Metody wstępnego trenowania LLM

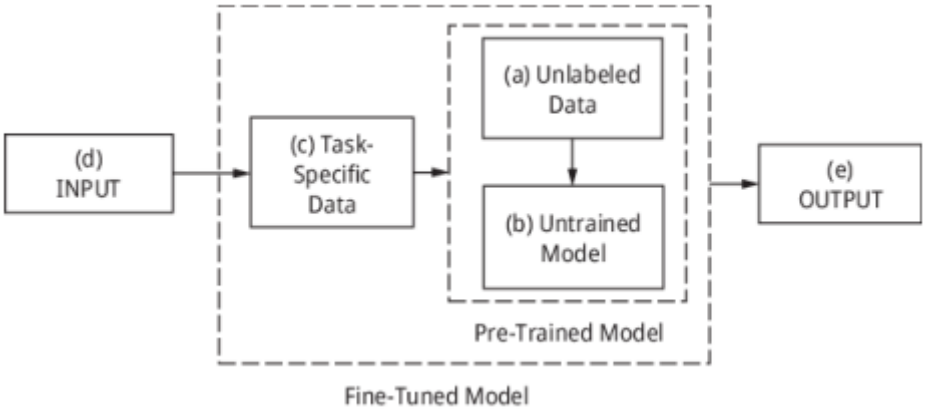
Streszczenie

Generatywna sztuczna inteligencja (AI), technika AI, generuje oryginalny tekst, dźwięki, modele 3D, animacje i obrazy. Jest ona oparta na modelach uczenia maszynowego na dużą skalę (ML), które wykorzystują wstępnie wytrenowane głębokie sieci neuronowe na ogromnych zbiorach danych. Wstępne trenowanie ma na celu głównie odgadnięcie kolejnego słowa w zdaniu lub uzupełnienie zamaskowanych słów w sekwencji. Poprzez to ćwiczenie uczenia się bez nadzoru, model uczy się rozumieć struktury językowe i trendy statystyczne. Dzięki wstępnemu trenowaniu, model dużego języka (LLM) nabywa ogólne zrozumienie składni, gramatyki i semantyki. Pomaga to w budowaniu solidnych podstaw do rozumienia języka i umożliwia modelowi przedstawianie relacji między słowami. Wstępne trenowanie danych jest przede wszystkim odpowiedzialne za niezwykłe możliwości LLM. Jest to podobne do bombardowania mózgu ogromnymi ilościami danych, aby nauczył się on praw języka i świata zewnętrznego, zanim nauczy się określonych umiejętności. Tu przewidywanie kolejnych zdań, uczenie kontrastywne, maskowane modelowanie języka, cele na poziomie zdań i dokumentów, autoenkodery odszumiające (DAE) oraz inne metody wstępnego trenowania LLM, wraz z ich znaczeniem. Wstępnie wytrenowane modele zwiększają wydajność i zmniejszają potrzebę dodatkowego trenowania, umożliwiając zastosowanie informacji w kolejnych zadaniach. Wstępne trenowanie danych ma szerszy zakres obszarów zastosowań, takich jak transfer uczenia, klasyfikacja i ekstrakcja cech.

Słowa kluczowe: generatywna sztuczna inteligencja (AI), wstępne trenowanie, duży model języka (LLM), uczenie maszynowe (ML), uczenie nadzorowane i nienadzorowane.

Wprowadzenie

W przetwarzaniu języka naturalnego (NLP) duże modele językowe (LLM) to modele które zostały wytrenowane na ogromnych ilościach danych tekstowych bez uczenia się wykonywania konkretnych zadań; proces ten znany jest jako „wstępne trenowanie". Koncepcja tego modelu polega na przetwarzaniu ogromnych ilości danych i wydobywaniu z nich wiedzy ogólnej. Nie ma to miejsca w przypadku konwencjonalnych metod uczenia maszynowego (ML), w których model jest często trenowany. Programista pozwala modelowi jedynie na pozyskanie pewnych ogólnych informacji podczas „wstępnego trenowania", po czym można go wytrenować do wykonywania określonych zadań. „Dostrajanie" to ostatnia faza, w której model LLM uczy się wykonywania określonych zadań. Jedna z metod głębokiego uczenia się, zwana generatywnym wstępnym trenowaniem (GPT), polega na nienadzorowanym trenowaniu modelu języka na obszernym korpusie danych tekstowych. Głównym celem GPT jest przewidywanie kolejnego słowa w danej sekwencji i generowanie tekstu, który ściśle naśladuje tekst tworzony przez ludzi. Działania NLP, w tym generowanie tekstu, tłumaczenie, podsumowywanie i analiza sentymentu, szeroko wykorzystują modele GPT, które oferują dokładne odpowiedzi na pytania bez żadnych niepożądanych informacji. Wstępne trenowanie jest na ogół jak wprowadzenie dużej ilości danych do systemu i nauczenie go formy przepływu tekstu, gramatyki i tworzenia zdań. Po zakończeniu wstępnego trenowania do systemu zostaną wprowadzone drobne poprawki określonych danych, tak aby system nauczył się formułować GPT, aby idealnie odpowiadać na zapytania użytkownika. Ta metoda jest sformalizowana na Rysunku



Niewytrenowany model, pokazany jako (b) na rysunku, otrzymuje znaczną ilość danych wejściowych (na przykład książek), oznaczonych jako nieoznaczone dane (a) na diagramie. To kończy fazę „wstępnego trenowania" i tworzy LLM. Następnie chcemy przypisać określone zadanie (pokazane jako (c) na rysunku) do LLM. W rezultacie, obecnie posiadany jest „dopracowany" model, który jest w stanie wykonać zadanie zlecone mu przez użytkownika. To jest moment, w którym system może wygenerować dopracowane instrukcje lub odpowiedzi na zapytanie użytkownika. Podobnie, (d) i (e) pokazują, że po dopracowaniu modelu można wprowadzić dane wejściowe (przykładowe zapytanie użytkownika), a on wygeneruje dane wyjściowe (zdefiniowaną odpowiedź na zapytanie użytkownika). W tym momencie można się zastanawiać, czym ta metoda różni się od innych procedur uczenia maszynowego. Aby omówić ten temat, przedstawiono przegląd ogólnych podejść do uczenia maszynowego powszechnie stosowanych w uczeniu maszynowym. Istnieją trzy rodzaje metod uczenia maszynowego [4], a mianowicie (1) uczenie nadzorowane, (2) uczenie nienadzorowane i (3) uczenie samonadzorowane, z których każde opisano poniżej:

1. Uczenie nadzorowane: W tej procedurze wykorzystywane są dobrze oznaczone dane szkoleniowe, tj. oznaczone dane będą używane do trenowania modelu we wszystkich aspektach. Na przykład system może mieć źródło w języku źródłowym i jego tłumaczenie na język docelowy. Oznaczone dane mogą się różnić w zależności od zadania, do wykonania którego trenowany jest model uczenia maszynowego.

2. Uczenie nienadzorowane: Dane szkoleniowe wykorzystywane w tej procedurze nie są oznaczone. W sztucznej inteligencji uczenie maszynowe, które odbywa się bez nadzoru człowieka, jest znane jako uczenie nienadzorowane. Modele uczenia maszynowego bez nadzoru, w przeciwieństwie do uczenia nadzorowanego, otrzymują nieoznakowane dane i pozwalają im samodzielnie, tj. bez wyraźnego kierownictwa lub instrukcji, wyszukiwać wzorce i spostrzeżenia.

3. Uczenie samonadzorowane: W tym podejściu etykiety dla danych treningowych są generowane automatycznie przez system, który łączy techniki nadzorowane i nienadzorowane. W uczeniu maszynowym metoda ta stanowi paradygmat, w którym model jest trenowany w ramach zadania i zamiast polegać na zewnętrznych etykietach dostarczanych przez ludzi, wykorzystuje same dane do generowania sygnałów nadzorczych.

Użytkownik może wykorzystać duże ilości nieoznakowanych danych (podobnie jak w przypadku treningu bez nadzoru) na etapie „wstępnego treningu" w celu utworzenia modelu LLM, a następnie mniejszą ilość oznaczonych danych (podobnie jak w przypadku treningu z nadzorem) na etapie „dostrajania" w celu stworzenia finalnego, dostrojonego modelu, który jest faktycznie zdolny do wykonywania zadań. Jak widać z podstawowego opisu modeli LLM, to podejście różni się od tradycyjnych systemów uczenia maszynowego pod wieloma względami. W porównaniu z tworzeniem

od podstaw, wstępnie wytrenowany model ma tę zaletę, że pozwala szybciej identyfikować parametry, które prawdopodobnie zapewnią pozytywne wyniki. Modele wstępnie wytrenowane mają również tę zaletę, że wymagają mniej danych niż konwencjonalne modele tworzone od podstaw.

Smart Factory

LLM to program komputerowy trenowany na ogromnych ilościach danych tekstowych, który wykorzystuje architekturę transformatorową do uczenia się i generowania języka podobnego do języka ludzkiego. Modele LLM to fundamentalne modele uczenia maszynowego, które przetwarzają i rozumieją język naturalny za pomocą metod głębokiego uczenia. Do trenowania tych modeli wykorzystuje się duże ilości danych tekstowych, aby mogły one rozpoznawać wzorce językowe i relacje między encjami. Maszyny uczące się języków potrafią wykonywać szeroki zakres zadań językowych, w tym analizę sentymentu, tłumaczenie języków, dialogi w chatbotach i wiele innych. Potrafią one rozumieć zawiły materiał tekstowy, rozpoznawać rzeczy i powiązania między nimi oraz tworzyć nowe, poprawne gramatycznie i spójne treści. Dwa główne powody wyboru modeli LLM zamiast budowania modelu ML od podstaw do wykonywania zadań w NLP to:

Dostępność danych

Jedną z najbardziej obiecujących architektur w dziedzinie NLP są sieci neuronowe, a w szczególności modele głębokiego uczenia, które obejmują wiele ukrytych warstw przed wygenerowaniem wyniku. Modele te wymagają dużej ilości oznaczonych danych treningowych, aby zapobiec nadmiernemu dopasowaniu podczas treningu, ponieważ mają wiele parametrów i dlatego są reprezentowane jako „duże" modele językowe [7]. Ponieważ trening nadzorowany wymaga interakcji z człowiekiem, a dane treningowe z etykietami są trudniejsze do uzyskania niż dane nieoznakowane, jest on dość kosztowny w dziedzinie przetwarzania języka naturalnego (NLP). Na przykład, gdy konieczne jest zaprojektowanie systemu tłumaczeniowego, wydaje się to droższe, ponieważ wszystkie zdania w danym języku muszą zostać zebrane, a następnie potrzebny jest specjalista do ich przetłumaczenia. Z drugiej strony, dane nieoznakowane są dostępne w całym internecie, podobnie jak książki i artykuły na blogach w jednym konkretnym języku, ale nie w wersji przetłumaczonej. Zamiast opracowywać modele uczenia maszynowego specjalnie dla zadań podrzędnych, takich jak tłumaczenie maszynowe, naukowcy zajmujący się NLP postanowili podzielić proces treningu na dwie fazy: etap „wstępnego treningu" z wykorzystaniem nieoznakowanych danych monolingwalnych (tj. uczenie nienadzorowane) oraz etap „dostrajania" z wykorzystaniem znacznie mniejszych danych oznaczonych (tj. uczenie nadzorowane) dla różnych zadań podrzędnych.

Skalowalność

Aby poprawić zdolność modelu LLM do uczenia się i generalizacji, skalowanie modelu polega na jego powiększeniu poprzez dodanie większej liczby parametrów. Ta metoda oferuje bardziej modułową konstrukcję, ponieważ dzieli proces uczenia na dwa etapy: wstępne uczenie i dostrajanie. W rezultacie model LLM może być wykorzystywany do wielu dalszych operacji po jego wygenerowaniu. Innymi słowy, można wygenerować kilka modeli, dostrajając pojedynczy model LLM przy użyciu danych oznaczonych dla każdego dalszego zadania niezależnie. Na przykład, jeden model LLM można zoptymalizować pod kątem jego dalszych zadań klasyfikacji intencji, a następnie można go zoptymalizować pod kątem kolejnego dalszego zadania identyfikacji nazwanych jednostek.

Zalety wstępnego trenowania w LLM

Wstępne trenowanie danych w dużych modelach językowych oferuje szeroki wachlarz korzyści [8] w różnych dziedzinach i zastosowaniach. Oto kilka kluczowych korzyści:

1. Wydajność i automatyzacja:

Wstępne trenowanie LLM może zautomatyzować analizę danych, obsługę języka i tworzenie treści, znacznie redukując nakład pracy ręcznej i czas przetwarzania danych. Umożliwia przetwarzanie dużych ilości danych i wykonywanie złożonych operacji, takich jak podsumowywanie, tłumaczenie i generowanie kodu, szybciej niż metody ręczne.

2. Skalowalność i adaptowalność:

Dane można trenować i dostosowywać do nowych kontekstów i zadań w miarę upływu czasu. Pozwala to na ich wykorzystanie na wiele różnych sposobów. W przypadku aplikacji na dużą skalę dane można łatwo skalować w odpowiedzi na rosnące wolumeny danych i wymagania dotyczące przetwarzania.

3. Wydajność i dokładność:

Współczesne LLM wykazują się niezwykłą wydajnością i dokładnością w różnych zadaniach, często przewyższając ludzkie umiejętności w określonych obszarach. Może to prowadzić do wiarygodnych i rzetelnych wyników. Stale doskonalą swoją zdolność uczenia się z dużych zbiorów danych, co może prowadzić do jeszcze większej wydajności i dokładności w przyszłości.

4. Innowacyjność i kreatywność:

Wstępne trenowanie danych może pomóc w projektach artystycznych, procesach projektowania i przełomach naukowych poprzez proponowanie innowacyjnych pomysłów i rozwiązań.

Zalety wstępnego trenowania w LLM

Wstępne trenowanie danych w dużych modelach językowych oferuje szeroki wachlarz korzyści [8] w różnych dziedzinach i zastosowaniach. Oto kilka kluczowych korzyści:

1. Wydajność i automatyzacja:

- Wstępne trenowanie LLM może zautomatyzować analizę danych, obsługę języka i tworzenie treści, znacznie redukując nakład pracy ręcznej i czas przetwarzania danych.

- Mogą przetwarzać duże ilości danych i wykonywać złożone operacje, takie jak podsumowywanie, tłumaczenie i generowanie kodu, szybciej niż metody ręczne.

2. Skalowalność i adaptowalność:

- Dane można trenować i dostosowywać do nowych kontekstów i zadań w miarę upływu czasu. Pozwala to na ich wykorzystanie na wiele różnych sposobów.

- W przypadku aplikacji na dużą skalę dane można łatwo skalować w odpowiedzi na rosnące wolumeny danych i wymagania dotyczące przetwarzania.

3. Wydajność i dokładność:

- Współczesne LLM wykazują się niezwykłą wydajnością i dokładnością w różnych zadaniach, często przewyższając ludzkie umiejętności w określonych obszarach. Może to prowadzić do wiarygodnych i rzetelnych wyników.

- Nieustannie doskonalą swoją zdolność uczenia się z dużych zbiorów danych, co może prowadzić do jeszcze większej wydajności i dokładności w przyszłości.

4. Innowacyjność i kreatywność:

- Wstępne trenowanie danych może pomóc w projektach artystycznych, procesach projektowania i przełomach naukowych poprzez proponowanie innowacyjnych pomysłów i rozwiązań.

- Co więcej, potrafi interpretować języki, tworzyć kreatywne struktury tekstowe i oferować różne perspektywy, co może inspirować nowatorskie koncepcje w różnych sektorach.

5. Demokratyzacja dostępu:

- Wstępnie przeszkoleni LLM-owie oferują szerszemu gronu użytkowników, w tym osobom i organizatorom bez rozległej wiedzy technicznej, możliwość wykorzystania zaawansowanego przetwarzania języka i możliwości sztucznej inteligencji.

- Otwiera to możliwości dla różnych osób i organizacji w zakresie wykorzystania LLM-ów do ich konkretnych zastosowań i przedsięwzięć.

Etapy szkolenia modeli LLM

Nauczanie dużych modeli językowych rozumienia i tworzenia języka ludzkiego nazywa się wstępnym trenowaniem LLM. Osiąga się to poprzez dodanie dużej ilości danych tekstowych (lub danych tekstowych i graficznych w architekturach multimodalnych) do modelu, który wykorzystuje algorytmy do rozpoznawania wzorców i przewidywania struktury zdań. Produktem końcowym jest system sztucznej inteligencji, który może generować tekst wyglądający jak tekst ludzki, tłumaczyć na różne języki, odpowiadać na zapytania użytkowników i wykonywać wiele innych funkcji poznawczych. W LLM liczba parametrów w modelu jest określana jako „duża". Model tworzy prognozy, wykorzystując te parametry jako zmienne. Możliwości sztucznej inteligencji w zakresie zaawansowanego i złożonego rozumienia języka rosną wraz z liczbą czynników. Jednak do trenowania tak dużych modeli z dużą ilością danych potrzebna jest znaczna moc obliczeniowa i kilka złożonych algorytmów. Naukowcy badają i przesuwają granice możliwości dużych modeli językowych w wyniku ich ciągłej ewolucji. Wraz z każdą nową architekturą i większym zestawem parametrów rośnie zapotrzebowanie na lepszą wydajność, co napędza ewolucję metod szkolenia LLM. Ze względu na rozmiar i złożoność nowoczesnych LLM , wymagana jest duża moc obliczeniowa. W tym miejscu potrzebne są procesory graficzne (GPU) hostowane w chmurze. Te chmurowe procesory graficzne (GPU) są niezwykle szybkie i wydajne, ponieważ mogą wykonywać kilka skomplikowanych obliczeń matematycznych jednocześnie. Jednak oprócz mocy obliczeniowej, precyzja i opóźnienie okazują się istotnymi zmiennymi wpływającymi na wydajność LLM. Naukowcy badają nowe i alternatywne sposoby szkolenia LLM ze względu na rosnące zapotrzebowanie na szybkie i dokładne odpowiedzi modeli. Szkolenie dużych modeli językowych to nie lada wyzwanie, które obejmuje kilka kluczowych kroków. Uproszczony, krok po kroku opis procesu wstępnego szkolenia LLM przedstawiono na Rysunku 2.

1. Gromadzenie danych:

Zebranie znacznej ilości danych tekstowych to pierwszy krok w szkoleniu LLM. Książki, blogi, artykuły i platformy mediów społecznościowych mogą dostarczyć tych informacji. Celem jest uchwycenie bogactwa języka ludzkiego.

2. Oczyszczanie danych:

Następnie, w etapie znanym jako przetwarzanie wstępne, nieprzetworzone dane tekstowe są oczyszczane. Tokenizacja, czyli podział tekstu na łatwe do zarządzania fragmenty zwane tokenami, oraz formatowanie materiału do formatu zrozumiałego dla modelu, to tylko niektóre przykłady działań związanych z tym procesem.

3. Podział danych: Oczyszczone dane są następnie dzielone na dwie grupy. Model będzie trenowany przy użyciu jednego zestawu, znanego jako dane treningowe. Wydajność modelu będzie następnie oceniana przy użyciu drugiego zestawu, danych walidacyjnych.

4. Konfiguracja modelu:

Następnie tworzona jest architektura (tj. struktura) LLM. Wymaga to wyboru sieci neuronowej i określenia parametrów, takich jak liczba warstw i jednostek ukrytych w sieci.

5. Trening modelu:

To jest krok, na którym rozpoczyna się właściwy trening modelu. Podczas treningu model LLM dostosowuje wewnętrzne parametry danych, aby zminimalizować rozbieżności między prognozą modelu a danymi rzeczywistymi. Trening został przeprowadzony po przeanalizowaniu zbioru treningowego i wygenerowaniu prognoz w oparciu o wcześniejszą wiedzę o danych.

6. Sprawdzanie modelu:

Po treningu, uczenie modelu LLM jest weryfikowane za pomocą danych walidacyjnych. Ten krok ułatwia ocenę wydajności modelu i dostosowanie jego ustawień w celu uzyskania lepszych rezultatów.

7. Korzystanie z modelu:

Po zakończeniu oceny modelu staje się on operacyjny. Następnie model jest włączany do programów lub systemów, aby mógł generować tekst w odpowiedzi na nowe dane wejściowe.

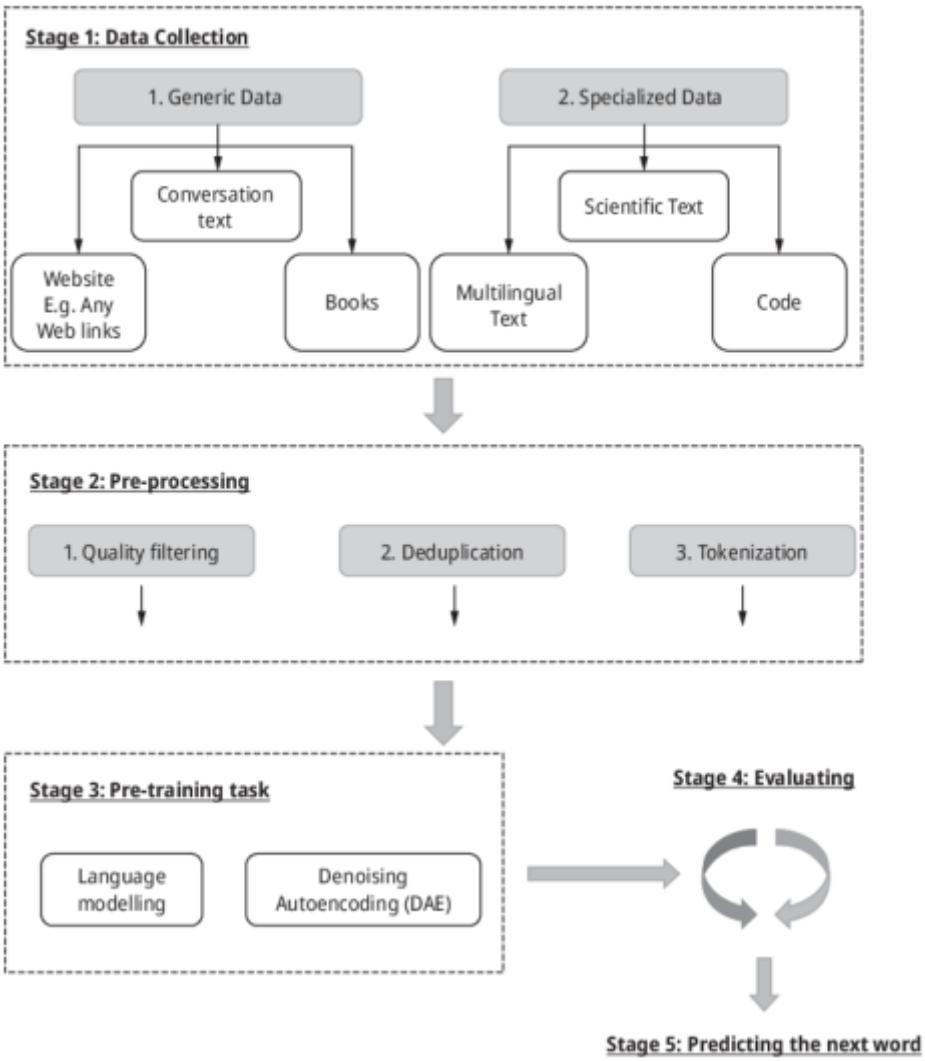
8. Ulepszanie modelu:

Wreszcie, wraz z odpowiedzią na opinie użytkowników, istnieje przestrzeń do udoskonalenia poprzez wykorzystanie dodatkowych danych lub zmianę parametrów w odpowiedzi na opinie użytkowników i rzeczywiste użytkowanie.

Należy pamiętać, że procedura ta wymaga znacznych zasobów obliczeniowych, w tym wydajnych procesorów i dużej pamięci masowej, a także specjalistycznej wiedzy z zakresu uczenia maszynowego. Z tego powodu jest ona zazwyczaj przeprowadzana przez zaangażowane zespoły badawcze lub firmy, które mają dostęp do niezbędnych zasobów i wiedzy specjalistycznej.

Badanie wstępnego trenowania w modelu LLM

Podstawową mocą modeli LLM jest wstępne trenowanie. Modele te osiągają biegłość w rozumieniu i tworzeniu języka poprzez intensywny trening na materiale tekstowym [14]. Wstępne trenowanie jest najważniejszym krokiem w celu uzyskania wyjątkowo dobrych modeli, a modele LLM muszą być trenowane na dużej ilości danych. Przebieg procesu wstępnego trenowania zilustrowano na Rysunku 3.



Ilość i jakość tych danych mają kluczowe znaczenie. Ulepszone zbiory danych umożliwiają modelowi działanie na wyższym poziomie.

Gromadzenie danych

Ilość wymaganych danych treningowych zależy od różnych czynników, takich jak rodzaj problemu, złożoność modelu, liczba cech i tolerancja na błędy. Chociaż nie ma sztywnych reguł, powszechnym zaleceniem jest posiadanie co najmniej 10 razy więcej instancji niż cech . Biorąc pod uwagę ambitny charakter celu modelu LLM, do wstępnego trenowania wymagana jest duża ilość wysokiej jakości danych. Absolwenci studiów prawniczych (LLM) są szkoleni z wykorzystaniem dwóch podstawowych kategorii danych, a mianowicie (1) danych ogólnych, np. książek, rejestrów czatów i stron internetowych, oraz (2) danych specjalistycznych, np. materiałów naukowych, kodu programistycznego i zestawów danych w kilku językach. Motywacją do korzystania z danych ogólnych jest ich szeroka dostępność, różnorodność, pomoc w rozwoju rozumienia języka ogólnego przez absolwentów studiów prawniczych (LLM) oraz elastyczność w posługiwaniu się nim. Podobnie, dane specjalistyczne pomagają absolwentom studiów prawniczych (LLM) w zdobywaniu biegłości w określonych dziedzinach lub obowiązkach. Odkryto, że absolwentów studiów prawniczych (LLM) można uczyć na podstawie określonych rodzajów danych, aby doskonalić się w zadaniach

specjalistycznych, mimo że zazwyczaj są szkoleni na podstawie danych ogólnych, aby rozumieć codzienny język angielski. Te dwie podstawowe kategorie danych zostały szczegółowo wyjaśnione.

Dane ogólne

Dane ogólnego przeznaczenia są wykorzystywane przez większość osób studiujących na kierunku LLM. Trzy podstawowe kategorie to:

* Strony internetowe: Ta funkcja udostępnia szeroki wachlarz informacji z internetu, oferując różnorodne możliwości językowe, np. Common Crawl. Główną wadą takich danych jest to, że tekst w sieci może mieć dwa różne poziomy jakości: spam i tekst wysokiej jakości, który można znaleźć w Wikipedii. Aby zagwarantować jakość, konieczne jest oczyszczenie i przetworzenie tych danych.

* Tekst konwersacji: Ta funkcja zwiększa możliwości osób studiujących na kierunku LLM w zakresie prowadzenia konwersacji i odpowiadania na pytania. Służy do rejestrowania odpowiedzi, czyli konwersacji na platformach internetowych, sformatowanych w formie drzew. Umożliwia to podzielenie rozmów wielostronnych na łatwiejsze w zarządzaniu dialogi do celów szkoleniowych, np. korpus Reddita z PushShift.io. Mogą wystąpić problemy, jeśli dane dialogowe opierają się na nadmiernej ilości danych.

* Książki: Znaczenie takich danych polega na tym, że dostarczają one absolwentom studiów magisterskich (LLM) formalnych, obszernych tekstów, które pomagają im w zrozumieniu zawiłych struktur językowych, długoterminowego kontekstu i tworzeniu spójnych narracji. Zbiór danych Pile zawiera Books3, a Bookcorpus2 jest najlepszym przykładem. Te ogólne źródła danych zasadniczo pomagają absolwentom studiów magisterskich (LLM) w rozumieniu i tworzeniu różnorodnych tekstów języka naturalnego; jednak każde źródło ma swoje własne zalety i wady.

Dane specjalistyczne

Dane specjalistyczne pomagają w poprawie wydajności LLM w określonych zadaniach. Podstawowe trzy kategorie danych specjalistycznych to:

Tekst wielojęzyczny: Celem tekstu wielojęzycznego jest usprawnienie generowania tekstu wielojęzycznego i zrozumienia języka. Dwa przykładowe zestawy danych to PaLM (obejmujący 122 języki) i BLOOM (obejmujący 46 języków). Modele te przewyższają modele trenowane wyłącznie na danych w języku docelowym i doskonale sprawdzają się w zadaniach takich jak tłumaczenie, wielojęzyczne podsumowania i wielojęzyczne pytania i odpowiedzi.

Tekst naukowy: Celem tych danych jest pomoc LLM w zrozumieniu wiedzy naukowej. Przykładami są strony internetowe poświęcone matematyce, podręczniki naukowe i artykuły w ArXiv. LLM trenowani na tekstach naukowych mogą lepiej radzić sobie i rozumować w dziedzinach naukowych. Głównym wyzwaniem w korzystaniu z danych naukowych jest to, że składają się one z symboli matematycznych i sekwencji białek. Są one specjalnie tokenizowane i wstępnie przetwarzane, aby dopasować je do formatu, który LLM może wykorzystać do ich obsługi.

Kod: Kształcenie absolwentów LLM w zakresie kodowania ułatwia syntezę programów, co jest popularnym kierunkiem studiów. Nawet dla doświadczonych absolwentów LLM, takich jak GPT-J, tworzenie precyzyjnych i wysokiej jakości programów jest trudne. Kod można uzyskać z otwartych repozytoriów oprogramowania, takich jak GitHub, lub forów pytań i odpowiedzi, takich jak Stack Exchange. Zawiera on rzeczywisty kod, komentarze i dokumentację. Absolwenci LLM mogą generować odpowiedzi z większą dokładnością, gdy zadania są prezentowane w formie kodu. Krótko mówiąc, absolwenci LLM posiadają specjalistyczne umiejętności, od generowania kodu po rozumienie

wielojęzyczne, dzięki wykorzystaniu specjalistycznych danych. Kod znacznie różni się od zwykłego tekstu tym, że ma własną gramatykę i logikę.

Jednak szkolenie w zakresie kodowania może wyposażyć absolwentów LLM w zaawansowane zdolności rozumowania.

Absolwenci LLM analizują ogromne zbiory danych tekstowych, a następnie wykorzystują techniki oparte na sieciach neuronowych do zrozumienia struktur i wzorców językowych. Systemy wykorzystują zdobytą wiedzę do opracowywania odpowiedzi na polecenia i pytania, przewidując najbardziej prawdopodobną strukturę słowa lub zdania. Aby model języka funkcjonował prawidłowo, konieczne są odpowiednie dane treningowe.

Wstępne przetwarzanie danych

Wstępne przetwarzanie danych jest kluczowym etapem w trenowaniu każdego modelu LLM, który uwzględnia szum, braki danych, niespójności i zmienność danych. Do grupowania technik wstępnego przetwarzania danych można użyć czterech kategorii: redukcji, integracji, transformacji i czyszczenia . W przypadku wstępnego badania modeli LLM, do wstępnego przetwarzania danych brane są pod uwagę najpopularniejsze formaty danych zebrane w poprzednim etapie; następnie dane są przetwarzane w celu integracji i czyszczenia. Wstępne przetwarzanie danych można ogólnie podzielić na trzy główne kategorie:

1. Filtrowanie jakości: Eliminowanie danych niskiej jakości za pomocą metod heurystycznych lub opartych na klasyfikatorach z korpusu nazywa się filtrowaniem jakości. Techniki oparte na klasyfikatorach wykorzystują dane kandydatów jako przykłady negatywne i starannie wyselekcjonowane dane jako instancje pozytywne do trenowania klasyfikatora binarnego, który następnie przewiduje wynik jakości każdego przykładu danych. Techniki oparte na heurystyce wykorzystują reguły lub heurystyki do eliminowania danych niskiej jakości zgodnie z ustalonymi standardami.

2. Deduplikacja: Wiele źródeł, w tym scrapowanie online, scalanie i augmentacja danych, może generować duplikaty dokumentów, co może prowadzić do szeregu problemów, takich jak nadmierne dopasowanie, stronniczość i nieefektywność. Aby rozwiązać te problemy, wcześniejsze badania opierały się głównie na współczynniku nakładania się cech powierzchniowych (takich jak nakładanie się słów i n-gramów) w celu identyfikacji i eliminacji duplikatów artykułów o identycznej treści. Ponadto, eliminując wszelkie potencjalne duplikaty tekstów ze zbioru treningowego, niezbędne jest wyeliminowanie nakładania się zbiorów treningowego i oceniającego, aby uniknąć problemu zanieczyszczenia zbioru danych. Wykazano, że trzy poziomy deduplikacji, tj. dokument, zdanie i token, są pomocne w ulepszaniu szkolenia LLM i powinny być łączone w rzeczywistych zastosowaniach.

3. Tokenizacja: Podział oryginalnego tekstu na oddzielne tokeny lub jednostki podsłowne nazywa się tokenizacją, aby dane wejściowe mogły być łatwo wykorzystane do zasilania modelu. Do przeprowadzenia procesu tokenizacji można użyć wielu algorytmów, w tym opartych na regułach, statystycznych i bazujących na odstępach.

Współczynnik nakładania się

Współczynnik nakładania się definiuje się jako odsetek słów lub n-gramów, jaki mają dwa zdania, stwierdzenia lub dokumenty zawierające słowa wspólne. Oblicza się go jako stosunek liczby słów

wspólnych dla dwóch dokumentów lub zdań do całkowitej liczby słów w całym zdaniu lub dokumencie. Rozważmy na przykład następujące stwierdzenia:

Twierdzenie 1: Uwielbiam pić kawę

Twierdzenie 2: Uwielbiam pić kawę Starbucks

W tym przypadku współczynnik nakładania się obu stwierdzeń wynosi 50%, ponieważ oba stwierdzenia mają większość wspólnych słów. Współczynnik nakładania się można obliczyć w celu identyfikacji duplikatów słów lub duplikatów dokumentów. Jeśli dwa dokumenty mają wysoki współczynnik nakładania się, prawdopodobnie istnieje więcej duplikatów lub można to powiedzieć, ponieważ oba dokumenty są bardziej podobne. Podobnie, gdy mają niski współczynnik nakładania się, prawdopodobnie oba stwierdzenia lub dokumenty są unikatowe lub różne od siebie. Współczynnik nakładania się jest powszechnie stosowany w zadaniach przetwarzania języka naturalnego, takich jak deduplikacja, klasyfikacja tekstu i tłumaczenie maszynowe.

Zadanie wstępnego trenowania

Wstępne trenowanie to proces nienadzorowanego trenowania modelu sieci neuronowej na dużym korpusie danych tekstowych. Jest to pierwszy etap treningu uczenia maszynowego i niezbędny krok w wyposażaniu absolwenta studiów prawniczych (LLM) w umiejętność interpretacji języka angielskiego ogólnego. Można go dostosować po wstępnym trenowaniu, aby osiągnąć zamierzone rezultaty. Modele językowe mogą efektywnie radzić sobie z nowymi zadaniami podczas dostrajania, korzystając z wcześniejszych doświadczeń, zamiast zaczynać od zera [18]. Pozwala to modelowi czerpać korzyści z wcześniejszego treningu. Podobne wrodzone zdolności u ludzi pozwalają nam czerpać z dotychczasowej wiedzy, aby uniknąć konieczności zaczynania od zera w obliczu nowych sytuacji. Modele wstępnie wytrenowane z kolei nie posiadają żadnej specjalizacji, ale posiadają podstawową wiedzę i mogą wykonywać szeroki zakres zadań. Istnieje kilka dodatkowych etapów nauki, które student LLM musi ukończyć, aby osiągnąć poziom biegłości w generowaniu tekstu, umiejętnościach konwersacyjnych i tworzeniu innych treści na żądanie. Dwa powszechnie stosowane zadania wstępnego treningu to modelowanie języka (LM) i autokodowanie odszumiające.

Modelowanie języka w LLM

Jednym z podstawowych zadań w NLP jest modelowanie języka (LM). Polega ono na wykorzystaniu historii tokena do prognozowania, który token pojawi się jako następny w serii. Jest to kluczowa czynność przedtreningowa dla modeli LLM, szczególnie w przypadku modeli opartych wyłącznie na dekoderze. Oto zwięzłe podsumowanie dostarczonych informacji:

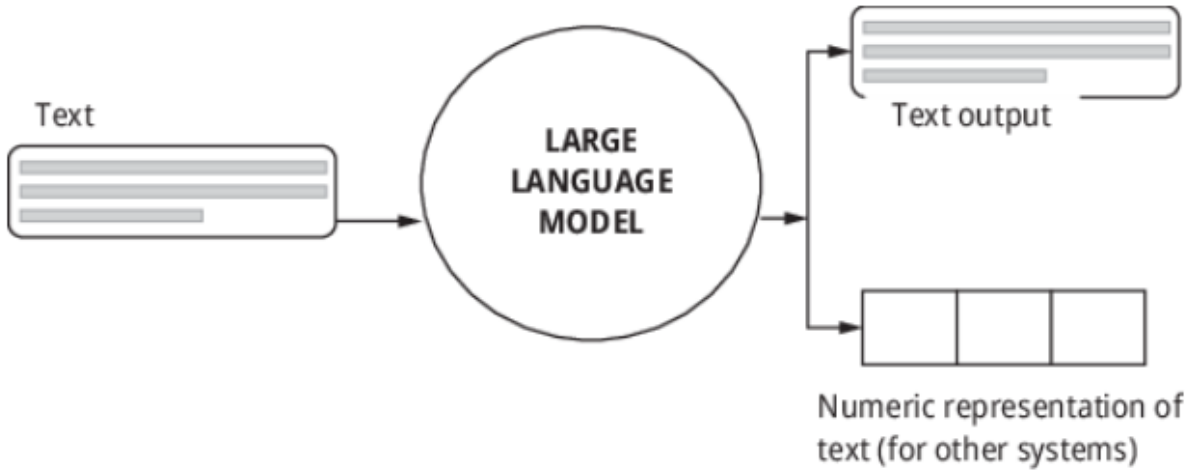
Definicja: Weźmy sekwencję tokenów, która jest dana wzorem $d = d_1, d_2, d_3, d_4, \ldots, d_n$

Teraz celem jest autoregresyjne przewidywanie każdego tokena. Niech zadaniem modelu LM będzie przewidzenie tokena o nazwie „di” w danej sekwencji na podstawie podanej historii ($d < i$). Proces ten można zdefiniować matematycznie w następujący sposób:

$$\mathrm{LLM}(d) = \sum_{i=1}^{n} \log P(di \mid d < i) \tag{5.1}$$

To równanie zasadniczo oblicza logarytm przewidywania każdego tokena (di) z podanymi tokenami poprzedzającymi sekwencję. Modele oparte wyłącznie na dekoderze, takie jak GPT3 i PaLM, w dużej mierze opierają się na zadaniu LM w przypadku wstępnego trenowania. Jedną z najmocniejszych stron tych modeli jest to, że wiele zadań językowych można przeformułować jako przewidywanie

problemów, co dobrze wpisuje się w wrodzoną tendencję modeli LLM do trenowania zgodnie z celami LM. Ciekawym odkryciem jest to, że niektóre modele LLM oparte wyłącznie na dekoderze mogą być wykorzystywane do zadań, w których wystarczy prognozować kolejne tokeny autoregresywnie. Oznacza to, że mogą one czasami wykonywać zadania bez konieczności precyzyjnego dostrajania. Jednym znaczącym wyjątkiem od typowej pracy LM jest prefiks LM. Podstawową strukturę procesu modelu języka zilustrowano na Rysunku 4.



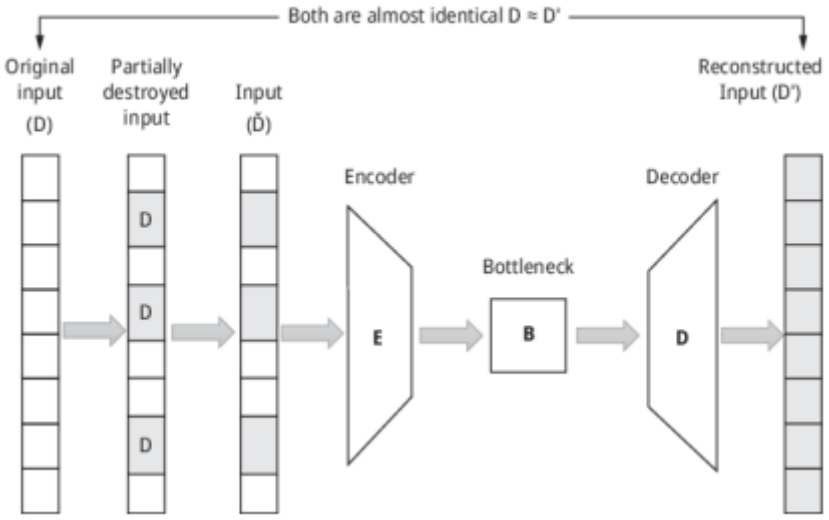
W tym wariancie obliczanie strat uwzględnia tylko tokeny dłuższe niż prefiks i wybrane losowo. Prefiks LM generalnie działa gorzej niż standardowy LM, nawet gdy model widzi taką samą liczbę tokenów podczas wstępnego trenowania, jak w standardowym LM, ponieważ mniej tokenów w sekwencji jest wykorzystywanych podczas wstępnego trenowania modelu. Mówiąc prościej, podstawą wstępnego trenowania LLM z samym dekoderem jest LM. Ze względu na swoją autoregresyjną strukturę, LLM mogą uczyć się szerokiego zakresu zadań niejawnie, często bez potrzeby precyzyjnego dostrajania specyficznego dla danego zadania. Modyfikacje i adaptacje, takie jak prefiks LM, zapewniają alternatywne zastosowania dla tego problemu, ale podstawowa idea przewidywania sekwencji tokenów pozostaje ta sama.

Zadanie odszumiania w procesie wstępnego uczenia LLM

Sieci neuronowe, czyli autoenkodery, są często wykorzystywane w procesach ekstrakcji i selekcji cech. Sieć narażona jest na ryzyko uczenia się tzw. „funkcji tożsamości", znanej również jako „funkcja zerowa", która wskazuje, że dane wyjściowe są równe danym wejściowym i sprawia, że autoenkoder jest nieskuteczny, gdy w warstwie ukrytej jest więcej węzłów niż danych wejściowych. Aby zaradzić temu problemowi, autoenkodery celowo manipulują danymi, arbitralnie ustawiając niektóre wartości wejściowe na zero. Ogólnie rzecz biorąc, ponad 50% węzłów wejściowych jest ustawianych na zero. Niektóre źródła proponują niższą wartość, np. 30%. Decydują o tym takie czynniki, jak ilość danych i wybrane węzły wejściowe. Podczas obliczania funkcji straty kluczowe jest porównanie wartości wyjściowych z oryginalnymi danymi wejściowymi, a nie z uszkodzonymi danymi wejściowymi. W ten sposób nie ma możliwości przypadkowego odkrycia funkcji identyfikacyjnej zamiast identyfikacji cech. OpenDeep.org udostępnił fantastyczną implementację, w której trenuje bardzo prosty autokoder odszumiający na zbiorze danych MNIST za pomocą Theano. Artykuły OpenDeep są przeznaczone dla początkujących i są niezwykle podstawowe. Dzięki temu, nawet jeśli czytelnik nie jest zbyt obeznany z sieciami neuronowymi, z łatwością zrozumie ten proces.

(1) Autokodowanie odszumiające (DAE): Autokodowanie odszumiające jest częścią modeli sieci neuronowych, które wykorzystują swoją zdolność do odzyskiwania oryginalnych danych z ich

zaszumionych odpowiedników w celu eliminacji szumu z danych zniekształconych lub zaszumionych. Aby zmniejszyć różnicę między danymi oryginalnymi a odtworzonymi, można trenować model (rysunek 5).



Te autokodery można łączyć w stosy, tworząc głębokie sieci, co poprawi wydajność sieci.

Definicja – Aby zrealizować zadanie autokodowania odszumiającego, niektóre fragmenty tekstu wejściowego są celowo uszkadzane poprzez zmianę określonych spanów. Głównym celem jest trenowanie modelu w celu odzyskania oryginalnych, tj. nieuszkodzonych tokenów z tekstu wejściowego. Cel ten jest wyrażony matematycznie jako

$$L_{\mathrm{DAE}}(d) = \log Pd'|d\backslash d' \qquad (5.2)$$

W powyższym równaniu d' odnosi się do tokenów, które są celowo zmieniane w celu utworzenia uszkodzonej sekwencji. Dlatego model jest trenowany w celu przewidywania takich tokenów na podstawie uszkodzonego wejścia d\d'.

W porównaniu z typowym zadaniem LM, zadanie DAE może być bardziej złożone w implementacji, pomimo swojej teoretycznej siły. W rezultacie nie było ono tak szeroko stosowane do wstępnego trenowania LLM. Z drugiej strony, modele autoregresyjne, takie jak T5 i GLM-130B, wykorzystują DAE jako cel wstępnego treningu i próbują odzyskać podstawione zakresy.

(2) Mieszanka denoiserów (MoD): MoD zapewnia jednolity cel wstępnego treningu dla modeli językowych; jest również nazywana stratą UL2 [20]. Sugeruje ona traktowanie zadań LM i DAE jako dwóch różnych typów zadań odszumiania. Istnieją trzy typy denoiserów, a mianowicie:

i. S-denoiser (LM): Jest podobny do tradycyjnego celu LM.

ii. R-denoiser: Odmiana DAE wprowadzająca zniekształcenia do krótkich segmentów tekstu. Charakteryzuje się krótkim zakresem i niskimi zniekształceniami.

iii. X-denoiser: Inna wersja DAE, charakteryzująca się dłuższym zakresem zniekształceń lub większym współczynnikiem zniekształceń, nazywana jest X-denoiser. Ten typ DAE charakteryzuje się długim zakresem zniekształceń lub wysokim współczynnikiem zniekształceń.

Model można trenować na różnych celach wstępnego treningu, wykorzystując framework UL2, który można również wykorzystać do nadania modelowi możliwości i zalet wynikających z odchyleń indukcyjnych z różnych działań wstępnego treningu. Trenując na mieszance, model jest w stanie zrównoważyć swoje wady i wykorzystać swoje mocne strony w innych zadaniach. W przeciwieństwie do modelu T5 z uszkodzeniem tylko rozpiętości, cel mieszanki denoiserów może znacząco zwiększyć potencjał uczenia się modelu na podstawie podpowiedzi. Różne denoisery z wyżej wymienionych typów są wykorzystywane do optymalizacji modelu w zależności od początkowych tokenów specjalnych we frazach wejściowych (takich jak {[R], [S] i [X]}). Przykładową frazą wykorzystującą denoiser S (LM) byłaby ta, która zaczyna się od tokena [S]. MoD został uwzględniony w modelach takich jak PaLM 2. Silny uczeń kontekstowy UL2 jest biegły zarówno w podpowiadaniu na podstawie kilku strzałów, jak i łańcucha myśli (CoT). →Tabela 5.1 przedstawia wydajność UL2 w porównaniu z różnymi najnowocześniejszymi modelami (takimi jak T5 XXL i PaLM) dla podpowiedzi z niewielką liczbą uderzeń z wykorzystaniem podsumowującego zbioru danych XSUM. Na podstawie uzyskanych wyników, UL2 20B działa lepiej niż T5 i PaLM, które charakteryzują się podobnymi kosztami obliczeniowymi.

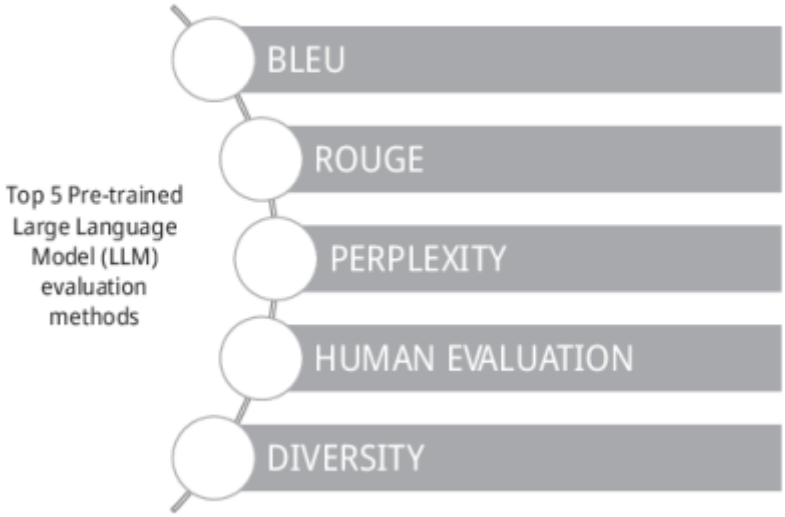
Tabela 1: Porównanie UL2 ze stanem techniki.

S. no.	Model	Rouge-1	Rouge-2	Rouge-L
1	LaMDA 137B	-	5.4	-
2	PaLM 62B	-	11.2	-
3	PaLM 540B	-	12.2	-
4	PaLM 8B	-	4.5	-
5	T5 XXL 11B	0.6	0.1	0.6
6	T5 XXL 11B + LM	13.3	2.3	10.7
7	UL2 20B	25.5	8.6	19.8

Większe modele językowe, takie jak GPT-3 175B, PaLM 540B lub LaMDA 137B, zostały wykorzystane do uzyskania większości wyników podpowiedzi CoT. Wykazano, że UL2 20B, publicznie dostępny model, który jest kilkakrotnie mniejszy niż poprzednie modele wykorzystujące podpowiedzi CoT, może być użyty do osiągnięcia rozumowania za pomocą podpowiedzi CoT. Stwarza to jasną ścieżkę dla badaczy do przeprowadzenia dostępnych badań nad podpowiedziami CoT i rozumowaniem. Tabela 1 ilustruje, jak w przypadku UL2 podpowiedzi CoT działają lepiej niż standardowe podpowiedzi w przypadku różnych matematycznych problemów tekstowych (GSM8K, SVAMP, ASDiv, AQuA i MAWPS). Podobnie, z demonstracji wynika, że samospójność również poprawia wydajność. Głównym celem odszumiania zadań w pretreningu LLM jest umożliwienie modelowi odzyskiwania brakujących lub błędnie sformatowanych segmentów sekwencji wejściowych. LLM można wstępnie wytrenować i udoskonalić, stosując różne techniki oferowane przez obiektywy odszumiające, takie jak DAE i MoD, mimo że podstawowe zadanie LM nadal pozostaje najważniejsze.

Ocena wstępnie wytrenowanego modelu

Podobnie jak każdy inny model uczenia maszynowego, modele LLM muszą zostać ocenione po treningu, aby określić, czy szkolenie było skuteczne, a także określić, jak model wypada w porównaniu z innymi modelami, testami porównawczymi lub alternatywnymi technikami . Oceny LLM wykorzystują zarówno strategie wewnętrzne, jak i zewnętrzne. Niektóre z najczęściej stosowanych przez oceniających metod wymieniono jak pokazano na Rysunku 6.



Metody wewnętrzne

Dokładność językowa modelu lub skuteczność przewidywania słów jest mierzona obiektywnymi, ilościowymi kryteriami stosowanymi w analizie wewnętrznej do śledzenia wydajności. Wśród tych metryk znajdują się:

Płynność językowa – Ta metoda mierzy, jak naturalnie brzmi język generowany przez model LLM, zapewniając, że zdania tworzone przez model brzmią, jakby zostały napisane przez człowieka, poprzez badanie poprawności gramatycznej i różnorodności składniowej.

Spójność – Ocenia zdolność modelu do utrzymania spójności tematycznej w akapitach i frazach, upewniając się, że kolejne zdania wzmacniają się i wzajemnie się rozumieją. Perpleksja – statystycznym wskaźnikiem zdolności modelu do przewidywania próbki jest perpleksja. Model o niższym wyniku perpleksji lepiej pasuje do obserwowanych danych, a zatem lepiej przewiduje słowo, które pojawi się jako następne w sekwencji.

Metody zewnętrzne

Podejścia zewnętrzne są coraz częściej preferowane do oceny wyników absolwentów studiów prawniczych (LLM) ze względu na ich najnowsze osiągnięcia. Wiąże się to z oceną wyników modeli w zakresie praktycznych umiejętności, takich jak rozumowanie, rozwiązywanie problemów, informatyka i matematyka, a także w testach konkursowych, takich jak US Uniform Bar Exam, LSAT i GRE.

Do oceny LLM często stosuje się następujące techniki zewnętrzne:

Kwestionariusze – ocena wyników LLM w odpowiedziach na pytania skierowane do ludzi i porównanie ich wyników z wynikami uzyskanymi przez ludzi.

Wnioski zdroworozsądkowe – testowanie zdolności LLM do wyciągania wniosków, które są proste i zrozumiałe dla ludzi.

Wielozadaniowość – badanie, jak dobrze model radzi sobie z wieloma zadaniami w różnych dziedzinach, w tym z historią, prawem i matematyką.

Faktyczność – testowanie dokładności modelu w udzielaniu odpowiedzi zgodnych z faktami (a także poziomu halucynacji w tych odpowiedziach) jest znane jako faktowość.

Przewidywanie następnego słowa

Przewidywanie następnego słowa, czyli LM, ma na celu przewidzenie słowa, które pojawi się jako następne. LM jest jednym z zadań wzorcowych w NLP. W najprostszej formie polega na wybraniu słowa, które najprawdopodobniej wystąpi jako następne w ciągu słów. LM ma liczne zastosowania w wielu dziedzinach. Na przykład, sugerowanie tekstu na klawiaturze telefonu komórkowego i automatyczne uzupełnianie w wyszukiwarce Google. Przewidywanie następnego słowa w tytule to intrygujący problem w NLP. Dzięki analizie wzorców i korelacji modele są w stanie sugerować najbardziej prawdopodobny dyskurs na podstawie danych tekstowych. Ta moc predykcyjna umożliwia zastosowania takie jak systemy sugestii tekstu i automatyczne uzupełnianie. Zaawansowane metody, takie jak topologie oparte na transformatorach i sieci neuronowe (RNN), mogą uchwycić relacje kontekstowe i poprawić dokładność. Interesujące zadanie trenowania modelu w celu przewidywania następnego słowa w ciągu słów jest znane jako przewidywanie następnego słowa z wykorzystaniem dwukierunkowego LSTM w NLP:

* Aby przygotować surowe dane tekstowe do treningu BI-LSTM, wymagane jest przygotowanie danych. Ten etap obejmuje wektoryzację tekstu, tokenizację i tworzenie słownika.

* Efektywna architektura głębokiego uczenia BI-LSTM może uchwycić kontekst fraz i korelacje dalekiego zasięgu w celu sekwencyjnego przetwarzania danych.

* Utworzenie funkcji straty, użycie optymalizatora do zbudowania modelu, dopasowanie go do wstępnie przetworzonych danych i ocena jego wydajności na zbiorach walidacyjnych to kroki związane z trenowaniem modelu BI-LSTM.

* Algorytmy rekomendacji tekstu, generowanie języka i automatyczne uzupełnianie to niektóre z zastosowań modeli przewidywania następnego słowa.

* Aby osiągnąć biegłość w przewidywaniu następnego słowa w BI-LSTM, konieczne jest połączenie wiedzy teoretycznej z doświadczeniem praktycznym.

Uzupełnianie tekstu, tłumaczenie maszynowe i chatboty to tylko niektóre z zastosowań przewidywania następnego słowa. Dzięki dalszym badaniom i rozwojowi możliwe będzie tworzenie modeli przewidywania następnego słowa, które będą dokładniejsze i uwzględniać kontekst.

Wpływ wstępnego szkolenia na LLM

Do klasyfikacji wpływu wstępnego szkolenia na LLM można zastosować trzy ogólne kryteria.

1. Mieszanie źródeł

2. Ilość danych wstępnych szkolenia

3. Jakość danych wstępnych szkolenia

Nie zaleca się wielokrotnego szkolenia LLM dla każdego nowego zadania ze względu na ogromne koszty związane z zasobami i architekturą. Dlatego wstępne szkolenie LLM musi zapewnić mu solidny zestaw parametrów, aby mógł on później generalizować na zadania w dalszej kolejności:

(1) Mieszanie źródeł – Integracja danych tekstowych z kilku dziedzin może zwiększyć zdolność LLM do generalizacji w obrębie zadań i zapewnić im pełniejsze zrozumienie. Należy uwzględnić różnorodne, wysokiej jakości źródła danych i dokładnie rozważyć sposób ich dystrybucji i łączenia z różnych źródeł. Ważne jest, aby badacze dokładnie rozważyli, ile danych wstępnych szkolenia wykorzystują z każdego źródła. Celem jest stworzenie LLM, które spełniają ich unikalne wymagania, zachowując jednocześnie ogólne możliwości.

Wnioski: (i) Podczas wstępnego trenowania modeli LLM należy wziąć pod uwagę różnorodność źródeł danych; (ii) Wypróbować różne rozkłady danych, aby określić, który z nich najlepiej odpowiada potrzebom programisty; (iii) Rozpoznać zagrożenia związane z nadmiernym poleganiem na danych z jednej domeny.

(2) Ilość danych do wstępnego trenowania: Duże wolumeny wysokiej jakości danych są niezbędne do efektywnego wstępnego trenowania modeli LLM. Ilość danych potrzebnych do idealnego trenowania jest silnie skorelowana z rozmiarem modelu LLM. Brak odpowiednich danych do wstępnego trenowania uniemożliwia wielu modelom LLM osiągnięcie ich pełnej wydajności. Badania wskazują, że modele o większej wydajności obliczeniowej wynikają ze stałego zwiększania rozmiaru modelu i danych. Mniejsze modele, przy zapewnieniu większej ilości danych i dłuższego czasu trenowania, mogą generować niezwykłe rezultaty.

Implikacja: Zwiększając parametry modelu, należy wziąć pod uwagę, jak wystarczający jest zbiór treningowy. Należy zwrócić uwagę zarówno na objętość, jak i jakość danych. Należy próbować zmieniać rozmiar modelu i ilość danych, aż do znalezienia idealnej kombinacji odpowiadającej konkretnym wymaganiom użytkownika.

(3) Jakość danych przedtreningowych: Wydajność modeli LLM jest w dużym stopniu uzależniona od jakości danych treningowych. Wstępne trenowanie modelu z danymi niskiej jakości może mieć negatywny wpływ na jego wydajność. Badania wykazały, że modele trenowane na czystych, wysokiej jakości danych radzą sobie lepiej w zadaniach niż modele trenowane w dalszej części strumienia. Duplikowanie danych może prowadzić do szeregu problemów, w tym do upośledzenia zdolności kopiowania z kontekstu, schodzenia z duplikatów i nadmiernej duplikacji przejmującej proces trenowania. Dokładne wstępne przetwarzanie korpusu przedtreningowego jest niezbędne, aby zagwarantować stabilność podczas treningu i uniknąć niezamierzonych, szkodliwych skutków dla wydajności modelu.

Implikacja: Do wstępnego trenowania modeli LLM należy wybrać lepsze zbiory danych. Należy oczyścić i odfiltrować dane z szumu, toksyczności i duplikacji. Wypróbuj różne metody wstępnego przetwarzania, aby sprawdzić, która z nich jest najskuteczniejsza w przypadku Twojego konkretnego zestawu danych.

Kluczowe zagadnienia dotyczące przygotowania do studiów LLM

Rozpoczęcie szkolenia LLM od podstaw to trudny proces, który może być kosztowny i złożony. Istnieje kilka głównych przeszkód, które należy wziąć pod uwagę podczas szkolenia LLM:

1. Utworzenie infrastruktury: Do szkolenia LLM wykorzystuje się duże korpusy tekstowe, zazwyczaj o rozmiarze co najmniej 1000 GB.

Co więcej, modele używane do trenowania na tego typu zbiorach danych są ogromne i zawierają miliardy parametrów. Trenowanie tak dużych modeli wymaga infrastruktury z wieloma procesorami graficznymi (GPU). Trenowanie GPT-3, modelu poprzedniej generacji o 175 miliardach parametrów, wymagałoby 288 lat na jednym procesorze graficznym NVIDIA V100, aby zademonstrować wymagania obliczeniowe. Modele LLM są zazwyczaj trenowane równolegle na tysiącach procesorów graficznych. Na przykład Google wykorzystało 6144 chipy TPU v4 do rozłożenia treningu na nich, aby wytrenować swój model PaLM o 540 miliardach parametrów.

2. Koszt: Jednak dla większości organizacji zakup i hosting tak wielu procesorów graficznych jest nieopłacalny. Nawet OpenAI, firma stojąca za znanym modelem ChatGPT i serią modeli GPT, wykorzystała platformę chmurową Microsoft Azure do trenowania swoich modeli, a nie własny sprzęt.

Microsoft zainwestował 1 miliard dolarów w OpenAI w 2019 roku i prawdopodobnie znaczna część tych pieniędzy została przeznaczona na wykorzystanie zasobów chmury Azure do trenowania swoich modeli LLM. 3. Strategie dystrybucji modeli: Oprócz zakresu i kosztów, wdrażając instrukcje LLM na zasobach komputera, należy wziąć pod uwagę zawiłe szczegóły. W szczególności:

* Modele LLM są początkowo trenowane na jednym procesorze GPU, aby ocenić, ile zasobów będą zużywać.

* Paralelizm modeli to przydatna taktyka. Obejmuje on podział modeli na wiele procesorów GPU, a następnie optymalizację podzielonych modeli w celu maksymalizacji pamięci i przepustowości wejścia/wyjścia.

* Paralelizm modeli jest wymagany w przypadku naprawdę dużych modeli. W tej metodzie warstwy składowe modelu są dystrybuowane na kilka procesorów GPU. Aby zapewnić poprawne i efektywne wykonanie, wymaga to precyzyjnego kodowania, konfiguracji i starannej implementacji.

* Istotą szkolenia LLM jest iteracja. Naukowcy eksperymentują z różnymi konfiguracjami, dostosowując przebiegi szkoleniowe do unikalnych wymagań modelu i dostępnego sprzętu. Często stosuje się różnorodne metodologie obliczeń równoległych.

1. Wpływ wyboru architektury modelu: Złożoność treningu jest bezpośrednio zależna od wybranej architektury LLM. Poniższe zalecenia pomogą w modyfikacji architektury, aby dopasować ją do zasobów dostępnych w modelach:

* Ważne jest, aby dobrać głębokość i szerokość modelu (liczbę parametrów) w sposób zapewniający równowagę między złożonością a mocą obliczeniową.

* Preferowane jest stosowanie architektur z połączeniami resztkowymi. W rezultacie optymalizacja wykorzystania zasobów jest prostsza.

* Przeanalizuj, czy konieczna jest architektura transformatorowa z autoattencją, ponieważ ma ona szczególne wymagania treningowe.

* Określ wymagania funkcjonalne modelu, w tym uczenie wielozadaniowe, modelowanie generatywne, dwukierunkowe/maskowane modelowanie LM i analizę multimodalną.

* Przeprowadź treningi na znanych modelach, takich jak GPT, BERT i XLNet, aby sprawdzić, jak dobrze pasują do danego przypadku użycia.

* Wybierz tokenizację opartą na słowach, podsłowach lub znakach. Może to wpłynąć na długość danych wejściowych i rozmiar słownika, co bezpośrednio wpłynie na liczbę potrzebnych obliczeń.

Charakterystyka szkolenia wstępnego LLM

Poniżej przedstawiono charakterystykę szkolenia wstępnego LLM:

Edukacja bez nadzoru: W uczeniu się bez nadzoru, wstępne trenowanie jest analogiczne do zanurzenia modelu w oceanie danych tekstowych bez z góry określonych poprawnych lub niepoprawnych odpowiedzi. Jest to podobne do zapisania kogoś na program immersji językowej, gdzie nabywa on umiejętności językowe naturalnie poprzez ekspozycję i kontekst, a następnie stopniowo przyswaja różnice językowe bez wyraźnych instrukcji.

Dyskretne modelowanie języka – Ponieważ nie ma odpowiedzi poprawnej ani niepoprawnej, maskowane modelowanie języka (LM) jest popularną metodą nadania modelowi ram uczenia. Wyobraź sobie model jako badacza języka próbującego zrozumieć wypowiedź. Do jej przedstawienia

używane są zdania z celowymi pominięciami lub maskami słów. Na podstawie kontekstu model musi wnioskować, czym mogą być te brakujące słowa. Po otrzymaniu poprawnej odpowiedzi ocenia, jak bardzo się ona pomyliła, aby lepiej przewidywać. Ta procedura pomaga modelowi zrozumieć relacje między słowami i ich wpisanie w ogólną strukturę zdania.

Architektura transformatorów – Rozważ architekturę transformatora jako złożoną sieć powiązań między słowami. Zdolność modelu do rozpoznawania skojarzeń między słowami, nawet gdy są one od siebie odległe, jest podobna do sposobu, w jaki zbudowany jest mózg. Poprzez ręczny wybór odpowiednich słów z całego poprzedniego tekstu, modele wykorzystujące architekturę transformatora robią więcej niż tylko prognozowanie kolejnego słowa na podstawie bezpośredniej sekwencji słów poprzedzających. Mechanizm uwagi w tej architekturze pozwala modelowi nadawać priorytet odpowiednim fragmentom kontekstu i uwzględniać je jako całość, wychwytując niuanse niezbędne do dekodowania lub tłumaczenia znaczenia.

Przykłady zastosowań wstępnego szkolenia LLM

Wstępnie wyszkolone modele LLM można wykorzystać do różnych zadań lingwistycznych [26], z których niektóre wymieniono poniżej.

Tworzenie tekstu – Rozważ wstępnie wytrenowane modele jako najlepszych dostępnych narratorów. Potrafią tworzyć wciągające historie, tworzyć pomysłową poezję i oferować niezwykle osobiste reakcje. Firmy wykorzystują tę możliwość w chatbotach, które zapewniają pomoc w czasie rzeczywistym lub wirtualnych asystentach, którzy pomagają klientom w rozwiązywaniu problemów (choć to ostatnie często wymaga dopracowania). Na przykład, oparty na sztucznej inteligencji chatbot „Woebot" pomaga ludziom w dbaniu o zdrowie psychiczne, prowadząc z nimi rozmowy przypominające sesje terapeutyczne.

Tłumaczenie językowe – Wyobraź sobie znajomego, który mówi wieloma językami i potrafi tłumaczyć rozmowy na żądanie. Właśnie to potrafią wstępnie wytrenowane modele, które miały kontakt z różnymi językami. Firmy takie jak Airbnb wykorzystały tę funkcję, aby poprawić doświadczenia użytkowników, automatycznie tłumacząc wiadomości i opinie gospodarzy na wiele języków.

To prawie jak noszenie przy sobie przenośnego urządzenia do tłumaczenia. Analiza sentymentu – Pomyśl o technologii znanej jako analiza sentymentu, która mierzy emocje w tekstach. Aby to osiągnąć, konieczne byłoby udoskonalenie wstępnie wytrenowanych modeli z wykorzystaniem danych oznaczonych sentymentem. Na przykład Twitter (obecnie X) wykorzystuje analizę sentymentu, aby dowiedzieć się, co ludzie myślą na różne tematy i pomaga firmom zrozumieć, jak postrzegane są ich produkty i usługi. Poza tymi obszarami, sztuczna inteligencja generatywna odgrywa również większą rolę w systemach opieki zdrowotnej [27, 28], gdzie w celu uzyskania lepszych prognoz konieczne jest wytrenowanie obszernych danych na platformie LLM w przypadku wrażliwych przypadków [29–30].

Podsumowanie

Możliwości i błędy modeli LLM są w dużej mierze determinowane przez ich architekturę i zadania wstępnego trenowania. Chociaż wymagane są dodatkowe badania na temat innych modeli, w szczególności struktur koder-dekoder, obecne trendy wskazują na znaczne skłanianie się ku architekturze dekodera przyczynowego. Co więcej, modele LLM zmieniają się, ponieważ aplikacje wymagają szerszych okien kontekstowych; poczyniono nowe postępy zarówno pod względem wydajności obliczeniowej, jak i możliwości ekstrapolacji. Pomimo potencjalnej skuteczności i wartości modelu wstępnego trenowania, brakuje mu cech takich jak uczenie się przez wzmacnianie z wykorzystaniem ludzkiej informacji zwrotnej (RLFH) i dostrajanie, co czyni go nieodpowiednim do

wykorzystania jako kompletnego modelu LLM. Po zakończeniu wszystkich etapów treningu można wykonać kluczowy dodatkowy krok oceny wyników LLM, znany jako ciągła ewaluacja modelu. Zatem samo wstępne trenowanie nie wystarczy do stworzenia w pełni spersonalizowanego i wydajnego modelu. Deweloper musi również dodać inne procesy, takie jak RLFH i ewaluacja wyników modelu, oraz dostroić model LLM. Programiści mogą projektować niestandardowe modele językowe, dopasowane do unikalnych celów i założeń swoich aplikacji, włączając te kroki do procesu szkoleniowego. Oprócz zapewnienia, że model posiada niezbędną wiedzę, te dodatkowe procesy gwarantują również, że jest on w stanie dostosowywać się do zmieniających się sytuacji, uczyć się na podstawie opinii użytkowników i utrzymywać wysoką wydajność przez dłuższy czas.

Student oceny dwujęzycznej (punktacja BLEU) – ocenia podobieństwo między wynikami maszyny i człowieka, podkreślając dokładność przetłumaczonego tekstu lub generowanych odpowiedzi poprzez zliczanie odpowiadających sobie podciągów słów.