Sebastian Raschka

STWÓRZ WŁASNE AI

Jak od podstaw zbudować duży model językowy

Helion

Tytuł oryginału: Build a Large Language Model (From Scratch)

Tłumaczenie: Radosław Meryk

ISBN: 978-83-289-2497-0

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
helion.pl/user/opinie/stwlai
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

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

1 Czym są duże modele językowe?

W tym rozdziale:

- Ogólne objaśnienie podstawowych pojęć dotyczących dużych modeli językowych
- Architektura transformera, z którego wywodzą się modele LLM
- Plan budowania modelu LLM od podstaw

Duże modele językowe (ang. *Large Language Models* — LLM), takie jak te oferowane w systemie ChatGPT firmy OpenAI, to modele głębokich sieci neuronowych, które opracowano w ciągu ostatnich kilku lat. Modele LLM zapoczątkowały nową erę przetwarzania języka naturalnego (ang. *Natural Language Processing* — NLP). Zanim pojawiły się modele LLM, tradycyjne metody AI doskonale sprawdzały się w zadaniach kategoryzacji, takich jak klasyfikacja spamu e-mail, czy też proste zadania rozpoznawania wzorców, które można było opisać za pomocą ręcznie tworzonych reguł lub z użyciem prostszych modeli. Zazwyczaj jednak, zwłaszcza w zadaniach językowych, które wymagały złożonych umiejętności rozumienia i tworzenia, takich jak parsowanie szczegółowych instrukcji, przeprowadzanie analizy kontekstowej czy tworzenie spójnego i odpowiedniego do kontekstu dokumentu, wyniki uzyskiwane za pomocą tradycyjnych metod były gorsze. Na przykład poprzednie generacje modeli językowych nie były w stanie wykonać tak trywialnego dla współczesnych modeli LLM zadania jak napisanie wiadomości e-mail na podstawie listy słów kluczowych.

Modele LLM mają niezwykłe możliwości rozumienia, generowania i interpretowania ludzkiego języka. Warto jednak wyjaśnić, że kiedy mówimy, że modele językowe „rozumieją", mamy na myśli ich zdolność przetwarzania i generowania tekstu w sposób, jaki wydaje się spójny i odpowiedni do kontekstu, a nie to, że mają ludzką świadomość lub wiedzę.

Dzięki postępom w uczeniu głębokim, które jest podzbiorem uczenia maszynowego i sztucznej inteligencji (AI) skoncentrowanym na sieciach neuronowych, modele LLM szkoli się na ogromnych ilościach danych tekstowych. To wielkoskalowe szkolenie pozwala modelom LLM na uchwycenie głębszych informacji kontekstowych i więcej subtelności ludzkiego języka w porównaniu z poprzednimi podejściami. W rezultacie wprowadzenie modeli LLM znacznie poprawiło wydajność w szerokim zakresie zadań NLP: w tłumaczeniach tekstu, analizie tonu, odpowiadaniu na pytania i wielu innych.

Inną ważną różnicą między współczesnymi modelami LLM a wcześniejszymi modelami NLP jest to, że wcześniejsze modele NLP zwykle projektowano do konkretnych zadań, takich jak kategoryzacja tekstu, tłumaczenie języka itp. Podczas gdy wspomniane wcześniejsze modele NLP wyróżniały się w wąskich zastosowaniach, modele LLM wykazują większą biegłość w szerokim zakresie zadań NLP.

Sukces modeli LLM można przypisać architekturze transformerów, która leży u podstaw wielu modeli LLM, oraz ogromnej ilości danych, na których LLM są szkolone. Ta wielka ilość danych pozwala im uchwycić szeroką gamę niuansów językowych, kontekstów i wzorców, których ręczne zakodowanie byłoby wyzwaniem.

Zwrot w kierunku modeli opartych na architekturze transformerów i wykorzystania do szkolenia modeli LLM dużych zbiorów danych zasadniczo zmienił dziedzinę NLP — zapewnił bardziej wydajne narzędzia do rozumienia ludzkiego języka i wykonywania związanych z tym zadań.

Poniższe podrozdziały są podstawą do zrozumienia modeli LLM przez zaimplementowanie w kodzie krok po kroku opartego na architekturze transformera modelu LLM podobnego do ChatGPT.

1.1. *Czym jest model LLM?*

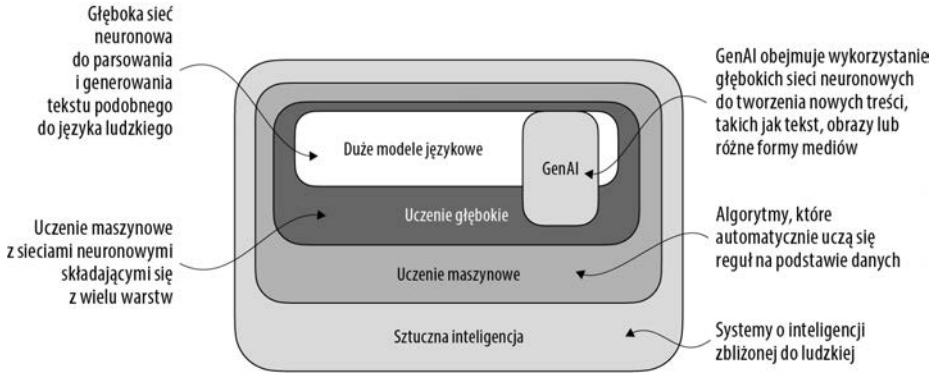
LLM to sieć neuronowa zaprojektowana do rozumienia ludzkiego języka, zdolna do generowania tekstu i reagowania na tekst w sposób podobny do tego, w jaki robią to ludzie. Modele LLM są głębokimi sieciami neuronowymi przeszkolonymi na ogromnych ilościach danych tekstowych, czasami obejmujących duże fragmenty całego tekstu publicznie dostępnego w internecie.

Słowo „duży” w nazwie „duży model językowy” odnosi się zarówno do rozmiaru modelu wyrażonego liczbą parametrów, jak i do ogromnego zbioru danych, na którym go przeszkolono. Takie modele często mają dziesiątki, a nawet setki miliardów parametrów. Są to dostrajalne wagi w sieci, optymalizowane podczas szkolenia w celu prognozowania następnego słowa w sekwencji. Prognozowanie następnego słowa ma sens, ponieważ wykorzystuje sekwencyjną naturę języka w kontekście szkolenia modeli w zakresie rozumienia kontekstu, struktury i relacji w tekście. Jest to jednak bardzo proste zadanie, dlatego wielu badaczy jest zaskoczonych tym, że w ten sposób można

stworzyć tak sprawne modele. W kolejnych rozdziałach omówię i zaimplementuję krok po kroku procedurę szkolenia w celu przewidywania następnego słowa.

Modele LLM wykorzystują architekturę określaną jako *transformer*, dzięki której podczas tworzenia prognoz mogą selektywnie zwracać uwagę na różne części danych wejściowych. Dzięki temu stają się szczególnie biegłe w radzeniu sobie z niuansami i złożonością ludzkiego języka.

Ponieważ modele LLM są zdolne do *generowania* tekstu, określa się je również jako formę *generatywnej sztucznej inteligencji*, nazywanej często w skrócie *GenAI*. Jak pokazano na rysunku 1.1, sztuczna inteligencja obejmuje szerszą dziedzinę maszyn zdolnych do tworzenia, które umieją wykonywać zadania wymagające inteligencji podobnej do ludzkiej, w tym rozumienia języka, rozpoznawania wzorców i podejmowania decyzji. Sztuczna inteligencja obejmuje takie poddziedziny jak uczenie maszynowe i uczenie głębokie.



Rysunek 1.1. Jak sugeruje to hierarchiczne przedstawienie relacji między różnymi dziedzinami AI, modele LLM reprezentują konkretne zastosowanie technik uczenia głębokiego. Wykorzystują ich podobną do ludzkiej zdolność do przetwarzania i generowania tekstu. Uczenie głębokie to wyspecjalizowana gałąź uczenia maszynowego, która koncentruje się na wykorzystaniu wielowarstwowych sieci neuronowych. Uczenie maszynowe i uczenie głębokie to dziedziny mające na celu implementację algorytmów umożliwiających komputerom uczenie się na podstawie danych i wykonywanie zadań, które zwykle wymagają ludzkiej inteligencji

Algorytmy wykorzystywane do implementowania sztucznej inteligencji są przedmiotem zainteresowania dziedziny uczenia maszynowego. W szczególności uczenie maszynowe obejmuje rozwijanie algorytmów, które potrafią uczyć się z danych i wykonywać prognozy lub podejmować decyzje na podstawie tych danych bez wyraźnego zaprogramowania logiki wnioskowania. Aby to zilustrować, jako praktyczne zastosowanie uczenia maszynowego wyobraźmy sobie filtr antyspamowy. Zamiast ręcznie pisać reguły identyfikacji spamu, wystarczy przekazać do algorytmu uczenia maszynowego przykłady wiadomości e-mail oznaczonych jako spam i wiadomości e-mail, które nie są spamem. Przez minimalizowanie błędu w swoich prognozach model uczy

się na zbiorze danych szkoleniowych rozpoznawać wzorce i cechy wskazujące na spam, co umożliwia klasyfikowanie nowych wiadomości e-mail jako spam bądź nie.

Jak pokazałem na rysunku 1.1, uczenie głębokie to podzbiór uczenia maszynowego, który koncentruje się na wykorzystaniu do modelowania w danych złożonych wzorców i abstrakcji sieci neuronowych obejmujących co najmniej trzy warstwy (znane również jako głębokie sieci neuronowe). W przeciwieństwie do uczenia głębokiego, tradycyjne uczenie maszynowe wymaga ręcznego wyodrębniania cech. Oznacza to, że zadanie zidentyfikowania i wybrania najbardziej odpowiednich cech dla modelu spoczywa na ludzkich ekspertach.

Chociaż dziedzina sztucznej inteligencji jest obecnie zdominowana przez uczenie maszynowe i uczenie głębokie, obejmuje również inne podejścia — na przykład wykorzystuje systemy oparte na regułach, algorytmy genetyczne, systemy eksperckie, logikę rozmytą oraz rozumowanie symboliczne.

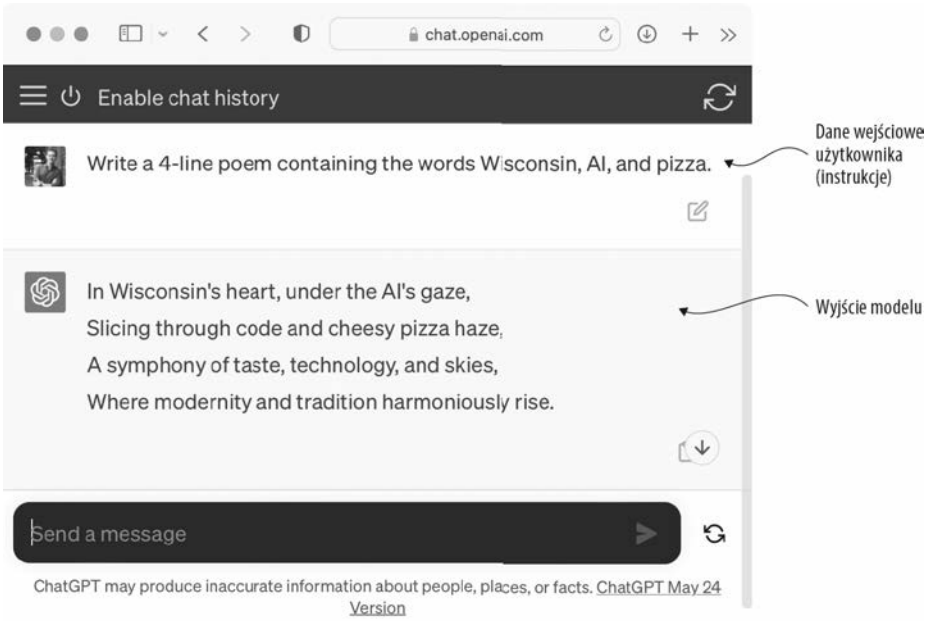
Wracając do przykładu klasyfikacji spamu, w tradycyjnym uczeniu maszynowym ludzcy eksperci mogą ręcznie wyodrębniać cechy z tekstu wiadomości e-mail, takie jak częstość niektórych słów wyzwalających (na przykład „nagroda", „wygrałeś", „za darmo"), liczba wykrzykników, użycie wszystkich wielkich liter lub obecność podejrzanych odnośników. Ten zbiór danych, utworzony na podstawie zdefiniowanych przez ekspertów cech, jest następnie wykorzystywany do przeszkolenia modelu. W przeciwieństwie do tradycyjnego uczenia maszynowego, uczenie głębokie nie wymaga ręcznej ekstrakcji cech. Oznacza to, że ludzcy eksperci nie muszą identyfikować i wybierać najbardziej odpowiednich cech dla modelu (jednak zarówno tradycyjne techniki uczenia maszynowego, jak i techniki uczenia głębokiego stosowane do klasyfikacji spamu nadal wymagają gromadzenia etykiet, takich jak „spam" lub „nie spam", które muszą być wybierane przez eksperta w dziedzinie lub użytkowników).

Przyjrzyjmy się wybranym problemom, które można dziś rozwiązać za pomocą modeli LLM, wyzwaniom, którym zastosowanie modeli LLM pozwoliło sprostać, oraz ogólnej architekturze modeli LLM, którą zaimplementujemy w dalszej części książki.

1.2. *Zastosowania modeli LLM*

Dzięki zaawansowanym możliwościom parsowania i rozumienia nieustrukturyzowanych danych tekstowych modele LLM mają szeroki zakres zastosowań w różnych dziedzinach. Obecnie modele LLM wykorzystuje się do tłumaczenia maszynowego, generowania nowych tekstów (patrz rysunek 1.2), analizy tonu, tworzenia streszczeń dokumentów i wielu innych zadań. Ostatnio używa się ich także do tworzenia treści, na przykład pisania beletrystyki, artykułów, a nawet komputerowego kodu.

Modele LLM oferują również duże możliwości zastosowania w zaawansowanych chatbotach i wirtualnych asystentach, takich jak ChatGPT firmy OpenAI i Gemini firmy Google (dawniej znany jako Bard), które umieją odpowiadać na pytania użytkowników



Rysunek 1.2. Interfejsy modeli LLM umożliwiają komunikację między użytkownikami a systemami sztucznej inteligencji w języku naturalnym. Ten zrzut ekranu pokazuje ChatGPT piszący wiersz zgodnie ze specyfikacją użytkownika

i uzupełniać funkcjonalność tradycyjnych wyszukiwarek, takich jak Google Search i Microsoft Bing.

Co więcej, modele LLM można wykorzystywać do skutecznego wyszukiwania wiedzy z ogromnych ilości tekstu w specjalistycznych dziedzinach, takich jak medycyna czy prawo. Obejmuje to przeglądanie dokumentów, streszczanie długich fragmentów i odpowiadanie na techniczne pytania.

W skrócie — modele LLM są nieocenione w automatyzacji prawie każdego zadania związanego z analizą i generowaniem tekstu. Ich zastosowania są niemal nieograniczone, a dzięki ciągłym innowacjom i odkrywaniu nowych sposobów ich wykorzystania mają one potencjał, by na nowo określić naszą relację z techniką, czyniąc ją bardziej konwersacyjną, intuicyjną i dostępną.

W tej książce skoncentruję się na gruntownym zrozumieniu, jak działa LLM, oraz pokażę, jak zakodować model LLM, który potrafi generować teksty. Zapoznasz się także z technikami, które pozwalają modelom LLM wykonywać zapytania — od odpowiadania na pytania po tworzenie streszczeń tekstu, tłumaczenie tekstu na różne języki itp. Mówiąc inaczej, przy okazji budowania krok po kroku złożonego asystenta LLM, podobnego do ChatGPT, dowiesz się, jak działają takie systemy.

1.3. *Etapy tworzenia modeli LLM i korzystania z nich*

Po co mielibyśmy tworzyć własne modele LLM? Kodowanie modeli LLM od podstaw jest doskonałym ćwiczeniem pozwalającym zrozumieć ich mechanikę i ograniczenia. Ponadto pozwala zdobyć wiedzę niezbędną do wstępnego szkolenia istniejących architektur LLM typu open source bądź dostrajania ich do własnych zbiorów danych lub zadań specyficznych dla domeny.

> *UWAGA* Większość współczesnych modeli LLM jest implementowana z użyciem biblioteki głębokiego uczenia `PyTorch`. W tej książce również skorzystamy z tej biblioteki. Obszerne wprowadzenie w tematykę biblioteki `PyTorch` znajdziesz w „Dodatku A".

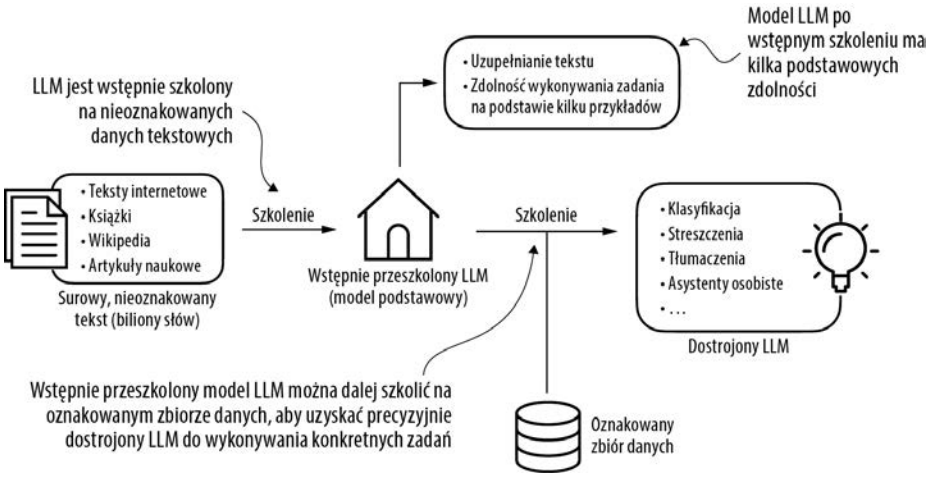
Z badań wynika, że niestandardowe modele LLM — te dostosowane do konkretnych zadań lub dziedzin — mogą przewyższać wydajnością modele LLM ogólnego przeznaczenia, służące do wykonywania szerokiej gamy zadań, takie jak modele dostarczane przez system ChatGPT. Do przykładów takich programów można zaliczyć BloombergGPT (wyspecjalizowany w finansach) oraz modele LLM dostosowane pod kątem odpowiadania na pytania medyczne (więcej szczegółów w „Dodatku B").

Korzystanie z niestandardowych rozwiązań LLM daje szereg korzyści, zwłaszcza w zakresie prywatności danych. Na przykład ze względu na obawy dotyczące poufności firmy mogą nie chcieć udostępniać wrażliwych danych zewnętrznym dostawcom LLM, takim jak OpenAI. Dodatkowo tworzenie mniejszych, niestandardowych modeli LLM pozwala na ich instalowanie bezpośrednio na urządzeniach klienckich, takich jak laptopy i smartfony. Nad takimi zastosowaniami pracuje obecnie wiele firm, na przykład Apple.

Lokalna implementacja pozwala znacznie zmniejszyć opóźnienia i obniżyć koszty związane z użytkowaniem serwera. Co więcej, niestandardowe modele LLM zapewniają programistom pełną autonomię, co pozwala im w razie potrzeby uzyskać kontrolę nad aktualizacjami i modyfikacjami modelu.

Ogólny proces tworzenia modelu LLM obejmuje szkolenie wstępne i dostrajanie. Słowo „wstępne" w pojęciu „wstępne szkolenie" odnosi się do początkowej fazy, w której model taki jak LLM jest szkolony na dużym, zróżnicowanym zbiorze danych. Celem tej fazy jest rozwinięcie w modelu ogólnego zrozumienia języka. Taki wstępnie przeszkolony model można następnie wykorzystać jako podstawowy zasób i udoskonalać go przez *dostrajanie*. Proces dostrajania polega na specjalistycznym szkoleniu modelu na węższym zestawie danych, który jest dobrany pod kątem określonych zadań lub dziedzin. To dwuetapowe podejście do szkolenia, składające się ze szkolenia wstępnego i dostrajania, jest przedstawione na rysunku 1.3.

Pierwszym krokiem w tworzeniu LLM jest przeszkolenie go na dużym korpusie danych tekstowych, czasami nazywanym *surowym* tekstem. W tym przypadku słowo „surowy" odnosi się do faktu, że dane są zwykłym tekstem bez żadnych informacji pełniących



Rysunek 1.3. Wstępne szkolenie modelu LLM obejmuje przewidywanie następnego słowa na podstawie obszernych zbiorów danych tekstowych. Wstępnie przeszkolony model LLM można następnie dostroić z użyciem mniejszego, oznakowanego zbioru danych

funkcję etykiet (można zastosować filtrowanie, na przykład usuwanie znaków formatowania lub dokumentów w nieznanych językach).

> *UWAGA* Czytelnicy z doświadczeniem w uczeniu maszynowym być może zauważyli, że w przypadku tradycyjnych modeli uczenia maszynowego i głębokich sieci neuronowych szkolonych za pomocą konwencjonalnego paradygmatu uczenia nadzorowanego etykiety zwykle są wymagane. Nie dotyczy to jednak etapu wstępnego szkolenia modeli LLM. W tej fazie modele LLM wykorzystują uczenie samonadzorowane, w którym model samodzielnie generuje etykiety na podstawie danych wejściowych.

Ten pierwszy etap szkolenia modelu LLM jest również znany jako *szkolenie wstępne* (ang. *pretraining*), w którego wyniku powstaje początkowy, wstępnie przeszkolony model LLM, często nazywany modelem bazowym lub podstawowym (ang. *Foundation model*). Typowym przykładem takiego modelu jest GPT-3 (prekursor oryginalnego modelu oferowanego w ChatGPT). Ten model jest zdolny do uzupełniania tekstu — czyli potrafi dokończyć dostarczone przez użytkownika w połowie napisane zdanie. Ma także ograniczone możliwości uczenia się na podstawie niewielu przykładów, co oznacza, że potrafi nauczyć się wykonywać nowe zadania na podstawie zaledwie kilku przykładów i nie wymaga korzystania z obszernych danych szkoleniowych.

Po uzyskaniu po szkoleniu na dużych zbiorach danych tekstowych modelu LLM wstępnie przeszkolonego pod kątem przewidywania następnego słowa w tekście można szkolić model LLM dalej, na danych oznaczonych. Ten etap szkolenia określa się też jako *dostrajanie*.

Dwie najpopularniejsze kategorie dostrajania modeli LLM to *dostrajanie instrukcji* i *dostrajanie klasyfikacji*. W przypadku dostrajania instrukcji oznaczony zbiór danych składa się z par instrukcji i odpowiedzi, takich jak zapytanie o przetłumaczenie

tekstu razem z poprawnie przetłumaczonym tekstem. W przypadku dostrajania klasyfikacji oznaczony zbiór danych składa się z tekstów i powiązanych etykiet klas — na przykład wiadomości e-mail powiązanych z etykietami „spam" i „nie spam".

W dalszej części książki omówię implementację wstępnego szkolenia i dostrajania modeli LLM, a także zaprezentuję specyfikę dostrajania zarówno instrukcji, jak i klasyfikacji po wstępnym przeszkoleniu podstawowego modelu LLM.

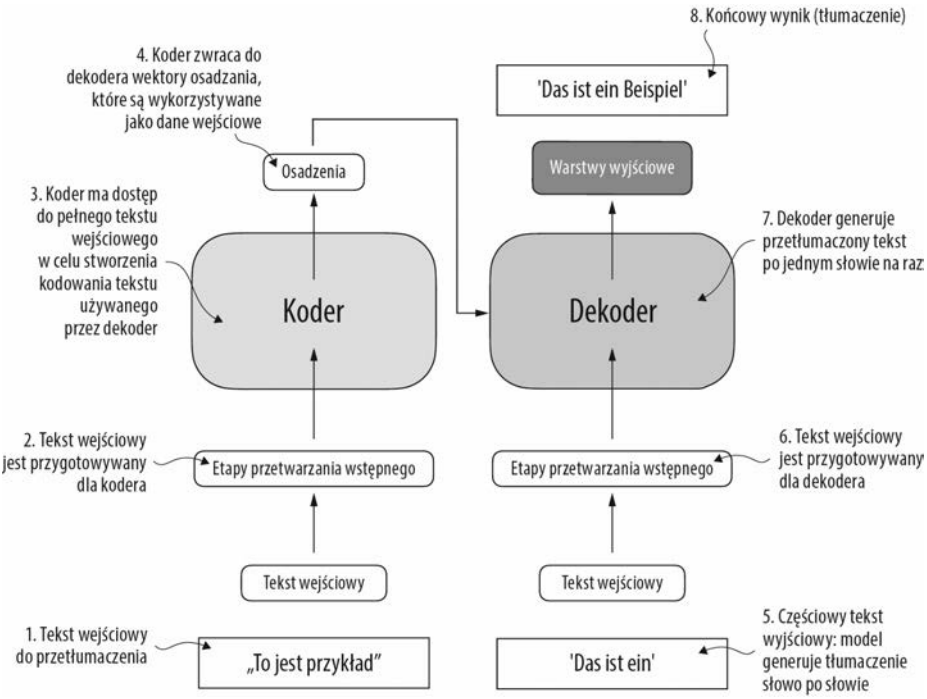
1.4. *Wprowadzenie do architektury transformerów*

Większość nowoczesnych modeli LLM opiera się na architekturze *Original Transformer*, czyli głębokiej sieci neuronowej przedstawionej w 2017 roku w artykule *Attention Is All You Need (https://arxiv.org/abs/1706.03762)*. Aby zrozumieć modele LLM, trzeba zapoznać się z tą architekturą, którą opracowano na potrzeby tłumaczenia maszynowego tekstów w języku angielskim na język niemiecki i francuski. Uproszczona wersja architektury transformera jest pokazana na rysunku 1.4.

Architektura transformera składa się z dwóch podmodułów: kodera i dekodera. Moduł kodera przetwarza tekst wejściowy i koduje go do postaci ciągu reprezentacji liczbowych lub wektorów opisujących kontekst danych wejściowych. Następnie moduł dekodera pobiera zakodowane wektory i generuje tekst wyjściowy. Na przykład w zadaniu tłumaczenia koder koduje tekst z języka źródłowego do postaci wektorowej, a dekoder dekoduje te wektory w celu wygenerowania tekstu w języku docelowym. Zarówno koder, jak i dekoder składają się z wielu warstw połączonych tak zwanym *mechanizmem samouwagi* (ang. *self-attention mechanism*). Na temat sposobu wstępnego przetwarzania i kodowania danych wejściowych można zadać wiele pytań. Odpowiem na nie krok po kroku w kolejnych rozdziałach.

Kluczowym komponentem transformerów i modeli LLM jest mechanizm samouwagi (na rysunku 1.4 go nie pokazano), który dostarcza modelowi wagi znaczenia różnych słów lub tokenów w sekwencji względem siebie. Dzięki temu mechanizmowi model może uchwycić w danych wejściowych odległe zależności i relacje kontekstowe, co zwiększa jego zdolności do generowania spójnych i kontekstowo istotnych danych wyjściowych. Ze względu na jego złożoność dokładniejszy opis jego działania odłożę do rozdziału 3., w którym go omówię i krok po kroku zaimplementuję.

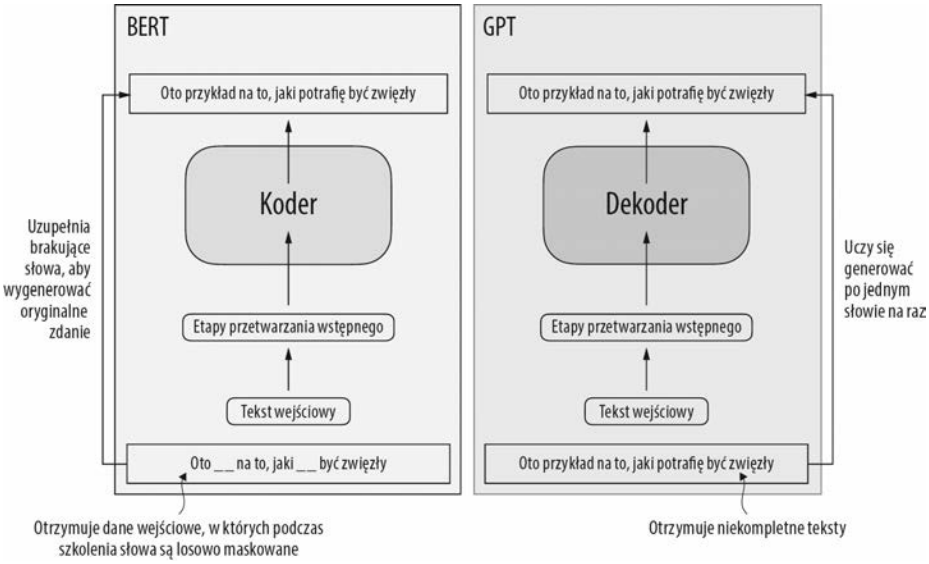
Na tym pojęciu opierały się późniejsze odmiany architektury transformerów, takie jak BERT (skrót od *bidirectional encoder representations from transformers* — dosłownie: dwukierunkowe reprezentacje kodera z transformerów) i różne modele GPT (skrót od *generative pretrained transformers* — dosłownie: generatywne, wstępnie przeszkolone transformery). Opracowanie tych transformerów pozwoliło dostosować architekturę transformera do wykonywania różnych zadań. Jeśli jesteś zainteresowany, zapoznaj się z „Dodatkiem B", w którym znajdziesz sugestie dotyczące lektury uzupełniającej.



Rysunek 1.4. Uproszczona reprezentacja architektury Original Transformer, będącej modelem uczenia głębokiego do tłumaczeń językowych. Transformer składa się z dwóch części: (a) kodera, który przetwarza tekst wejściowy i tworzy reprezentację osadzeń tekstu (reprezentację numeryczną wielu różnych czynników w różnych wymiarach), którą (b) dekoder może wykorzystać w celu wygenerowania przetłumaczonego tekstu po jednym słowie na raz. Ten rysunek przedstawia końcowy etap procesu tłumaczenia, w którym dekoder, aby wykonać tłumaczenie oryginalnego tekstu wejściowego („This is an example"), musi wygenerować tylko słowo końcowe („Beispiel") i częściowo przetłumaczone zdanie („Das ist ein")

Model BERT, który jest zbudowany na podstawie podmodułu kodera architektury *Original Transformer*, różni się od modeli GPT podejściem do szkolenia. O ile GPT jest przeznaczony do zadań generatywnych, o tyle BERT i jego odmiany specjalizują się w przewidywaniu zamaskowanych słów (model przewiduje zamaskowane lub ukryte słowa w danym zdaniu — rysunek 1.5). Dzięki tej unikatowej strategii szkoleniowej model BERT sprawdza się w zadaniach klasyfikacji tekstu, w tym zadaniach oznaczania tonu i kategoryzacji dokumentów. W chwili gdy piszę te słowa, modelu BERT używa się do wykrywania toksycznych treści na platformie X (dawniej Twitter).

Z drugiej strony model GPT skupia się na części dekodera architektury *Original Transformer* i jest przeznaczony do zadań wymagających generowania tekstów. Obejmuje to tłumaczenie maszynowe, streszczanie tekstów, pisanie beletrystyki, pisanie kodu komputerowego i wiele innych.

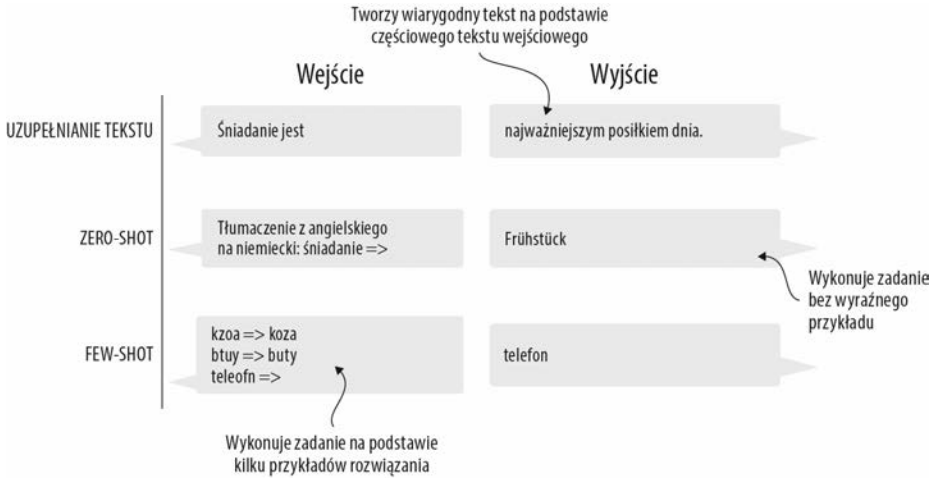


Rysunek 1.5. Wizualna reprezentacja podmodułów transformera: kodera i dekodera. Po lewej: segment kodera jest przykładem modelu LLM podobnego do BERT, który koncentruje się na prognozowaniu zamaskowanych słów i jest używany głównie do takich zadań, jak klasyfikacja tekstu. Po prawej: segment dekodera prezentuje modele LLM podobne do GPT, zaprojektowane do zadań generatywnych i tworzenia spójnych sekwencji tekstowych

Modele GPT, zaprojektowane i przeszkolone głównie do wykonywania zadań uzupełniania tekstu, również wykazują niezwykłą wszechstronność. Modele te doskonale sprawdzają się w wykonywaniu zarówno zadań typu *zero-shot* (niewymagających podawania przykładów rozwiązań), jak i *few-shot* (wymagających podania zaledwie kilku przykładów rozwiązań). Pojęcie *zero-shot* odnosi się do zdolności uogólniania modelu na zadania, których model nigdy nie widział (tzn. nigdy wcześniej nie przekazano mu żadnych konkretnych przykładów rozwiązania tego zadania). Z drugiej strony uczenie typu *few-shot* polega na uczeniu się na podstawie minimalnej liczby przykładów, które użytkownik podaje jako dane wejściowe (rysunek 1.6).

1.5. *Wykorzystanie dużych zbiorów danych*

Duże zbiory danych szkoleniowych dla popularnych modeli GPT i BERT reprezentują różnorodne i obszerne korpusy tekstowe, obejmujące miliardy słów i dotyczące szerokiego zakresu tematów oraz języków naturalnych i komputerowych. W ramach konkretnego przykładu w tabeli 1.1 zestawiłem zbiory danych wykorzystane do wstępnego szkolenia modelu GPT-3 — bazowego modelu dla pierwszej wersji ChatGPT.



Rysunek 1.6. Modele LLM podobne do GPT oprócz uzupełniania tekstu potrafią na podstawie danych wejściowych rozwiązywać różne zadania bez konieczności ponownego szkolenia, dostrajania ani zmian architektury modelu specyficznych dla zadania. Czasami w ramach danych wejściowych przydaje się podanie przykładów celu. Takie zadania określa się terminem few-shot. Jednak modele LLM podobne do GPT są również zdolne do wykonywania zadań bez konkretnego przykładu, co nazywa się konfiguracją zero-shot

Transformery kontra modele LLM

Współczesne modele LLM są oparte na architekturze transformera. Z tego względu pojęcia transformer i model LLM często są używane w literaturze jako synonimy. Należy jednak pamiętać, że nie wszystkie transformery są modelami LLM. Transformery można także wykorzystywać w zadaniach widzenia komputerowego. Ponadto nie wszystkie modele LLM są transformerami — istnieją modele LLM oparte na architekturach sieci rekurencyjnych i konwolucyjnych. Główną motywacją stojącą za tymi alternatywnymi podejściami jest poprawa wydajności obliczeniowej LLM. To, czy te alternatywne architektury LLM mogą konkurować z możliwościami modeli LLM opartych na transformerach i czy ostatecznie się przyjmą, dopiero się okaże. Dla uproszczenia terminu „LLM" będę używać w odniesieniu do podobnych do GPT modeli LLM opartych na transformerach (zainteresowani Czytelnicy mogą sięgnąć do „Dodatku B", w którym zamieściłem odnośniki do zasobów opisujących te architektury).

W tabeli 1.1 zestawiono liczbę tokenów, przy czym token rozumie się jako jednostkę tekstu odczytywaną przez model, a liczba tokenów w zbiorze danych jest w przybliżeniu równoważna liczbie słów i znaków interpunkcyjnych w tekście. Proces tokenizacji, czyli przekształcania tekstu na tokeny, opisałem w rozdziale 2.

Dzięki ogromnej skali i różnorodności zestawu danych modele zyskują zdolność radzenia sobie z wieloma różnorodnymi wyzwaniami. Potrafią analizować składnię, rozumieją znaczenie i kontekst języka, a nawet potrafią sprostać zadaniom wymagającym szerszej wiedzy o świecie — to najważniejszy wniosek, jaki można wyciągnąć z danych zaprezentowanych w tej tabeli.

Tabela 1.1. Zbiór danych użyty do wstępnego szkolenia popularnego modelu LLM GPT-3

Nazwa zbioru danych	Opis zbioru danych	Liczba tokenów	Proporcja w danych szkoleniowych
CommonCrawl (filtrowany)	Dane indeksowania stron internetowych	410 miliardów	60%
WebText2	Dane indeksowania stron internetowych	19 miliardów	22%
Books1	Internetowy korpus książek	12 miliardów	8%
Books2	Internetowy korpus książek	55 miliardów	8%
Wikipedia	Tekst wysokiej jakości	3 miliardy	3%

Szczegóły zbioru danych modelu GPT-3

W tabeli 1.1 zaprezentowałem zbiór danych wykorzystany w modelu GPT-3. Kolumna proporcji w tabeli sumuje się do 100% próbkowanych danych (w granicach błędu wynikającego z zaokrągleń). Chociaż podzbiory w kolumnie **Liczba tokenów** obejmują łącznie 499 miliardów, model przeszkolono tylko na 300 miliardach tokenów. Autorzy artykułu poświęconego modelowi GPT-3 nie sprecyzowali, dlaczego model nie został przeszkolony na wszystkich 499 miliardach tokenów.

Aby zrozumieć kontekst, rozważmy rozmiar zbioru danych *CommonCrawl*, który sam w sobie składa się z 410 miliardów tokenów i wymaga około 570 GB pamięci. Dla porównania, w późniejszych iteracjach modeli podobnych do GPT-3, takich jak LLaMA firmy Meta, rozszerzono zakres szkolenia o dodatkowe źródła danych, na przykład o artykuły naukowe w serwisie Arxiv (92 GB), a także pytania i odpowiedzi związane z kodem z serwisu StackExchange (78 GB).

Autorzy artykułu poświęconego modelowi GPT-3 nie udostępnili szkoleniowego zbioru danych, ale porównywalnym zbiorem, który jest publicznie dostępny, jest zbiór udostępniony w pracy Soldaini et al. 2024, *Dolma: An Open Corpus of Three Trillion Tokens for LLM Pretraining Research* (*https://arxiv.org/abs/2402.00159*). Ten zbiór może jednak zawierać utwory chronione prawem autorskim, a dokładne warunki korzystania mogą zależeć od zamierzonego przypadku użycia i kraju.

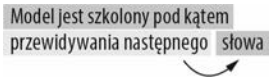
Modele, które wstępnie przeszkolono, są niezwykle wszechstronne i łatwo można je dostosować do różnych zadań. Z tego względu nazywa się je modelami bazowymi lub podstawowymi. Wstępne szkolenie modeli LLM wymaga dostępu do znacznych zasobów i jest bardzo kosztowne. Przykładowo koszt wstępnego szkolenia modelu GPT-3 z uwzględnieniem cen usług chmurowych oszacowano na 4,6 miliona dolarów (*https://mng.bz/VxEW*).

Na szczęście do pisania, wyodrębniania i edytowania tekstów, które nie były częścią danych szkoleniowych, można wykorzystać narzędzia ogólnego przeznaczenia oraz wiele wstępnie przeszkolonych modeli LLM typu *open source*. Ponadto modele LLM można precyzyjnie dostroić do określonych zadań z użyciem znacząco mniejszych zbiorów danych, co zmniejsza ilość potrzebnych zasobów obliczeniowych i poprawia wydajność.

W dalszej części książki zaprezentuję przeznaczony do wstępnego szkolenia modelu LLM kod, z którego skorzystamy w celach edukacyjnych. Wszystkie obliczenia można wykonać na sprzęcie konsumenckim. Po opracowaniu kodu wstępnego szkolenia nauczysz się wykorzystywać dostępne za darmo wagi modeli i ładować je do tworzonej architektury, co pozwoli pominąć kosztowny etap wstępnego szkolenia podczas dostrajania modelu LLM.

1.6. Szczegóły architektury modeli GPT

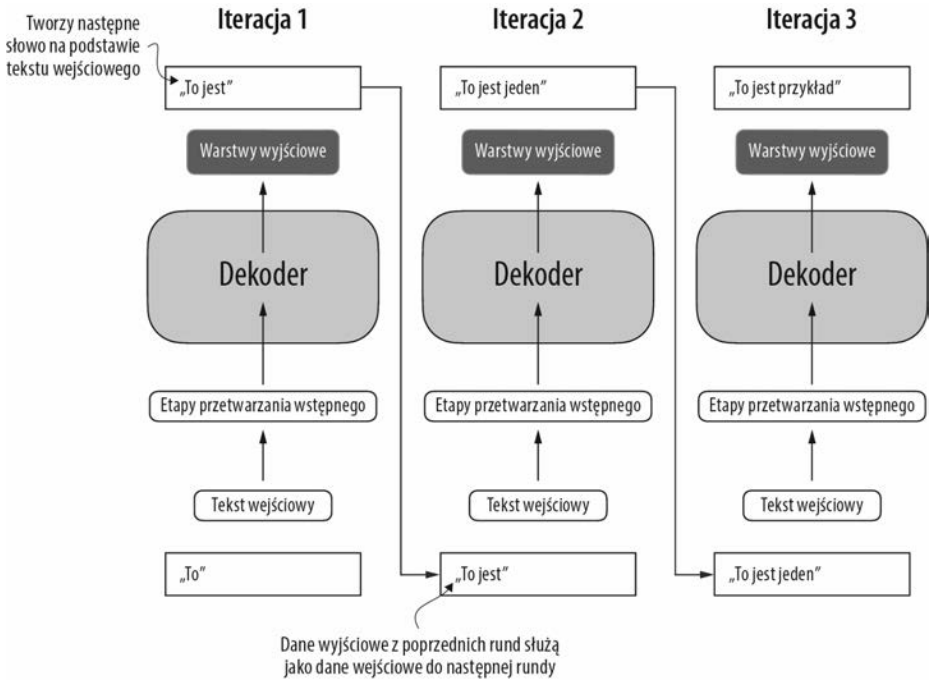
Model GPT po raz pierwszy przedstawiono w artykule *Improving Language Understanding by Generative Pre-Training* (*https://mng.bz/x2qg*), opracowanym przez zespół Radford et al. z firmy OpenAI. GPT-3 to skalowana wersja tego modelu, która ma więcej parametrów i została przeszkolona na większym zbiorze danych. Ponadto oryginalny model oferowany w systemie ChatGPT stworzono przez dostrojenie modelu GPT-3 na dużym zbiorze danych z użyciem metody opisanej w opublikowanym przez firmę OpenAI artykule *InstructGPT* (*https://arxiv.org/abs/2203.02155*). Jak pokazałem na rysunku 1.6, modele te są zdolne do wykonywania zadań uzupełniania tekstu, a także innych, takich jak korekta pisowni, klasyfikacja i tłumaczenia. To niezwykłe, zważywszy na to, że modele GPT wstępnie przeszkolono na stosunkowo prostym zadaniu przewidywania następnego słowa (co pokazałem na rysunku 1.7).



Rysunek 1.7. W zadaniu wstępnego szkolenia modeli GPT w przewidywaniu następnego słowa system uczy się przewidywać kolejne słowo w zdaniu na podstawie słów, które pojawiły się przed nim. Takie podejście pomaga modelowi zrozumieć, w jaki sposób w języku zazwyczaj pasują do siebie słowa i frazy. To jest podstawa, którą można zastosować do różnych innych zadań

Zadanie przewidywania następnego słowa jest formą uczenia samonadzorowanego, które jest rodzajem samodzielnego znakowania. Oznacza to, że nie ma potrzeby jawnego wybierania etykiet dla danych szkoleniowych, a do tego celu można wykorzystać strukturę samych danych. Następne słowo w zdaniu lub dokumencie może posłużyć jako etykieta, którą model powinien przewidzieć. Ponieważ zadanie przewidywania następnego słowa pozwala tworzyć etykiety „w locie", do szkolenia modeli LLM można użyć ogromnej liczby nieoznakowanych zbiorów danych tekstowych.

W porównaniu z architekturą *Original Transformer*, którą omówiłem w podrozdziale 1.4, ogólna architektura modelu GPT jest stosunkowo prosta. W istocie jest to sam dekoder, bez kodera (rysunek 1.8). Ponieważ modele typu „sam dekoder", takie jak GPT, generują tekst przez przewidywanie tekstu po jednym słowie na raz, uważa się je za rodzaj modelu *autoregresyjnego*. Modele autoregresyjne wykorzystują swoje



Rysunek 1.8. Architektura GPT wykorzystuje tylko część dekodera z architektury *Original Transformer*. Zaprojektowano ją do przetwarzania jednokierunkowego, od lewej do prawej, dzięki czemu dobrze nadaje się do generowania tekstu i zadań przewidywania następnego słowa w celu generowania tekstu w sposób iteracyjny, po jednym słowie na raz

poprzednie wyniki jako dane wejściowe do przyszłych prognoz. W rezultacie w modelu GPT każde nowe słowo jest wybierane na podstawie poprzedzającej je sekwencji, co poprawia spójność wynikowego tekstu.

Takie architektury jak GPT-3 są również znacznie obszerniejsze od modelu *Original Transformer*. Na przykład w architekturze *Original Transformer* bloki kodera i dekodera były powielone sześciokrotnie. Model GPT-3 ma 96 warstw transformerów i łącznie 175 miliardów parametrów.

Model GPT-3 wprowadzono w 2020 roku, co według standardów uczenia głębokiego i rozwoju modeli LLM uznaje się za odległą przeszłość. Jednak nowsze architektury, takie jak modele Llama firmy Meta, nadal opierają się na tych samych podstawowych pojęciach i wprowadzono w nich jedynie niewielkie modyfikacje. W związku z tym zrozumienie modeli GPT nadal jest bardzo istotne. Z tego powodu skupię się na zaimplementowaniu istotnej architektury modelu GPT, a jednocześnie wskażę konkretne poprawki zastosowane w alternatywnych modelach LLM.

Chociaż model *Original Transformer*, składający się z bloków kodera i dekodera, pierwotnie zaprojektowano do tłumaczenia języków, modele GPT — pomimo ich obszerniejszej, ale prostszej architektury „tylko dekoder", której celem jest przewidywanie

następnego słowa — również są zdolne do wykonywania zadań tłumaczeniowych. Ta zdolność była początkowo zaskoczeniem dla badaczy, ponieważ wyłoniła się z modelu przeszkolonego głównie w przewidywaniu następnego słowa, czyli zadaniu, które nie było specjalnie ukierunkowane na tłumaczenie.

Zdolność modelu do wykonywania zadań, do których specjalnie go nie szkolono, nazywa się *zachowaniem emergentnym*. Zdolność ta nie jest wyraźnie nauczana podczas szkolenia, ale pojawia się jako naturalna konsekwencja ekspozycji modelu na ogromne ilości wielojęzycznych danych w różnych kontekstach. Fakt, że modele GPT mogą „uczyć się" wzorców tłumaczeń między językami i wykonywać zadania tłumaczeniowe, nawet jeśli nie zostały do tego specjalnie przeszkolone, pokazuje korzyści i możliwości tych wielkoskalowych, generatywnych modeli językowych. Za ich pomocą można wykonywać różne zadania bez konieczności używania różnych modeli do różnych typów zadań.

1.7. Tworzenie dużego modelu językowego

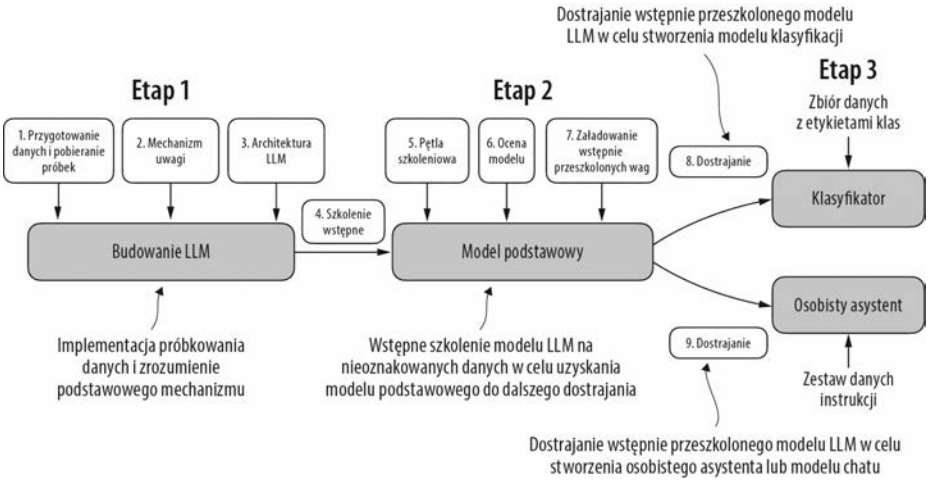
Po zaprezentowaniu podstaw dotyczących modeli LLM spróbujemy zakodować taki model od początku. Przyjmiemy za plan fundamentalną ideę modelu GPT i zrealizujemy implementację modelu w trzech etapach (rysunek 1.9).

Podczas etapu 1. zapoznam Czytelnika z podstawowymi krokami wstępnego przetwarzania danych i sposobem kodowania mechanizmu uwagi, będącego sercem każdego modelu LLM. Następnie, w etapie 2., pokażę, jak zakodować i wstępnie przeszkolić model LLM podobny do GPT, zdolny do generowania nowych tekstów. Omówię również podstawy oceny modeli LLM, która jest niezbędna do tworzenia wydajnych systemów NLP.

Wstępne szkolenie modelu LLM od podstaw to znaczące przedsięwzięcie, z którym w przypadku modeli podobnych do GPT wiążą się koszty od kilku tysięcy do wielu milionów dolarów. Z tego powodu w etapie 2. skoncentruję się na implementacji szkolenia dla celów edukacyjnych, z użyciem niewielkiego zbioru danych. Ponadto udostępnię przykłady kodu umożliwiające załadowanie ogólnodostępnych wag modeli.

Na koniec, w trakcie etapu 3., pokażę, jak dostroić wstępnie przeszkolony model LLM do wykonywania takich instrukcji jak odpowiadanie na pytania lub klasyfikowanie tekstów — czyli najczęstszych zadań w wielu rzeczywistych aplikacjach i badaniach.

Mam nadzieję, że z niecierpliwością czekasz, aż wyruszymy w tę ekscytującą podróż!



Rysunek 1.9. Trzy główne etapy kodowania modelu LLM to implementacja architektury LLM i proces przygotowania danych (etap 1.), wstępne szkolenie modelu LLM w celu utworzenia modelu podstawowego (etap 2.) oraz dostrajanie modelu podstawowego, aby stał się osobistym asystentem lub klasyfikatorem tekstu (etap 3.)

Podsumowanie

- Modele LLM znacząco zmieniły dziedzinę przetwarzania języka naturalnego, w której wcześniej wykorzystywano głównie systemy oparte na jawnych regułach i prostszych metodach statystycznych. Wraz z modelami LLM pojawiły się nowe podejścia oparte na uczeniu głębokim, co pozwoliło osiągnąć postępy w zakresie rozumienia, generowania i tłumaczenia ludzkiego języka.
- Szkolenie współczesnych modeli LLM przebiega w dwóch głównych etapach:
  - Po pierwsze, wstępnie szkoli się je na dużym korpusie nieoznakowanego tekstu, z wykorzystaniem w roli etykiety prognozy następnego słowa w zdaniu.
  - Następnie dostraja się je na docelowym, mniejszym i oznakowanym, zbiorze danych pod kątem postępowania według instrukcji lub wykonywania zadań klasyfikacji.
- Modele LLM są oparte na architekturze transformera. Kluczowym komponentem architektury transformera jest mechanizm uwagi, który podczas generowania wyjścia po jednym słowie na raz daje modelowi LLM selektywny dostęp do całej sekwencji wejściowej.
- Architektura *Original Transformer* składa się z kodera do parsowania tekstu i dekodera do generowania tekstu.

- Modele LLM do generowania tekstu i wykonywania instrukcji, takie jak GPT-3 i ChatGPT, implementują tylko moduły dekodera, co upraszcza architekturę.
- Do wstępnego szkolenia modeli LLM niezbędne są duże zbiory danych, składające się z miliardów słów.
- Podczas gdy ogólnym zadaniem szkolenia wstępnego dla modeli podobnych do GPT jest przewidywanie następnego słowa w zdaniu, te modele LLM wykazują właściwości emergentne, takie jak zdolność do klasyfikowania, tłumaczenia lub tworzenia streszczeń dokumentów.
- W wyniku wstępnego przeszkolenia modelu LLM powstaje model podstawowy, który można bardziej efektywnie dostroić do różnych zadań końcowych.
- Modele LLM precyzyjnie dostrojone na niestandardowych zestawach danych mogą w określonych zadaniach przewyższać wydajnością uniwersalne modele LLM.

Najbardziej przystępne i kompleksowe wyjaśnienie modeli językowych!

Cameron Wolfe, starszy specjalista do spraw AI, Netflix

Duże modele językowe przestały już być szokującą nowinką technologiczną. Dziś są używane do najróżniejszych celów, a lista ich rzeczywistych i potencjalnych zastosowań stale się wydłuża. A to oznacza, że programiści, inżynierowie i architekci muszą dogłębnie rozumieć zasady działania LLM, a także techniki ich budowania.

Kompletne, aktualne opracowanie. Gorąco polecam!

dr Vahid Mirjalili, starszy danolog, FM Global

W tej unikalnej książce znajdziesz kompleksowe omówienie procesu tworzenia LLM, od pracy z zestawami danych po implementację architektury modelu, wstępne szkolenie na nieoznakowanych danych i dostrajanie do określonych zadań. Bez korzystania z gotowych bibliotek LLM samodzielnie zbudujesz podstawowy model, przekształcisz go w klasyfikator tekstu, a ostatecznie stworzysz chatbota, który będzie wykonywał Twoje polecenia. I co najważniejsze — naprawdę zrozumiesz, jak działa model, w końcu będziesz jego twórcą!

Niezwykle inspirująca pozycja!

Benjamin Muskalla, starszy inżynier, GitHub

Z tą książką:

- **zaprojektujesz i zbudujesz funkcjonujący model LLM**
- **nauczysz się korzystać ze wstępnie wyuczonych wag**
- **skonstruujesz kompletny potok szkoleniowy**
- **dostosujesz model LLM do zadań klasyfikacji tekstu**
- **stworzysz model LLM zdolny do wykonywania przekazywanych mu instrukcji**

Zbuduj AI — niech przemówi Twoim kodem!

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 978-83-289-2497-0

Cena: 99,00 zł

Dr Sebastian Raschka

Jest badaczem i autorem bestsellerowych książek. Pracuje w Lightning AI, gdzie implementuje i szkoli modele LLM. Wcześniej był adiunktem na University of Wisconsin-Madison, zajmował się między innymi badaniami nad uczeniem głębokim. Jest znany z praktycznego podejścia i klarownego wyjaśniania zaawansowanych koncepcji inżynierii.