka aplikacja może mieć sens, jeśli zostanie oparta na modelach o otwartym oprogramowaniu i skierowana do użytkowników chcących uruchamiać modele lokalnie.

Partner wiodący w jednym z głównych funduszy venture capital powiedział mi, że miał do czynienia z wieloma startupami, których pełne produkty mogłyby być zaledwie jedną z funkcji usług Google Docs lub Microsoft Office. Jeśli takie produkty jednak odniosą sukces, co powstrzyma firmy Google lub Microsoft przed przydzieleniem trzech inżynierów do odtworzenia ich w dwa tygodnie?

Sztuczna inteligencja zapewnia trzy główne przewagi konkurencyjne: technologię, dane i dystrybucję, czyli zdolność do dotarcia z produktem do użytkowników. W przypadku modeli podstawowych kluczowe technologie większości firm będą zbliżone do siebie. Przewagę w dystrybucji prawdopodobnie będą miały duże firmy.

Przewaga związana z danymi stanowi bardziej złożone zagadnienie. Duże firmy mają zazwyczaj większe zasoby danych. Startupy, które jako pierwsze wejdą na rynek i zdobędą wystarczającą liczbę użytkowników, mogą jednak wykorzystać te dane do nieustannego doskonalenia swoich produktów. Nawet jeśli dane użytkowników nie mogą być bezpośrednio wykorzystywane do trenowania modeli, informacje o sposobie korzystania z produktu dostarczają bezcennych wskazówek na temat zachowań użytkowników i niedoskonałości aplikacji. Można je następnie wykorzystać do ukierunkowania procesu gromadzenia danych i trenowania modeli[21].

[20] W początkowym okresie rozwoju generatywnej sztucznej inteligencji krążył żart, że startupy AI to jedynie nakładki na OpenAI lub Claude.

[21] Podczas pracy nad tą książką niemal w każdym moim kontakcie ze startupem AI poruszano temat „samonapędzającego się mechanizmu opartego na danych".

Wiele firm z ugruntowaną pozycją zaczynało od produktów, które mogłyby być jedynie składnikiem większych systemów. Calendly mogło być składnikiem Kalendarza Google. Mailchimp mógł być składnikiem Gmaila. Photoroom mógł być składnikiem Google Photos[22]. Wiele startupów ostatecznie wyprzedziło większych konkurentów dzięki utworzeniu aplikacji, którą ci silniejsi rywale przeoczyli. Może Twoja firma będzie kolejna?

Ustalenie oczekiwań

Po podjęciu decyzji, że będziesz samodzielnie tworzyć niesamowitą aplikację AI, kolejnym krokiem powinno być ustalenie, kiedy rozpoznasz, że osiągnąłeś sukces. Inaczej mówiąc, jakie będą Twoje kryteria sukcesu? Najważniejszym wskaźnikiem jest wpływ na Twoją działalność. Na przykład jeśli chodzi o chatbot do obsługi klientów, wskaźniki biznesowe mogą być następujące:

- Jaki odsetek wiadomości od klientów powinien zautomatyzować chatbot?
- Ile dodatkowych wiadomości powinien przetworzyć chatbot?
- O ile szybciej będziesz w stanie odpowiedzieć dzięki chatbotowi?
- Ile pracy ludzkiej może zaoszczędzić chatbot?

Chatbot może odpowiadać na większą liczbę wiadomości, ale nie oznacza to, że użytkownicy będą bardziej zadowoleni. Ważne więc, by analizować satysfakcję klientów i ogólną opinię o produkcie. W podrozdziale „Informacje zwrotne od użytkowników" dostępnym w dalszej części książki przeanalizowano, jak można zaprojektować system zarządzania opiniami użytkowników.

Aby upewnić się, że produkt nie trafi do klientów, zanim będzie gotowy, warto mieć jasno określone oczekiwania dotyczące jego użyteczności, czyli jak wysoki musi być poziom jego jakości, by był przydatny. Progi użyteczności mogą obejmować następujące grupy wskaźników:

- Wskaźniki jakości, które mierzą jakość odpowiedzi chatbota.
- Wskaźniki opóźnienia, takie jak TTFT (czas do uzyskania pierwszego tokena), TPOT (czas generowania jednego tokena) oraz całkowite opóźnienie. Akceptowalne opóźnienie jest zmienne i zależy od określonego przypadku użycia. Jeśli wszystkie zapytania Twoich klientów są obecnie obsługiwane przez ludzi, a średni czas odpowiedzi wynosi godzinę, to każda szybsza reakcja może być wystarczająca.
- Wskaźniki kosztów: ile kosztuje jedno zapytanie wysłane do modelu.
- Inne wskaźniki, takie jak interpretowalność i sprawiedliwość.

Jeśli nie jesteś pewien, jakie wskaźniki powinieneś wykorzystać, nie martw się. W dalszej części książki zostanie przeanalizowanych wiele takich wskaźników.

[22] Uwaga: jestem inwestorem aplikacji Photoroom.

Zaplanowanie kamieni milowych

Po ustaleniu celów, które można zmierzyć, musisz opracować plan ich realizacji. Sposób osiągnięcia celów zależy od punktu wyjścia. Zbadaj dostępne modele, aby zrozumieć ich możliwości. Im lepszymi modelami gotowymi do użycia dysponujesz, tym mniej pracy będziesz musiał włożyć. Na przykład jeśli celem jest zautomatyzowanie 60% zgłoszeń do obsługi klienta, a model, który chcesz wykorzystać, potrafi już zautomatyzować 30%, pracę można wykonać łatwiej niż w przypadku, gdyby model nie mógł automatyzować żadnych zgłoszeń.

Po przeprowadzeniu analizy Twoje cele mogą się zmienić. Na przykład możesz zdać sobie sprawę, że koszt zasobów potrzebnych do osiągnięcia progu użyteczności aplikacji przewyższa potencjalne zyski, co może skłonić Cię do porzucenia projektu.

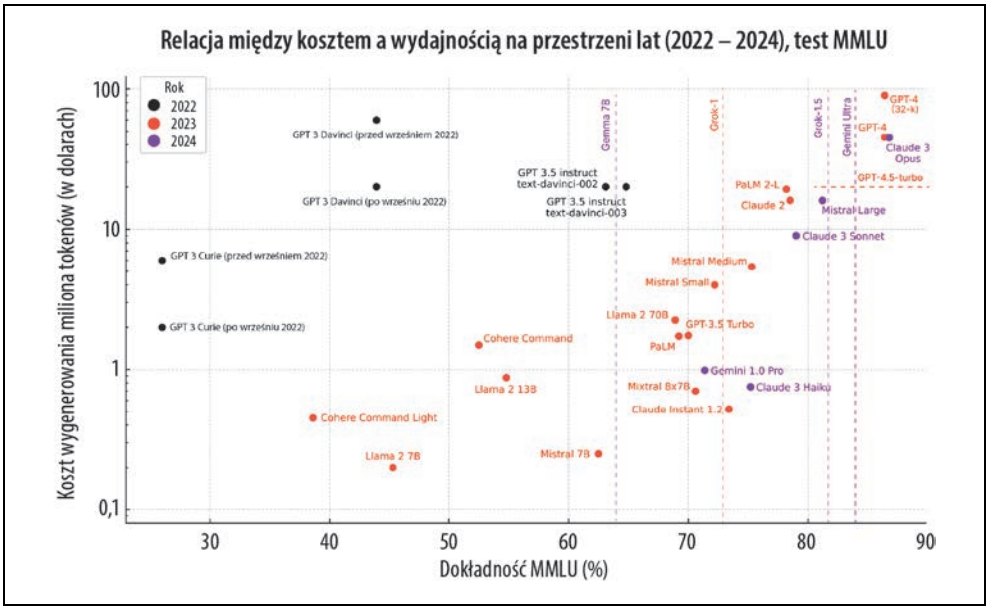
Planowanie produktu AI musi uwzględniać wyzwania związane z jego finalizowaniem. Wczesne sukcesy z modelami podstawowymi mogą być mylące. Ponieważ istniejące możliwości takich modeli są już imponujące, stworzenie prostej wersji demonstracyjnej może nie wymagać wiele czasu. Jednak tak osiągnięty sukces nie oznacza zwycięstwa dla końcowego produktu. Zbudowanie aplikacji w wersji demonstracyjnej może zająć weekend, ale utworzenie gotowego produktu może trwać miesiące, a nawet lata.

W artykule o produkcie UltraChat Ding i in. (2023) (*https://arxiv.org/abs/2305.14233*) napisali, że „droga od 0 do 60 jest łatwa, ale przejście od 60 do 100 staje się wyjątkowo trudne". Firma LinkedIn (2024) (*https://www.linkedin.com/blog/engineering/generative-ai/musings-on-building-a-generative-ai-product*) wyraziła podobną opinię. By osiągnąć 80% zaplanowanego wyniku, poświęcono cały miesiąc pracy. Początkowy sukces sprawił, że zbyt optymistycznie oszacowano czas potrzebny na poprawę produktu. Okazało się, że upłynęły cztery kolejne miesiące, zanim przekroczono 95%. Pracownicy większość czasu spędzili na poprawianiu błędów i radzeniu sobie z halucynacjami. Powolny postęp w osiąganiu każdego kolejnego procentu był demotywujący.

Konserwacja i utrzymanie

Planowanie produktu nie kończy się na osiągnięciu jego celów. Musisz się zastanowić, jak aplikacja będzie ewoluować w czasie i w jaki sposób powinna być utrzymywana. Utrzymanie produktu opartego na sztucznej inteligencji wiąże się z dodatkowym wyzwaniem, jakim jest szybkie tempo zmian w dziedzinie AI. Przez ostatnią dekadę obszar sztucznej inteligencji rozwijał się niezwykle dynamicznie. Prawdopodobnie ta tendencja utrzyma się również w kolejnych latach. Budowanie rozwiązań opartych na modelach podstawowych oznacza zobowiązanie do dostosowywania się do tej szybkiej ewolucji.

Wiele zmian ma pozytywny charakter. Ograniczenia wielu modeli są stopniowo eliminowane. Modele obsługują coraz dłuższe konteksty. Generowane odpowiedzi charakteryzują się wyższą jakością. *Wnioskowanie* modelu, czyli proces wyznaczania odpowiedzi na podstawie dostarczonych danych wejściowych, staje się coraz szybszy i tańszy. Na rysunku 1.11 przedstawiono wykres ewolucji kosztów wnioskowania oraz wydajności aplikacji w latach 2022 – 2024, uzyskany na podstawie popularnego testu porównawczego dla modeli podstawowych *Massive Multitask Language Understanding* (MMLU) (Hendrycks i in., 2020 (*https://arxiv.org/abs/2009.03300*)).



Rysunek 1.11. Koszt wnioskowania AI zmniejsza się szybko z upływem czasu. Ilustracja pochodzi od Katriny Nguyen (https://oreil.ly/UyL8r) (2024)

Nawet te pozytywne zmiany mogą jednak powodować zakłócenia w Twoich procesach. Będziesz więc musiał być stale czujny i przeprowadzać analizę kosztów oraz korzyści dla każdej inwestycji technologicznej. Najlepsza opcja dostępna dzisiaj może już jutro stać się najgorszą. Na przykład możesz się zdecydować na samodzielne zbudowanie modelu, ponieważ będzie Ci się to wydawać tańsze niż korzystanie z usług dostawców. Po trzech miesiącach odkryjesz jednak, że firmy zewnętrzne obniżyły ceny o połowę, co sprawiło, że opcja wybrana przez Ciebie stała się dużo droższa. Możesz też zainwestować w rozwiązanie używane zewnętrznie i dostosować do niego swoją infrastrukturę, aby się wkrótce przekonać, że dostawca zbankrutował po nieudanej próbie pozyskania finansowania.

Niektóre zmiany są łatwiejsze do zaadaptowania. Na przykład dostawcy modeli coraz częściej stosują ujednolicone interfejsy API, co ułatwia ich modyfikację. Jednak każdy model ma pewne specyficzne cechy, a także mocne i słabe strony, dlatego programiści muszą dostosować do niego swoje procesy, zapytania oraz dane. Bez użycia odpowiedniej infrastruktury do wersjonowania i ewaluacji ten proces może być bardzo uciążliwy.

Inne zmiany są trudniejsze do wdrożenia — zwłaszcza te dotyczące regulacji prawnych. Technologie związane ze sztuczną inteligencją są w wielu krajach traktowane jako kwestia bezpieczeństwa narodowego, co oznacza, że zasoby AI — w tym moc obliczeniowa, wiedza i dane — podlegają ścisłym regulacjom. Na przykład po wprowadzeniu europejskiego ogólnego rozporządzenia o ochronie danych (RODO) dostosowanie przedsiębiorstw do przepisów kosztowało około 9 miliardów dolarów (*https://oreil.ly/eDfB8*). W wyniku nowych regulacji ograniczających handel zasobami obliczeniowymi dostępność mocy obliczeniowej może zmieniać się z dnia na dzień (patrz amerykański dekret wykonawczy z października 2023 roku (*https://oreil.ly/eYTmr*)).

Jeśli Twój dostawca GPU zostanie nagle objęty zakazem sprzedaży układów cyfrowych do Twojego kraju, znajdziesz się w poważnych tarapatach.

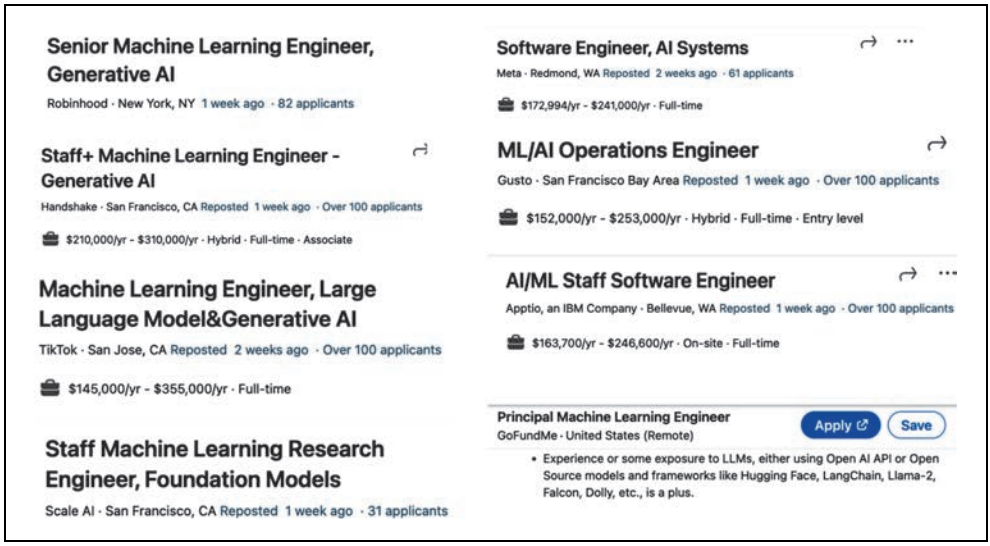
Niektóre zmiany mogą wręcz okazać się katastrofalne. Na przykład regulacje dotyczące własności intelektualnej (ang. *Intellectual Property*, w skrócie IP) i wykorzystania AI wciąż ewoluują. Jeśli produkt opiera się na modelu wytrenowanym na danych uzyskanych od innych osób, to czy możesz mieć pewność, że własność intelektualna Twojej aplikacji zawsze będzie należała do Ciebie? Wiele firm, takich jak studia gier, unika korzystania ze sztucznej inteligencji w obawie przed późniejszą utratą praw do własności intelektualnej.

Skoro jednak podjąłeś decyzję o utworzeniu aplikacji AI, przyjrzyjmy się teraz stosowi technologicznemu potrzebnemu w takim projekcie.

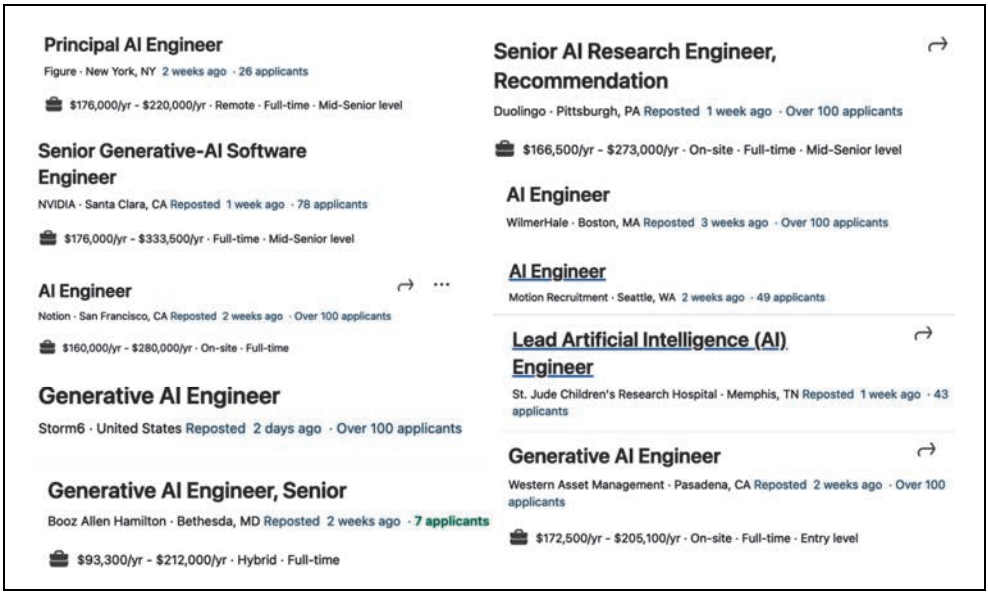
Stos technologiczny inżynierii AI

Szybki rozwój inżynierii AI doprowadził do powstania ogromnego szumu informacyjnego oraz syndromu FOMO (ang. *fear of missing out* — obawa przed wypadnięciem z obiegu). Każdego dnia pojawiają się nowe narzędzia, techniki, modele i aplikacje, co rzeczywiście może być przytłaczające. Zamiast starać się nadążyć za tym nieustannie zmieniającym się krajobrazem, warto się skoncentrować na fundamentalnych elementach inżynierii AI.

Aby zrozumieć inżynierię AI, należy zauważyć, że wywodzi się ona z inżynierii uczenia maszynowego. Gdy firma zaczyna eksperymentować z modelami podstawowymi, naturalnym krokiem jest powierzenie tego zadania istniejącemu zespołowi ML. Niektóre organizacje traktują inżynierię AI jako tożsamą z inżynierią ML, co ilustruje rysunek 1.12. Inne natomiast tworzą odrębne stanowiska dla inżynierów AI, jak pokazano na rysunku 1.13.



Rysunek 1.12. Wiele firm traktuje inżynierię AI i ML jako jedną kategorię, co pokazują nagłówki ofert pracy na portalu LinkedIn (zrzut ekranu z 17 grudnia 2023 roku)



Rysunek 1.13. Niektóre firmy udostępniają osobne opisy stanowisk dla inżynierii AI, jak pokazano w nagłówkach ofert pracy na portalu LinkedIn (zrzut ekranu z 17 grudnia 2023 roku)

Bez względu na to, jak firmy definiują role inżynierów AI i ML, ich obowiązki w dużej mierze się pokrywają. Inżynierowie ML mogą poszerzyć swoje kompetencje o inżynierię AI, co zwiększa ich możliwości zawodowe. Istnieją jednak inżynierowie AI bez wcześniejszego doświadczenia w uczeniu maszynowym.

Abyś mógł lepiej zrozumieć inżynierię AI i różnice względem tradycyjnej inżynierii ML, w następnym punkcie przedstawiono różne warstwy procesu budowy aplikacji sztucznej inteligencji oraz rolę, jaką każda z nich odgrywa w obu dziedzinach.

Trzy warstwy stosu AI

Każda aplikacja AI opiera się na trzech warstwach: projektowaniu aplikacji, projektowaniu modelu i infrastrukturze. Proces tworzenia aplikacji AI zaczyna się zwykle od najwyższej warstwy, a w razie potrzeby przechodzi do niższych poziomów:

Projektowanie aplikacji

Dzięki łatwemu dostępowi do modeli każdy może je wykorzystywać do budowy aplikacji. Jest to warstwa, która w ostatnich dwóch latach rozwijała się najszybciej i nadal ewoluuje. Projektowanie aplikacji obejmuje dostarczanie modelowi odpowiednich promptów oraz kontekstu. Kluczowe znaczenie ma tutaj dokładna ewaluacja. Dobrze działające aplikacje wymagają także poprawnie zaprojektowanych interfejsów.

Projektowanie modelu

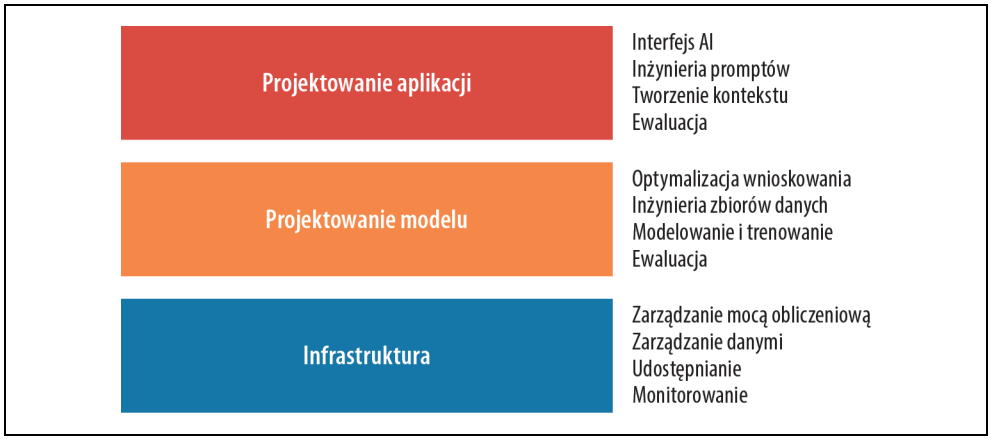
Ta warstwa obejmuje narzędzia do budowy modeli, w tym narzędzia do modelowania, trenowania, dostrajania oraz optymalizacji wnioskowania. Ponieważ dane odgrywają kluczową

rolę w rozwoju modeli, ta warstwa obejmuje również inżynierię zbiorów danych. Podobnie jak w przypadku aplikacji wymagana jest tu dokładna ewaluacja.

Infrastruktura

Najniższą warstwą stosu jest infrastruktura, obejmująca narzędzia do udostępniania modeli, zarządzania danymi i mocą obliczeniową, a także monitorowania systemu.

Te trzy warstwy oraz przykłady zakresów odpowiedzialności dla każdej z nich zostały przedstawione na rysunku 1.14.

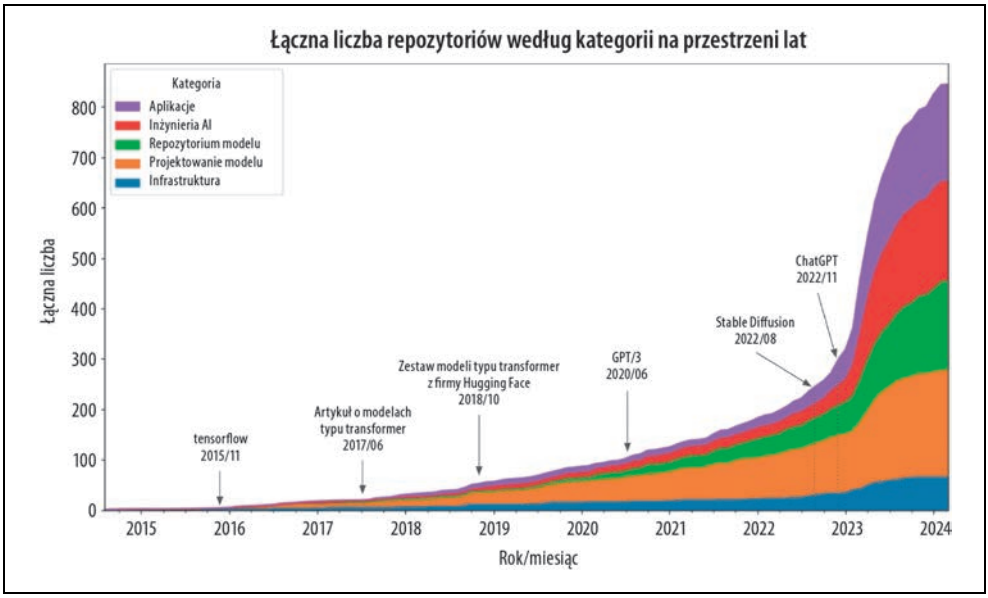


Rysunek 1.14. Trzy warstwy stosu technologicznego AI

Aby lepiej zrozumieć rozwój ekosystemu modeli podstawowych, w marcu 2024 roku przeanalizowałam GitHuba pod kątem repozytoriów związanych ze sztuczną inteligencją, które uzyskały co najmniej 500 gwiazdek. Ponieważ GitHub jest powszechnie wykorzystywaną platformą, uznałam te dane za dobry wskaźnik dynamiki branży. W swoim badaniu uwzględniłam również repozytoria dotyczące aplikacji i modeli, które odpowiadają warstwom projektowania aplikacji i modeli. Łącznie zidentyfikowałam 920 repozytoriów. Rysunek 1.15 przedstawia miesięczny przyrost ich liczby w poszczególnych kategoriach.

Dane pokazują znaczny wzrost liczby narzędzi AI w 2023 roku po premierze modeli Stable Diffusion i ChatGPT. Największy przyrost zanotowały wtedy kategorie aplikacji oraz ich projektowania. Warstwa infrastruktury również się rozwinęła, ale jej wzrost był znacznie mniej dynamiczny w porównaniu do pozostałych. Jest to zgodne z oczekiwaniami — mimo że modele i aplikacje się zmieniają, kluczowe aspekty infrastruktury, takie jak zarządzanie zasobami, udostępnianie czy monitorowanie, pozostają stabilne.

Prowadzi to do kolejnej istotnej obserwacji. Pomimo ogromnego entuzjazmu i innowacyjności w obszarze modeli podstawowych wiele zasad dotyczących budowy aplikacji AI pozostaje niezmiennych. W zastosowaniach biznesowych sztuczna inteligencja nadal musi odpowiadać na konkretne potrzeby, co oznacza konieczność przełożenia wskaźników biznesowych na wskaźniki ML i odwrotnie. Eksperymentowanie wciąż odgrywa kluczową rolę — w tradycyjnej inżynierii ML oznacza to testowanie różnych hiperparametrów, natomiast w przypadku modeli podstawowych



Rysunek 1.15. Łączna liczba repozytoriów w poszczególnych kategoriach na przestrzeni lat

obejmuje dobór modeli, promptów, algorytmów wyszukiwania, zmiennych próbkowania i innych czynników (więcej o tym w rozdziale 2.). Nie zmienia się także dążenie do optymalizacji modeli pod względem wydajności i kosztów. Równie istotne pozostaje wdrażanie pętli sprzężenia zwrotnego, umożliwiającej iteracyjne doskonalenie aplikacji na podstawie danych ze środowiska produkcyjnego.

Oznacza to, że wiedza i doświadczenie zdobyte w ciągu ostatnich 10 lat przez inżynierów ML są nadal cenne i mają zastosowanie. Ułatwia to rozpoczęcie pracy nad aplikacjami AI. Jednocześnie na tych trwałych zasadach opiera się także wiele innowacji unikatowych dla inżynierii sztucznej inteligencji, które zostaną przeanalizowane w tej książce.

Inżynieria AI a inżynieria ML

Chociaż podstawowe zasady dotyczące wdrażania aplikacji AI pozostają takie same, kluczowe jest także zrozumienie, jak ewoluowały procesy. Jest to istotne dla zespołów, które chcą dostosować swoje obecne platformy do nowych zastosowań AI, a także dla deweloperów, którzy zastanawiają się, jakie umiejętności warto rozwijać, by pozostać konkurencyjnym na zmieniającym się rynku.

Ogólnie rzecz ujmując, obecne budowanie aplikacji z wykorzystaniem modeli podstawowych różni się od tradycyjnej inżynierii ML w trzech kluczowych kwestiach:

1. Jeśli nie masz dostępu do modeli podstawowych, musisz trenować własne modele, które zostaną wykorzystane w aplikacji. W przypadku inżynierii AI korzystasz z modelu, który został wytrenowany przez kogoś innego. Oznacza to, że inżynieria AI koncentruje się mniej na modelowaniu i trenowaniu, a bardziej na adaptacji modelu.

2. Inżynieria AI zajmuje się modelami, które są bardziej złożone, wymagają większych zasobów obliczeniowych i generują wyższe opóźnienia w porównaniu do tradycyjnej inżynierii ML. Wiąże się to ze zwróceniem większej uwagi na optymalizację treningu i wnioskowania. Konieczność korzystania z bardziej zaawansowanych modeli oznacza, że wiele firm potrzebuje obecnie znacznej liczby procesorów GPU i rozbudowanych klastrów obliczeniowych, co stwarza popyt na odpowiednich specjalistów[23].
3. Inżynieria AI wykorzystuje modele, które mogą generować wyniki otwarte. Dzięki temu modele są bardziej uniwersalne i mogą być wykorzystywane do różnych zadań, lecz są trudniejsze do ewaluacji. Powoduje to, że ewaluacja modelu staje się istotnym wyzwaniem w inżynierii AI.

Podsumowując, inżynieria AI różni się od inżynierii ML tym, że mniej koncentruje się na tworzeniu modeli, a bardziej na ich adaptacji i ewaluacji. W tym rozdziale wspomniałam już o adaptacji modelu, więc zanim przejdziemy dalej, chciałabym się tylko upewnić, że wszyscy rozumieją, czym ona jest. Ogólnie rzecz ujmując, techniki adaptacji modeli można podzielić na dwie kategorie w zależności od tego, czy wymagają one zmiany wag modelu.

Rozwiązania oparte na promptach (a więc także na inżynierii promptów) pozwalają na adaptację modelu bez zmiany jego wag. W takim przypadku zamiast modyfikować sam model, dostosowujesz go poprzez odpowiednie instrukcje i kontekst. Zaletą inżynierii promptów jest prostota i mniejsze wymagania związane z ilością danych. Wiele udanych aplikacji zostało zbudowanych tylko za pomocą inżynierii promptów. Łatwość tej metody pozwala na eksperymentowanie z wieloma modelami, co zwiększa szansę na znalezienie takiego, który niespodziewanie dobrze sprawdzi się w aplikacji. W przypadku bardziej złożonych zadań lub aplikacji o wysokich wymaganiach wydajnościowych inżynieria promptów może jednak nie wystarczyć.

Dostrajanie wymaga z kolei zmiany wag modelu. Adaptacja modelu polega na jego modyfikacji. Ogólnie rzecz ujmując, techniki dostrajania są bardziej skomplikowane i wymagają większej ilości danych, ale mogą znacznie poprawić jakość, a także zmniejszyć opóźnienia i koszty modelu. Niektóre wymagania są w ogóle niemożliwe do zrealizowania bez zmiany wag modelu — chodzi na przykład o dostosowanie modelu do nowego zadania, które nie zostało mu zaprezentowane w czasie treningu.

Teraz przyjrzyjmy się szczegółowo procesom projektowania aplikacji i modeli, aby się dowiedzieć, jak zmieniły się one w przypadku inżynierii AI. Zaczniemy od zadań, które są bardziej znane inżynierom ML. W najbliższym punkcie przedstawimy ogólny przegląd różnych procesów mających miejsce podczas projektowania aplikacji AI, natomiast szczegóły zostaną przeanalizowane w dalszej części książki.

Projektowanie modelu

Projektowanie modelu jest warstwą najczęściej kojarzoną z tradycyjną inżynierią ML. Obejmuje trzy główne zadania: modelowanie i trenowanie, inżynierię zbioru danych oraz optymalizację

[23] Jak powiedział mi szef działu AI w firmie z listy Fortune 500 — jego zespół wie, jak obsłużyć 10 GPU, ale nie potrafi już sobie poradzić z tysiącem takich urządzeń.

wnioskowania. Ewaluacja jest również wymagana, jednakże większość osób zetknie się z nią po raz pierwszy w warstwie projektowania aplikacji. Przeanalizuję ją więc w dalszej części rozdziału.

Modelowanie i trenowanie. *Modelowanie i trenowanie* odnosi się do procesu tworzenia architektury modelu, jego trenowania i dostrajania. Przykłady dostępnych narzędzi w tej kategorii to TensorFlow firmy Google, Transformers firmy Hugging Face oraz PyTorch firmy Meta.

Projektowanie modeli ML wymaga specjalistycznej wiedzy z zakresu uczenia maszynowego, a więc znajomości różnych typów algorytmów ML (takich jak grupowanie, regresja logistyczna, drzewa decyzyjne czy filtrowanie zespołowe) oraz architektur sieci neuronowych (takich jak sieci ze sprzężeniem wyprzedzającym, rekurencyjne, konwolucyjne oraz transformer). Niezbędne jest także zrozumienie, jak model się uczy, a także pojęć takich jak spadek wzdłuż gradientu, funkcja straty, regularyzacja itp.

Dzięki dostępności modeli podstawowych wiedza z zakresu ML nie jest już niezbędna do tworzenia aplikacji sztucznej inteligencji. Poznałam wielu wspaniałych i odnoszących sukcesy twórców aplikacji AI, którzy w ogóle nie interesują się spadkiem wzdłuż gradientu. Wiedza z zakresu ML jest jednak nadal bardzo cenna, ponieważ poszerza zestaw narzędzi, które możesz wykorzystać, a także pomaga w rozwiązywaniu problemów, gdy model nie działa zgodnie z oczekiwaniami.

Czym się różnią od siebie trenowanie, wstępne trenowanie, dostrajanie i post-trening?

Trenowanie zawsze wiąże się ze zmianą wag modelu, ale nie wszystkie zmiany wag modelu są traktowane jako trenowanie. Na przykład kwantyzacja, czyli proces zmniejszania dokładności wag modelu, z technicznego punktu widzenia zmienia wartości wag modelu, ale nie jest uznawana za trenowanie.

Termin „trenowanie" często bywa używany razem z „wstępnym trenowaniem", „dostrajaniem" i „post-treningiem", które odnoszą się do różnych etapów treningu:

Wstępne trenowanie

> *Wstępne trenowanie* odnosi się do trenowania modelu od podstaw — wagi modelu są inicjalizowane losowo. W przypadku dużych modeli językowych (LLM) wstępne trenowanie polega często na trenowaniu modelu w celu uzupełniania tekstu. Wstępne trenowanie najczęściej jest najbardziej wymagające pod względem zasobów spośród wszystkich etapów trenowania. Na przykład w przypadku modelu InstructGPT wstępne trenowanie zużywa do 98% całkowitych zasobów obliczeniowych i danych (*https://oreil.ly/G3LUh*). Proces ten jest również czasochłonny, a nawet drobna pomyłka w trakcie może prowadzić do dużych strat finansowych i poważnych opóźnień w projekcie. Z uwagi na duże zużycie zasobów wstępne trenowanie stało się umiejętnością, którą opanowało tylko wąskie grono specjalistów. Osoby posiadające doświadczenie w tej dziedzinie są bardzo poszukiwane[24].

[24] Są im także oferowane niesamowite pakiety wynagrodzeń (*https://oreil.ly/AhANP*).

Dostrajanie

Dostrajanie oznacza kontynuację trenowania wcześniej wytrenowanego modelu — wagi modelu pochodzą z poprzedniego procesu treningowego. Ponieważ model posiada już wiedzę zdobytą w trakcie wstępnego treningu, dostrajanie zazwyczaj wymaga mniejszych zasobów (takich jak dane i moc obliczeniowa).

Post-trening

Wiele osób używa terminu „post-trening" (ang. *post-training*) do opisania procesu trenowania modelu występującego po fazie wstępnego trenowania. Z konceptualnego punktu widzenia terminy „post-trening" i „dostrajanie" dotyczą tego samego procesu i mogą być stosowane zamiennie. Jednak czasami stosuje się je w różnych kontekstach, aby podkreślić odmienne cele. Zwykle mówimy o post-treningu, gdy wykonują je twórcy modelu. Na przykład firma OpenAI może przeprowadzić post-trening, aby poprawić zdolność modelu do lepszego wykonywania instrukcji przed jego publicznym udostępnieniem. Dostrajanie występuje wówczas, gdy proces ten przeprowadzają twórcy aplikacji. Na przykład możesz dostroić model firmy OpenAI (który i tak mógł mieć już za sobą etap działań post-treningowych) do własnych potrzeb.

Wstępne trenowanie i post-trening to dwa końce spektrum[25]. Wykorzystywane w nich procesy oraz narzędzia są do siebie bardzo podobne. Szczegóły dotyczące różnic zostaną przeanalizowane w rozdziałach 2. i 7.

Niektórzy używają terminu „trenowanie" w odniesieniu do inżynierii promptów, co jednak nie jest precyzyjne. Przeczytałam pewien artykuł dostępny na portalu Business Insider (*https://oreil.ly/0VqmX*), w którym autorka stwierdza, że trenowała ChatGPT, aby odwzorowywał jej młodsze „ja". Zostało to zrealizowane poprzez wprowadzenie do modelu zapisków z dzieciństwa. Potocznie użycie słowa „*trening*" przez autorkę jest poprawne, ponieważ uczyła ona model wykonywania określonego zadania. Jeśli jednak uczysz model tego, co ma robić, używając przy tym odpowiedniego kontekstu, z technicznego punktu widzenia wykonujesz inżynierię promptów. Spotkałam się też z przypadkami, w których używa się terminu „dostrajanie", podczas gdy tak naprawdę chodzi o inżynierię promptów.

Inżynieria zbiorów danych. *Inżynieria zbiorów danych* polega na nadzorowaniu, generowaniu i oznaczaniu danych niezbędnych do trenowania i dostrajania modeli AI.

W tradycyjnej inżynierii ML większość przypadków użycia ma charakter zamknięty — wynik wygenerowany przez model może przyjąć jedynie zdefiniowane wartości. Przykładem może być klasyfikacja spamu, w przypadku której są dostępne tylko dwie opcje: „spam" i „brak spamu". Modele podstawowe są natomiast otwarte. Oznaczanie zapytań otwartych jest znacznie trudniejsze niż zamkniętych — łatwiej można sprawdzić, czy dany mail to spam, niż napisać wypracowanie. Oznaczanie danych stanowi z tego powodu większe wyzwanie w inżynierii AI.

[25] Jeśli uważasz, że terminy „wstępne trenowanie" i „post-trening" są mało kreatywne, nie jesteś w tym odosobniony. Społeczność zajmująca się sztuczną inteligencją świetnie sobie radzi w wielu obszarach, ale nadawanie nazw nie jest jej najmocniejszą strona. Już wcześniej wspomniałam, że termin „duże modele językowe" nie jest do końca precyzyjny, ponieważ słowo „duży" jest niejednoznaczne. Szczerze mówiąc, naprawdę życzyłabym sobie, żeby przestano już publikować artykuły zatytułowane *X to wszystko, czego potrzebujesz*.

Kolejną różnicą jest to, że tradycyjna inżynieria ML opiera się głównie na danych tabelarycznych, podczas gdy modele podstawowe pracują z danymi nieustrukturyzowanymi. W inżynierii AI przetwarzanie danych polega bardziej na usuwaniu duplikatów, tokenizacji, wyszukiwaniu kontekstu oraz kontroli jakości, w tym usuwaniu informacji wrażliwych i danych toksycznych. Projektowanie zbiorów danych zostało szczegółowo omówione w Rozdziale 8.

Wiele osób uważa, że modele stały się obecnie towarem, więc dane są kluczowym czynnikiem wyróżniającym. Sprawia to, że inżynieria zbiorów danych staje się ważniejsza niż kiedykolwiek. Ilość danych, jakiej potrzebujesz, zależy od zastosowanej techniki adaptacji. Trenowanie modelu od zera zazwyczaj wymaga więcej danych niż dostrajanie, które z kolei wymaga ich więcej niż inżynieria promptów.

Niezależnie od ilości wymaganych danych, ich znajomość jest nieoceniona podczas testowania modelu, ponieważ dane użyte do treningu dostarczają istotnych informacji o mocnych i słabych stronach aplikacji.

Optymalizacja wnioskowania. *Optymalizacja wnioskowania* oznacza przyspieszenie działania modeli i obniżenie kosztów ich użycia. Była ona zawsze kluczowa w inżynierii ML — użytkownicy nigdy nie mają nic przeciwko szybszym modelom, a firmy zawsze korzystają na tańszym wnioskowaniu. Jednak w miarę skalowania modeli podstawowych i zwiększania kosztów oraz opóźnień wnioskowania optymalizacja stała się jeszcze ważniejsza.

Jednym z wyzwań związanych z modelami podstawowymi jest ich *autoregresyjny* charakter — generują one tokeny w sposób sekwencyjny. Jeśli model potrzebuje 10 ms na wygenerowanie jednego tokena, wygenerowanie 100 zajmie sekundę, a nawet więcej w przypadku dłuższych wyników. W czasach, gdy użytkownicy stają się coraz bardziej niecierpliwi, zmniejszenie opóźnienia modelu AI do 100 ms, typowego dla aplikacji internetowych (*https://oreil.ly/gGXZ-*), stanowi ogromne wyzwanie. Optymalizacja wnioskowania stała się aktywnym obszarem badań zarówno w przemyśle, jak i w środowisku akademickim.

W tabeli 1.4 podsumowano, jak zmienia się znaczenie różnych kategorii projektowania modeli w kontekście inżynierii AI.

Tabela 1.4. W jaki sposób różne obowiązki związane z tworzeniem modeli uległy zmianie wraz z pojawieniem się modeli podstawowych?

Kategoria	Tworzenie za pomocą tradycyjnych narzędzi ML	Tworzenie za pomocą modeli podstawowych
modelowanie i trenowanie	wiedza z zakresu uczenia maszynowego jest wymagana do trenowania modelu od podstaw	wiedza z zakresu uczenia maszynowego jest zalecana, ale nie konieczna[a]
inżynieria zbiorów danych	więcej wiedzy o inżynierii cech, zwłaszcza w przypadku danych tabelarycznych	mniej wiedzy o inżynierii cech, a więcej o deduplikacji danych, tokenizacji, wyszukiwaniu kontekstu i kontroli jakości
optymalizacja wnioskowania	ważne zagadnienie	jeszcze ważniejsze zagadnienie

[a] Wiele osób zaprzeczyłoby temu twierdzeniu, mówiąc, że wiedza ML jest niezbędna

Techniki optymalizacji wnioskowania, w tym kwantyzacja, destylacja i równoległość, zostaną przeanalizowane w rozdziałach od 7 do 9.

Projektowanie aplikacji

W przypadku tradycyjnej inżynierii ML, w której zespoły tworzą aplikacje przy użyciu własnych modeli, czynnikiem wyróżniającym jest ich jakość. Gdy używa się modeli podstawowych, wiele zespołów korzysta z tego samego modelu, a przewagę osiąga się poprzez proces projektowania aplikacji.

Warstwa projektowania aplikacji składa się z następujących elementów: ewaluacja, inżynieria promptów i interfejs AI.

Ewaluacja. *Ewaluacja* polega na zmniejszaniu poziomu ryzyka i odkrywaniu możliwości. Jest niezbędna w całym procesie adaptacji modelu. Używa się jej podczas wyboru modeli, porównywania postępów, określania, czy aplikacja jest gotowa do wdrożenia, a także do wykrywania problemów i możliwości ulepszeń w środowisku produkcyjnym.

Ewaluacja była zawsze ważna w inżynierii ML, a w przypadku modeli podstawowych jest ona z wielu powodów jeszcze ważniejsza. Wyzwania związane z ewaluacją modeli podstawowych zostaną przeanalizowane w rozdziale 3. Wynikają one głównie z otwartego charakteru modeli podstawowych i ich rozszerzonych możliwości. Na przykład w zadaniach ML o zamkniętym charakterze, takich jak wykrywanie oszustw, istnieją zazwyczaj oczekiwane prawdy podstawowe, z którymi można porównać wyniki modelu. Jeśli uzyskany wynik różni się od oczekiwanego, wiadomo, że model działa niepoprawnie. W przypadku zadań takich jak chatboty istnieje jednak tak wiele możliwych odpowiedzi na każdy prompt, że niemożliwe jest utworzenie wyczerpującej listy podstawowych prawd, z którymi można porównać wyniki.

Obszerna lista wielu metod adaptacyjnych również utrudnia ewaluację. System, który działa słabo przy użyciu jednej metody, może działać znacznie lepiej z inną. Gdy firma Google w grudniu 2023 roku wprowadziła model Gemini, twierdzono, że przewyższa on ChatGPT w teście MMLU (Hendrycks i in., 2020 (*https://arxiv.org/abs/2009.03300*)). Firma Google ewaluowała model Gemini przy użyciu metody inżynierii promptów o nazwie CoT@32 (*https://oreil.ly/VDwaR*). Polegała ona na tym, że modelowi Gemini zaprezentowano 32 przykłady, podczas gdy do ChatGPT przekazano tylko 5 przykładów. Gdy obu modelom zaprezentowano po pięć przykładów, ChatGPT działał lepiej (tabela 1.5).

Inżynieria promptów i konstruowanie kontekstu. *Inżynieria promptów* polega na skłonieniu modeli AI do generowania pożądanych wyników tylko na podstawie wprowadzonego zapytania, bez zmiany wag modelu. Historia ewaluacji modelu Gemini podkreśla wpływ inżynierii promptów na wydajność modelu. Dzięki zastosowaniu innej techniki inżynierii promptów wydajność modelu Gemini Ultra podczas testu MMLU wzrosła z 83,7% do 90,04%.

Dzięki odpowiednio skonstruowanym promptom można uzyskać naprawdę niesamowite rezultaty. Właściwe instrukcje sprawiają, że model wykonuje zadanie dokładnie tak, jak chcieliśmy. Inżynieria promptów to nie tylko nakazywanie modelowi, co ma zrobić, ale także dostarczanie mu niezbędnego kontekstu i narzędzi do wykonania zadania. W przypadku złożonych zadań,

Tabela 1.5. Różne prompty mogą znacząco wpływać na wydajność modeli, co pokazano w raporcie technicznym dotyczącym modelu Gemini (grudzień 2023)

	Gemini Ultra	Gemini Pro	GPT-4	GPT-3.5	PaLM 2-L	Claude 2	Inflection-2	Grok 1	Llama-2
Wydajność testu MMLU	90,04% CoT@32	79,13% CoT@8	87,29% CoT@32 (przez API)	70% 5 przykładów	78,4% 5 przykładów	78,5% CoT 5 przykładów	79,6% 5 przykładów	73,0% 5 przykładów	68,0%
	83,7% 5 przykładów	71,8% 5 przykładów	86,4% 5 przykładów (raport)						

które wymagają poszerzonego kontekstu, może być konieczne udostępnienie modelowi systemu zarządzania pamięcią, aby mógł analizować swoje wcześniejsze odpowiedzi. Inżynieria promptów zostanie przeanalizowana w rozdziale 5., natomiast rozdział 6. jest poświęcony budowaniu kontekstu.

Interfejs AI. *Interfejs AI* jest punktem kontaktowym i pozwala użytkownikom końcowym na interakcję z aplikacjami AI. Przed erą modeli podstawowych takie aplikacje mogły być tworzone jedynie przez firmy mające dostęp do wystarczających zasobów. Aplikacje były zazwyczaj wbudowane w istniejące produkty tych firm. Na przykład systemy wykrywania oszustw były częścią platform takich jak Stripe, Venmo i PayPal, a systemy rekomendacji były integralną częścią serwisów społecznościowych i aplikacji multimedialnych, takich jak Netflix, TikTok czy Spotify.

Dzięki modelom podstawowym każdy może obecnie tworzyć aplikacje AI. Mogą być one następnie oferowane jako samodzielne produkty lub wbudowane w inne, w tym te utworzone przez innych deweloperów. Na przykład ChatGPT i Perplexity to samodzielne produkty, podczas gdy Copilot GitHub jest powszechnie używany jako wtyczka w VSCode, a Grammarly jako rozszerzenie przeglądarki podczas pracy z Google Docs. Midjourney można używać zarówno poprzez samodzielną aplikację internetową, jak i w ramach integracji z Discordem.

Muszą istnieć narzędzia, które zapewniają interfejsy dla samodzielnych aplikacji AI lub ułatwiają integrację sztucznej inteligencji z istniejącymi produktami. Oto lista interfejsów, które zyskują popularność w aplikacjach AI:

- Samodzielne aplikacje internetowe, desktopowe i mobilne[26].
- Rozszerzenia przeglądarki, które pozwalają użytkownikom na przesyłanie szybkich zapytań do modeli AI podczas przeglądania.
- Chatboty zintegrowane z aplikacjami do czatowania, takimi jak Slack, Discord, WeChat i WhatsApp.

[26] Streamlit, Gradio i Plotly Dash to popularne narzędzia do tworzenia aplikacji internetowych AI.

- Wiele produktów, w tym VSCode, Shopify i Microsoft 365, oferuje interfejsy API, które pozwalają deweloperom integrować AI jako wtyczki i dodatki. Jak zostanie zaprezentowane w rozdziale 6., te interfejsy API mogą być również wykorzystywane przez agentów AI do interakcji ze światem.

Chociaż interfejs czatowy jest najczęściej używany, interfejsy AI mogą być również oparte na głosie (jak w przypadku asystentów głosowych) lub udostępnione w postaci wirtualnej osoby (w przypadku rozszerzonej i wirtualnej rzeczywistości).

Te nowe interfejsy AI oznaczają również nowe sposoby gromadzenia i wyodrębniania opinii użytkowników. Interfejs konwersacyjny znacznie ułatwia użytkownikom udzielanie informacji zwrotnej w języku naturalnym, ale tak uzyskane dane są trudniejsze do przetworzenia. Projektowanie interfejsu do uzyskiwania opinii użytkowników zostało przeanalizowane w rozdziale 10.

W tabeli 1.6 przedstawiono podsumowanie znaczenia różnych kategorii projektowania aplikacji w kontekście inżynierii AI.

Tabela 1.6. Znaczenie różnych kategorii podczas projektowania aplikacji w przypadku inżynierii AI i inżynierii ML

Kategoria	Tworzenie za pomocą tradycyjnych narzędzi ML	Tworzenie za pomocą modeli podstawowych
interfejs AI	mniej znacząca	znacząca
inżynieria promptów	nie dotyczy	znacząca
ewaluacja	znacząca	bardziej znacząca

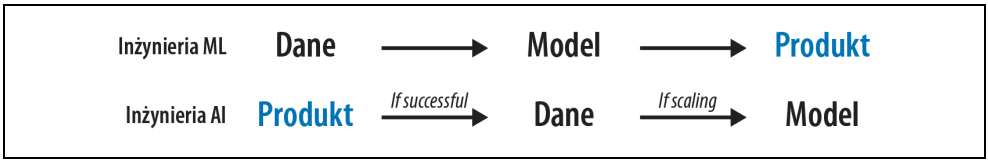
Inżynieria AI a inżynieria pełnowymiarowa

Zwiększone znaczenie projektowania aplikacji, a szczególnie interfejsów, zbliża inżynierię AI do projektowania pełnowymiarowego[27]. Rosnące znaczenie interfejsów prowadzi do zmian w projektowaniu narzędzi AI, co z kolei zachęca więcej inżynierów frontendowych. Dotychczas inżynieria ML opierała się głównie na Pythonie. Przed erą modeli podstawowych najpopularniejsze biblioteki ML wspierały głównie interfejsy API w Pythonie. Obecnie Python nadal cieszy się popularnością, ale rośnie również wsparcie dla interfejsów API w języku JavaScript, takich jak LangChain.js (*https://github.com/langchain-ai/langchainjs*), Transformers.js (*https://github.com/huggingface/transformers.js*), biblioteka Node OpenAI (*https://github.com/openai/openai-node*) czy AI SDK z firmy Vercel (*https://github.com/vercel/ai*).

Chociaż wielu inżynierów AI wywodzi się z typowego środowiska ML, coraz więcej z nich zajmowało się w przeszłości projektowaniem stron internetowych lub kompleksowym tworzeniem oprogramowania. Przewagą tych drugich nad tradycyjnymi inżynierami ML jest ich zdolność do szybkiego prototypowania pomysłów, pozyskiwania informacji zwrotnych i wdrażania zaplanowanych zmian.

[27] Anton Bacaj powiedział mi, że „inżynieria AI to po prostu inżynieria oprogramowania uwzględniająca modele AI umieszczone na stosie technologicznym".

W przypadku tradycyjnej inżynierii ML zazwyczaj zaczynasz od gromadzenia danych i trenowania modelu, a tworzenie produktu jest ostatnim etapem. Przy obecnej dostępności modeli AI można jednak zmienić kolejność i zacząć od budowy samego produktu, a dopiero później zainwestować w dane i modele, gdy aplikacja okaże się obiecująca (rysunek 1.16).



Rysunek 1.16. Nowy proces pracy w inżynierii AI nagradza tych, którzy potrafią szybko wdrażać zmiany. Obraz utworzony na podstawie artykułu The Rise of the AI Engineer (Shawn Wang, 2023 (https://oreil.ly/OOZK-)

W tradycyjnej inżynierii ML projektowanie modeli i produktów to często procesy rozłączne, a w wielu organizacjach twórcy rzadko uczestniczą w podejmowaniu ważniejszych decyzji. W przypadku modeli podstawowych inżynierowie AI są jednak zazwyczaj znacznie bardziej zaangażowani w tworzenie produktu.

Podsumowanie

Napisałam ten rozdział, ponieważ miał on spełnić dwa cele. Pierwszym z nich było wyjaśnienie powstania inżynierii AI jako dyscypliny, co stało się możliwe dzięki dostępności modeli podstawowych. Drugim celem było przedstawienie ogólnego procesu budowania aplikacji opartych na tych modelach. Mam nadzieję, że udało się to osiągnąć. Ponieważ był to rozdział wprowadzający, poruszano w nim jedynie pobieżnie wiele zagadnień, które bardziej szczegółowo zostaną przeanalizowane w dalszej części książki.

W rozdziale skoncentrowano się na dynamicznych zmianach, jakie zaszły w AI w ostatnich latach. Opisano przejście od tradycyjnych modeli językowych do dużych modeli językowych, co było możliwe dzięki metodzie treningowej opartej na samonadzorowaniu. Następnie przedstawiono proces, w którym modele językowe zaczęły łączyć różne rodzaje danych, przez co stały się modelami podstawowymi, co z kolei zapoczątkowało rozwój inżynierii AI.

Szybki rozwój inżynierii AI wynika z licznych zastosowań, które stały się możliwe dzięki nowo nabytym umiejętnościom modeli podstawowych. W tym rozdziale przeanalizowano niektóre z najważniejszych wzorców zastosowań, zarówno w przypadku użytkowników indywidualnych, jak i przedsiębiorstw. Mimo imponującej liczby już wdrożonych aplikacji AI jesteśmy wciąż na wczesnym etapie rozwoju inżynierii AI, ponieważ niezliczone innowacje są jeszcze przed nami.

Zanim rozpocznie się prace nad aplikacją, warto zadać sobie podstawowe pytanie: czy na pewno warto ją budować? W rozdziale poruszono tę kwestię i przeanalizowano kluczowe czynniki, które należy wziąć pod uwagę podczas projektowania aplikacji AI.

Choć termin „inżynieria AI” jest nowy, wywodzi się on z inżynierii ML — szeroko pojętej dziedziny zajmującej się budową aplikacji opartych na wszystkich modelach uczenia maszynowego. Wiele zasad inżynierii ML nadal znajduje zastosowanie w inżynierii AI. Ta nowa dziedzina niesie jednak ze sobą także nowe wyzwania i rozwiązania. W ostatniej części rozdziału przeanalizowano strukturę inżynierii AI oraz jej różnice w porównaniu z tradycyjną inżynierią ML.

Aspekty inżynierii AI, które trudno oddać w słowie pisanym, to niezwykła energia, kreatywność i talent inżynieryjny, wnoszone przez całą społeczność. Zbiorowy entuzjazm bywa przytłaczający — trudno nadążyć za nowymi technikami, odkryciami i osiągnięciami inżynieryjnymi, które wydają się pojawiać bez przerwy.

Na szczęście sztuczna inteligencja doskonale radzi sobie z agregacją informacji, co pozwala na sprawniejsze analizowanie nowych wydarzeń. Jednak nawet najlepsze narzędzia mają swoje ograniczenia. Im bardziej skomplikowana jest dana dziedzina, tym ważniejszy staje się dostęp do odpowiedniej metodologii, by uzyskiwać lepsze wyniki pracy. Ta książka powinna w tym pomóc.

W kolejnych rozdziałach będziesz stopniowo poznawać tę metodologię. Zaczniemy od zaprezentowania głównego elementu inżynierii AI — modeli podstawowych, które umożliwiają tworzenie wielu innowacyjnych aplikacji.

Skorowidz

A

adaptacja niskorzędowa, LoRA, 340
agenty, 280
  ewaluacja, 301
  hierarchia informacji, 305
  narzędzia, 283, 299
  planowanie, 286
  refleksja, 295
  środowisko, 281
  tryby błędów, 301
  zbiór akcji, 281
AI
  jako sędzia, 149, 150, 153
  zorientowana na dane, 364
  zorientowana na model, 364
akceleratory AI, 415
algorytm
  Annoy, 268
  Best Matching, 264
  FLANN, 268
  HNSW, 267
  IVF, 267
  LSH, 267
  osadzeń, 147
  Product Quantization, 267
  RLHF, 100
  SGD, 324
  SPTAG, 268
  TF-IDF, 263
algorytmy wyszukiwania, 261
  łączenie, 271
  porównanie, 268
analiza
  n-gramowa, 207
  nieokreśloności, 207
API
  modeli, 191, 199
  online, 407
  wsadowe, 407
aplikacja
  GitHub Copilot, 480
  Google Gemini, 477
  Google Photos, 478
  kryteria ewaluacji, 168
  Midjourney, 479
  projektowanie procesu ewaluacji, 208
  ścieżki rozwoju, 321
  wybór modelu, 187
architektura
  AlexNet, 81
  modelu CLIP, 148
  RWKV, 82
  seq2seq, 76
  SSM, 82
  systemów AI, 445
    dodanie wzorców agentowych, 459
    monitorowanie, 461
    orkiestracja potoku AI, 467
    rozszerzenie kontekstu, 446
    wprowadzenie bramki dostępowej, 451
    wprowadzenie routingu, 451
    wprowadzenie zabezpieczeń, 447
    zmniejszenie opóźnień, 456
  transformer, 75
atak typu wstrzykiwanie promptów, 243
autoewaluacja, 157
autokrytyka, self-critique, 233
autoweryfikacja, 175

B

BERT, 336, 337
bezpieczeństwo, 178, 240
biblioteka
  Annoy, 266
  Axolotl, 347
  Hnswlib, 266
  LitGPT, 347, 359
  Llama Police, 359
  LLaMA-Factory, 359
  PEFT, 347, 359
  PyTorch, 435
  ScaNN, 266
  TensorRT, 436
bit, 135
BloombergGPT, 318
błędy
  typograficzne, 238
  w planowaniu, 302
  związane z działaniem narzędzi, 303
bramka dostępowa, 454
bufor promptów, prompt cache, 439
buforowanie
  dokładne, 456
  semantyczne, 457

C

Claude, 179
Claude 3, 28
CLIP, 29, 74
CoT, chain-of-thought, 233
CPU, 416

D

DALL-E, 476
dane
  czyszczenie, 398
  deduplikacja, 396
  demonstracyjne, 98
  etykietowanie, 213, 376
  filtrowanie, 398
  formatowanie, 399
  generowanie, 379, 391
  ilość, 371
  inspekcja, 395
  jakość, 367
  pokrycie, 369
  pozyskiwanie, 376
  przetwarzanie, 394
  przygotowanie, 365
  publicznie dostępne, 378
  referencyjne, 140
  skażone, 205
  syntezowanie, 379
    metody, 381
    stosowanie, 379
    wspierane przez AI, 385
    z instrukcjami, 387
  treningowe, 68, 193
  ustrukturyzowane, 113
  weryfikacja, 389
dekodowanie
  autoregresyjne, 424
  równoległe, 426
  spekulatywne, 424
destylacja modelu, 393
determinizm, 119
dopasowanie, 161
  rozmyte, fuzzy matching, 143
dostrajanie, 58, 60, 97, 118, 310–361
  a RAG, 319
  biblioteki, 359
  częściowe, partial finetuning, 336
  do zadań specyficznych, 317
  efektywne parametrowo, 335
  hiperparametrów, 359
  instrukcji, 94
  jednoczesne, 349
  metoda, 358
    LoRA, 340
    PEFT, 338
  modele startowe, 357
  nadzorowane, SFT, 94, 96
  ograniczenia pamięciowe, 322
  parametry trenowane, 323
  preferencji, 94, 99
  propagacja wsteczna, 323
  sekwencyjne, 349
  stosowanie, 314
  techniki, 314, 334
  wielozadaniowe, 348
  z użyciem miękkich promptów, 339
  za pomocą modelu nagradzania, 103
dryf, 465

duży
  model językowy, LLM, 21, 27
  model multimodalny, LMM, 29
dzienniki, 464

E

efektywność obliczeniowa czasu testowania, 110
ekstrakcja
  danych treningowych, 250
  informacji, 240, 248
  promptu, 240
ekstrapolacja skalowania, 90
entropia, 133
  krzyżowa, 132, 134
epoka, 361
ewaluacja, 62, 64, 127
  dokładna, 138
  kryteria, 168
    koszty i opóźnienia, 185
    zdolności generacyjne, 172
    zdolności specyficzne dla danej dziedziny, 170
    zdolność do podążania za instrukcjami, 180
  MEWR, 149
  modeli podstawowych, 128
  poprawności funkcjonalnej, 139
  porównawcza, comparative evaluation, 160, 163, 166
  projektowanie procesu, 208
    komponenty systemu, 208
    ocena, 215
    określenie metod, 212
    tworzenie wytycznych, 210
  punktowa, pointwise evaluation, 159
  subiektywna, 138
  wyszukiwania, 277

F

faworyzowanie własnych odpowiedzi, 156
FLOP, 86
FLOP/s, 418
formaty numeryczne, 329
funkcje aktywacji, 80
funkcjonalność, 195

G

GAN, 81
Gemini, 28, 73, 179
generatywna sztuczna inteligencja, 24
  agregacja informacji, 45
  automatyzacja przepływu danych, 46
  boty konwersacyjne, 44
  edukacja, 43
  generowanie tekstów, 41
  organizowanie danych, 46
  programowanie, 39
  tworzenie obrazów i wideo, 41
  zastosowania, 37
generatywne sieci przeciwstawne, GAN, 81
generowanie
  języka naturalnego, NLG, 172
  planu, 290
  sztucznych danych, 379, 383
  wspomagane wyszukiwaniem, RAG, 31, 257
    a dostrajanie, 319
    algorytmy wyszukiwania, 261
    architektura systemu, 260
    dla danych tabularycznych, 278
    multimodalne, 277
    optymalizacja wyszukiwania, 272
    pamięć, 304
GPT, 73
GPT-4V, 28
GPTs, 179
GPU, 416
Grok, 92

H

halucynacje, 121
HBM, High-Bandwidth Memory, 419
hiperparametr, 90, 359
  liczba epok, 361
  waga straty dla promptu, 361
  wielkość paczki, 360
  współczynnik uczenia, 360

I

identyfikowanie szkodliwych treści, 179
IDP, Intelligent Data Processing, 46
informacja zwrotna
  od użytkowników, 469
  ograniczenia systemu gromadzenia, 482

rodzaje, 473
system gromadzenia, 475, 479
typ, 473
w języku naturalnym, 471
z rozmów, 469
infrastruktura, 56
InstructGPT, 101
inteligentne przetwarzanie danych, IDP, 46
interfejs AI, 63, 64
inżynieria
AI, 54, 57
planowanie aplikacji, 47
rozwój, 22
stos technologiczny, 54
ML, 54, 57
odwrotna promptów, 242
pełnowymiarowa, 64
promptów, 31, 62, 64, 218–256
najlepsze zasady, 226
strategia zabezpieczania, 240
zbiorów danych, 60, 61, 363

J

jądra obliczeniowe, 433

K

klasyfikator intencji, 452
kompilatory, 433
kompresja modelu, 423
konkatenacja, 357
kontekst, 221, 229
długość i efektywność, 224
tworzenie, 62, 447
wzorzec agentowy, 280, 459
wzorzec RAG, 258
koszty, 185
API, 196
wdrożenia, 196
kradzież danych, 248
kwantyzacja, 137, 330
trenowania, 333
wnioskowania, 331

L

licencje modeli, 189
liczba
FLOP-ów, 89
parametrów, 84
repozytoriów ewaluacyjnych, 131
tokenów, 85
Llama, 73
LLM, large language model, 21, 27
LMM, large multimodal model, 29
logarytm naturalny, nat, 135
LoRA, Low-Rank Adaptation, 340
działanie, 342
konfiguracje, 343
metoda kwantyzowana, 347
udostępnianie adapterów, 345

Ł

łańcuch rozumowania, CoT, 233, 365
łączenie warstw, 355

M

MBU, Model Bandwidth Utilization, 413
mechanizmy
ochronne, 450
skupiania uwagi, 78, 429, 431, 432
zarządzania umiejętnościami, 300
metoda Medusa, 429
MFU, Model FLOP/s Utilization, 413
mierzenie podobieństwa, 140
MLOps, ML operations, 31
model
jako usługa, 21
językowy, 22
autoregresyjny, 24
maskowany, 24
nagradzania, 100, 158
osadzania, 29
preferencji, 158
modele
open source, 191
podstawowe, 28, 30, 67–126
podstawowe jako planiści, 289
specyficzne, 73
wielojęzyczne, 69

modelowanie, 75
  architektura modelu, 75
  niespójność modelu, 119
  optymalizacja modelu, 422
  rozmiar modelu, 84
MoE, 355
monitorowanie systemu, 461, 462

N

nadzorowanie, 26
nadzór języka naturalnego, 29
nagradzanie, 100, 103
narzędzie
  Apache TVM, 436
  Azure/PyRIT, 252
  CHATS-lab/persuasive_jailbreaker, 252
  Continue Dev, 239
  Dotprompt, 239
  DSPy, 235
  greshake/llm-security, 252
  Guidance, 237
  Humanloop, 239
  Instructor, 237
  leondz/garak, 252
  lm-evaluation-harness, 199
  OpenPrompt, 235
  Outlines, 237
  Promptbreeder, 236
  Promptfile, 239
  TextGrad, 236
nieokreśloność, 135, 136
niespójność modelu, 119
NLG, natural language generation, 172

O

obliczanie nieokreśloności tekstu, 138
obrona przed atakami
  na poziomie modelu, 253
  na poziomie promptów, 254
  na poziomie systemu, 255
ocena punktowa, pointwise evaluation, 100
odgrywanie ról, 184, 245
odpowiedzi kanoniczne,
  canonical responses, 141
ograniczenia
  pamięciowe, 405
  przepustowości, 405
omijanie zabezpieczeń modelu, 243
Open CLIP, 74
open source, 191
opóźnienia, 185, 456
optymalizacja
  modelu, 422
  usługi wnioskowania, 436
  wnioskowania, 61, 402, 422, 435
orkiestracja potoku AI, 467
osadzanie, embedding, 145, 146, 265
otwarte oprogramowanie, 189

P

pamięć, 325
  CPU, 419
  długoterminowa, 304
  HBM GPU, 420
  krótkoterminowa, 304
  podręczna KV, 431, 432
  potrzebna do trenowania, 326
  potrzebna do wnioskowania, 325
  SRAM GPU, 420
PandaLM, 159
paradoks Simpsona, 213
parametr, 27, 90
parametry trenowane, 323
pętla sprzężenia zwrotnego, 484
planowanie, 286, 290
  aplikacji AI, 47
  błędy, 302
  refleksja, 295
  ziarnistość, 294
  złożone, 295
podobieństwo
  dopasowanie dokładne, 142
  leksykalne, 143
  osadzeń, embedding similarity, 145
  semantyczne, 145
poprawność funkcjonalna, 139
porównanie odpowiedzi, 477
postprocessing, 116
post-trening, 60, 93, 137
  dostrajanie nadzorowane, 96
  dostrajanie preferencji, 99
prawa autorskie, 193, 249
prawdopodobieństwo, 119
prawdziwe dane, ground truths, 141
precyzja, 329, 331

projektowanie
  aplikacji, 55, 62
  modelu, 55, 58
  sterowane ewaluacją, 169
prompty, 116, 218–256
  ataki zautomatyzowane, 246
  buforowanie, 439
  ciągłe eksperymentowanie, 234
  dołączanie kontekstu, 229
  inżynieria odwrotna, 242
  manipulacja formatowaniem odpowiedzi, 245
  narzędzia, 235
  obrona przed atakami, 252
  odgrywanie ról, 245
  omijanie zabezpieczeń, 243
  porządkowanie, 238
  precyzyjne instrukcje, 227
  rozkład, 97
  systemowe, 222, 242
  twarde i miękkie, 339
  tworzenie, 219
  upraszczanie złożonych zadań, 231
  użytkownika, 222
  wstrzykiwanie, 243
  wstrzykiwanie pośrednie, 247
  z łańcuchem rozumowania, 233
  zabezpieczanie, 240
  zaciemnianie treści, 244
  zarządzanie, 238
  zastrzeżone, 242
propagacja wsteczna, 323
próbkowanie, 68, 103
  dane ustrukturyzowane, 113
  efektywność TTS, 110
  ograniczone, 117
  strategia top-k, 108
  strategia top-p, 109
  temperatura, 105
  warunek zatrzymania, 109
prywatność, 248
  danych, 192
przejrzystość działania systemu, 461, 462
przepustowość, throughput, 411
przeszukiwanie wiązkowe, beam search, 110
przetwarzanie
  danych, 394
  języka naturalnego, NLP, 28
  równoległe, 441
  wsadowe, batching, 436

R

RAG, retrieval-augmented generation, 31, 257
rankingi
  publiczne, 200
  własne, 203
redukcja precyzji, 331
refleksja, 295
rekurencyjne sieci neuronowe, RNN, 76
reprezentacje numeryczne, 328
RNN, 77
routing, 451
rozkład
  długości odpowiedzi, 396
  dziedzin, 74
  promptów, 97
rozmiary osadzeń, 147

S

SAFE, Search-Augmented Factuality Evaluator, 175
samonadzorowanie, 26
scalanie modeli, 348, 351
  konkatenacja, 357
  łączenie warstw, 355
  sumowanie, 351
sędzia
  AI, 148, 150, 153
  wykorzystujący referencje, 158
skala, 21
skalowanie, 91
  odwrotne, 88
skażenie danych, 205
skupianie uwagi, 77, 429, 431, 432
słownik, 23
SMI, System Management Interface, 413
spójność faktograficzna, 173
stos technologiczny AI, 54
sumowanie modeli
  kombinacja liniowa, 351
  sferyczna interpolacja liniowa, 353
  usuwanie zbędnych parametrów, 354
symulacja wirtualna, 384
syntezowanie danych, 379
szkodliwe treści, 179

Ś

ślad
  wykonania, 464
  żądania, 466

T

tablica Chatbot Arena, 164
TBT, Time Between Tokens, 409
test
  Advbench, 252
  ARC-C, 200
  BBH, 202
  BIG-bench, 199
  GPQA, 202
  GSM-8K, 201
  HellaSwag, 201
  HELM, 205
  IFEval, 181, 202
  INFOBench, 181, 183
  LegalBench, 201
  MATH, 201
  MATH lvl 5, 202
  MedQA, 201
  MMLU, 171, 194, 200
  MMLU-PRO, 202
  MuSR, 202
  NarrativeQA, 201
  Natural Questions, 201
  PromptRobust, 252
  RoleLLM, 184
  TruthfulQA, 177, 201
  WinoGrande, 201
  WMT 2014, 201
testy
  agregacja, 200
  korelacja, 203
  wybór, 200
token, 23
tokenizacja, 23
TPOT, Time Per Output Token, 409
trenowanie, 59, 61, 95, 326
  kwantyzacja, 333
  zapotrzebowanie na pamięć, 326
TTFT, Time to First Token, 409
TTS, test time compute, 110
tworzenie
  kontekstu, 62, 447
  promptów, 219

U

uczenie
  few-shot, 220
  kontekstowe, 220
  maszynowe, ML, 21
  modeli za pomocą promptów, 220
  przez wzmacnianie, RL, 94, 123
    RLAIF, 94
    RLHF, 94
  transferowe, 311
  zero-shot, 220
  zespołowe, 351
uruchamianie modelu, 198

W

warstwy architektury modeli, 83
wąskie gardła
  obliczeniowe, 404
  skalowalności, 91, 163
wektor
  klucza, 78
  wartości, 78
  zapytania, 78
weryfikacja
  odpowiedzi, 123
  wspomagana wiedzą, 175
wizualizacja struktury wag, 81
wnioskowanie, 325, 403
  akceleratory AI, 415
  dekodowanie, 438
  interfejsy API, 407
  kwantyzacja, 331
  optymalizacja, 402, 403, 422, 435
  optymalizacja usługi, 436
  przetwarzanie równoległe, 441
  wskaźniki, 409
  wstępne uzupełnianie, 438
  wydajność, 409
  z odniesieniem, 426, 427
  zapotrzebowanie na pamięć, 325
wskaźnik
  bits-per-byte, BPB, 132–135
  bits-per-character, BPC, 132, 134
  CFR, 461
  efektywna przepustowość, 411
  entropia, 133
  entropia krzyżowa, 132, 134
  FLOP, 86

FLOP/s, 418
MAP, 269
MBU, 413
MFU, 413
MRR, 269
MTTD, 461
MTTR, 461
NDCG, 269
nieokreśloność, 132, 135, 136
opóźnienie, 409
płynność, 172
poprawność funkcjonalna, 139
poziom podobieństwa, 140
przepustowość, 411
spójność, 172
spójność faktograficzna, 173
TBT, 409
TPOT, 409, 464
TTFT, 409, 464
wykorzystanie GPU, 413
wskaźniki, 462
biznesowe, 211
ewaluacyjne, 211
współczynniki
akceptacji, 425
wygranych, 162
wstępne uzupełnianie, prefill, 77
wstrzykiwanie promptów, 243
wybór
metod ewaluacji, 212
modelu, 187
optymalnego rozwiązania, 453
wydajność modeli, 74, 89, 194, 204, 303
wykrywanie dryfu, 465
wyszukiwanie
gęste, 262
kontekstowe, 275
oparte na osadzeniach, 265, 270
oparte na terminach, 262, 270
optymalizacja, 272
rzadkie, 262
semantyczne, 270
wektorowe, 266
wywoływanie funkcji, 292
wzorce konstruowania kontekstu
agenty, 280, 459
generowanie wspomagane wyszukiwaniem, RAG, 258

Z

zabezpieczenia danych
wejściowych, 448
wyjściowych, 449
zaciemnianie treści, 244
zawody podatne na AI, 36
zbiory danych, 363
C4, 74
Common Crawl, 69
publicznie dostępne, 378
zdegenerowana pętla sprzężenia zwrotnego, 484
zdolności
generacyjne, 172
specyficzne dla danej dziedziny, 170
zdolność do podążania za instrukcjami, 180
zrównoleglenie
kontekstu, context parallelism, 443
modelu, model parallelism, 441
replik, replica parallelism, 441
sekwencji, sequence parallelism, 443
tensora, tensor parallelism, 441
zużycie energii, 420

Ta książka jest kompleksowym przewodnikiem po budowaniu generatywnych aplikacji AI w środowiskach produkcyjnych.

Luke Metz, były współtwórca ChatGPT w OpenAI

Modele bazowe (foundation models) zapoczątkowały prawdziwy rozkwit aplikacji opartych na sztucznej inteligencji. AI stała się potężnym narzędziem rozwojowym, którego dziś może używać niemal każdy. Decyzja o stworzeniu własnej aplikacji AI wymaga jednak zrozumienia procesu budowy i świadomego podejmowania decyzji projektowych.

Ten przystępnie napisany i praktyczny podręcznik pokazuje, czym jest inżynieria AI i w jaki sposób tworzy się aplikacje z wykorzystaniem łatwo dostępnych modeli bazowych. Dowiesz się, jak się poruszać w świecie sztucznej inteligencji, czym dokładnie są modele, zbiory danych, wskaźniki oceny i wzorce aplikacyjne. W książce przedstawiono również, krok po kroku, kompletny proces tworzenia aplikacji AI i ich efektywnego wdrażania. Poszczególne zagadnienia zostały zilustrowane dobrze udokumentowanymi studiami przypadków. Nie zabrakło również komentarzy odnoszących się do zaimplementowania systemu opinii użytkowników w celu zapewnienia przydatnych informacji zwrotnych.

Najważniejsze zagadnienia:

- istota inżynierii AI
- proces tworzenia aplikacji AI
- różne techniki adaptacji modeli, w tym inżynieria promptów, generowanie wspomagane wyszukiwaniem (RAG), dostrajanie modeli, agenty i inżynieria zbiorów danych
- wąskie gardła modeli podstawowych i ich pokonywanie
- zasady wyboru modelu, wskaźników, danych oraz wzorców projektowych

Chip Huyen działa na pograniczu AI, danych i storytellingu. Wcześniej prowadziła zajęcia z projektowania systemów uczenia maszynowego na Uniwersytecie Stanforda. Jej książka *Jak projektować systemy uczenia maszynowego* została przetłumaczona na ponad 10 języków.

Pozycja obowiązkowa dla każdego profesjonalisty, który chce wdrożyć AI w swojej firmie!

Vittorio Cretella, były dyrektor IT w P&G i Mars

Helion
helion.pl
HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-2628-8
9 788328 926288
Cena: 129,00 zł