Generatywne modele AI i LLM: Techniki szkoleniowe i metryki ewaluacji

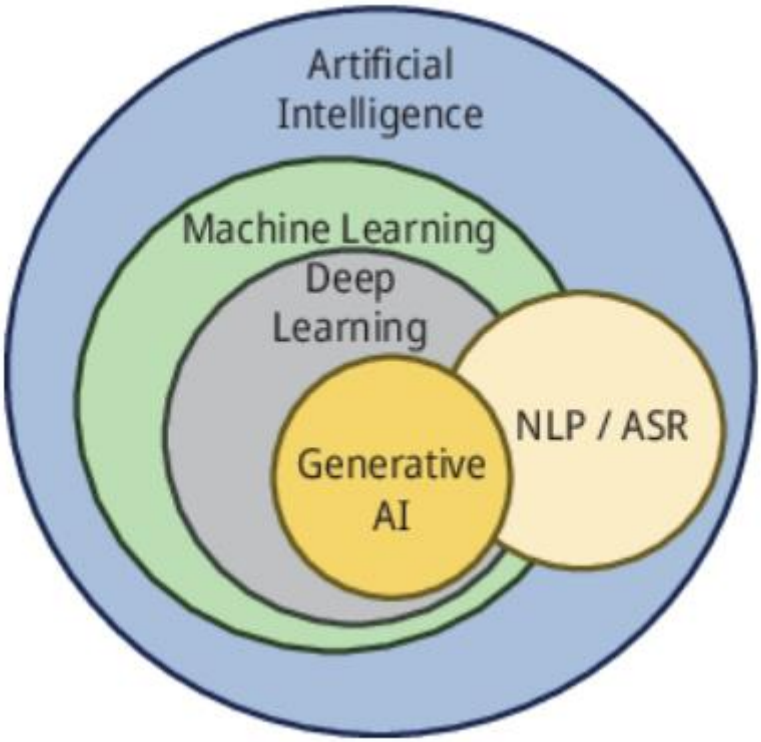
Streszczenie

Generatywna sztuczna inteligencja (AI) jest popularną techniką w aplikacjach opartych na danych, wykorzystującą architekturę głębokiego uczenia (ang. deep learning) do poznania podstawowych cech próbki i budowania bazy wiedzy w celu generowania syntetycznych próbek, które naśladują rzeczywisty rozkład. Generatywne modele AI to idealne rozwiązania w sytuacjach, gdy modele cierpią z powodu niedoboru próbek danych, który utrudnia proces szkolenia, niezależnie od tego, czy są to dane tekstowe, wideo, audio czy obrazy. Szkolenie modelu odgrywa kluczową rolę, ponieważ odkrywa on ukryty wzorzec i rozumie wewnętrzne zachowanie próbek, co pomaga w generowaniu realistycznych próbek. Ilość danych dostępnych do szkolenia i wymagana moc obliczeniowa stanowią zagrożenie dla wydajności inteligentnych systemów, w których duże modele językowe (LLM) okazały się idealnym rozwiązaniem. LLM to generatywne systemy AI, które rozumieją język ludzki i dostarczają inteligentnych, kreatywnych rozwiązań. Złożona architektura LLM pozwala im precyzyjniej uchwycić zawiłości języka, umożliwiając generowanie spójnych i kontekstowo istotnych wyników. W tym rozdziale zagłębiamy się w kompleksową analizę znanych modeli generatywnej sztucznej inteligencji (AI), takich jak generatywne sieci adwersaryjne, transformery i LangChain. Generatywna sztuczna inteligencja (AI) wykorzystuje różne techniki szkoleniowe, takie jak uczenie przez wzmacnianie, trening adwersaryjny, wnioskowanie wariacyjne, transfer uczenia i trening progresywny w różnych dziedzinach zastosowań. Ponadto, badanie analizuje kluczowy aspekt oceny skuteczności modeli generatywnych, wykorzystując różnorodne metryki, takie jak BLUE, wynik rozpoczęcia, perpleksja, odległość rozpoczęcia Frecheta, precyzja, ROUGE, recall, METEOR, BERT, MoverScore i wiele innych. Analiza porównawcza tych metryk pozwala zrozumieć ich zalety i wady, pomagając praktykom i badaczom w wyborze benchmarków, które odpowiadają ich konkretnym przypadkom użycia.

Słowa kluczowe: Generatywna sztuczna inteligencja (AI), LLM, transformery, BERT, VAE, GAN, LangChain,

Wprowadzenie

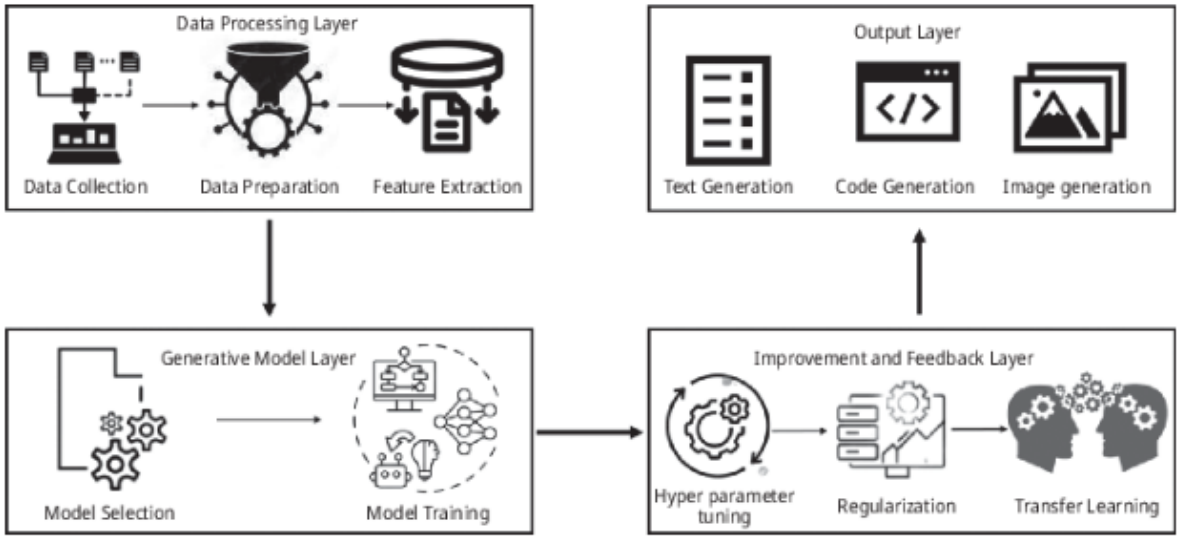
Rysunek przedstawia różne warstwy systemów inteligentnych.



Generatywna sztuczna inteligencja (AI) to rodzaj uczenia maszynowego, który wykazuje zdolność do tworzenia treści w reakcji na polecenia, których długość i złożoność mogą być różne – od krótkich i nieskomplikowanych po obszerne i kompleksowe. Potencjał generatywnej AI do tworzenia tekstów, obrazów, muzyki, opowiadań, powieści i filmów imitujących ludzi sprawił, że świat przypomina burzę. Wykorzystując techniki takie jak generatywne sieci przeciwstawne (GAN), modele autoregresyjne (ARM) i autoenkodery wariacyjne (VAE), syntetyzują nowe próbki w oparciu o odkryte wzorce i rozkład prawdopodobieństwa. Modele generatywnej AI uczą się złożonych koncepcji, które można zastosować w różnych kontekstach, trenując na dużych zbiorach danych o wielu parametrach. Synteza obrazu wykorzystuje model dyfuzji do generowania wysokiej jakości obrazów; podobnie streszczenia tekstu, proza i poezja mogą być generowane za pomocą dużych modeli językowych (LLM). Sieci neuronowe stanowią fundament generatywnych modeli sztucznej inteligencji (AI), które generują bazy wiedzy poprzez analizę ogromnych ilości danych. Następnie wykorzystują zdobytą wiedzę do generowania nowych danych, które ściśle odzwierciedlają rozkład. Zdolność generatywnej AI do adaptacji i ewolucji podczas analizy nowych danych wejściowych jest kluczową cechą, która pozwala systemowi stale zwiększać wydajność i poszerzać swoje koncepcje. Do najważniejszych przykładów w tej dziedzinie należą DALL-E i Midjourney, które koncentrują się na tworzeniu wizualizacji, oraz ChatGPT, usługa modelu językowego, która została wytrenowana na ogromnych ilościach danych. Generatywne modele AI, takie jak modele językowe, modele obrazkowe i modele wideo, są klasyfikowane według konkretnego typu generowanych danych wyjściowych. Zdolność generatywnych modeli AI do generowania oryginalnych instancji sprawia, że są one niezbędne do zadań takich jak podsumowywanie, generowanie treści, manipulacja obrazami, produkcja wideo, prognozowanie finansowe, wykrywanie oszustw i diagnostyka zdrowotna.

Warstwy generatywnego modelu AI

Rysunek 2 ilustruje podstawowe sekwencyjne procesy generatywnego modelu AI i przedstawia ogólną architekturę warstwową modelu.



Początkowa faza generatywnego modelu AI obejmuje gromadzenie danych z różnych źródeł, w tym baz danych, stron internetowych, platform mediów społecznościowych i interfejsów API. W fazie przygotowawczej dane są oczyszczane i normalizowane w celu usunięcia wszelkich niespójności, błędów i duplikatów. Następnie dane są przekształcane do formatu sprzyjającego analizie. Ekstrakcja cech obejmuje identyfikację istotnych cech, które są niezbędne do ekstrakcji wzorców. Warstwa modelu generatywnego odpowiada za tworzenie nowej treści za pomocą odpowiednich modeli, takich jak modele głębokiego uczenia, na przykład CNN, RNN, GANS, które są dość skuteczne w generowaniu obrazów, wideo i dźwięku. Podobnie, modele wzmacniania mogą być wykorzystywane do generowania odpowiedzi na podpowiedzi, a algorytmy genetyczne są w stanie ewoluować rozwiązania skomplikowanych problemów, generując dane lub treści, które są stale ulepszane w czasie. Model jest trenowany z wykorzystaniem znacznej ilości danych, co pozwala mu zrozumieć złożone wzorce i tworzyć nowe materiały. Dzięki sprzężeniu zwrotnemu model ewoluuje, generując coraz dokładniejsze i bardziej precyzyjne informacje. Warstwa ulepszeń i sprzężenia zwrotnego gromadzi komentarze użytkowników i analizuje uzyskane dane, aby zwiększyć wydajność systemu, co pomaga w dostrojeniu modelu, aby był bardziej precyzyjny. Optymalizacja modelu obejmuje dostrajanie hiperparametrów, stosowanie metod regularyzacji i wykorzystanie transferu danych. Dostosowanie hiperparametrów, takich jak tempo uczenia się, rozmiar partii i optymalizator, może poprawić wydajność modelu. Regularyzacja L1 i L2 łagodzi nadmierne dopasowanie i zwiększa zdolność modelu do generalizacji. Transfer danych to ekonomiczne i czasowo efektywne podejście, które polega na dostosowywaniu wstępnie wytrenowanych modeli do unikalnych zastosowań. Ostatnią fazą jest generowanie nowej treści na podstawie wprowadzonych zmian w modelu, aby zapewnić, że wygenerowana treść ma wewnętrzne cechy treści rzeczywistej, a jednocześnie różni się od treści oryginalnej.

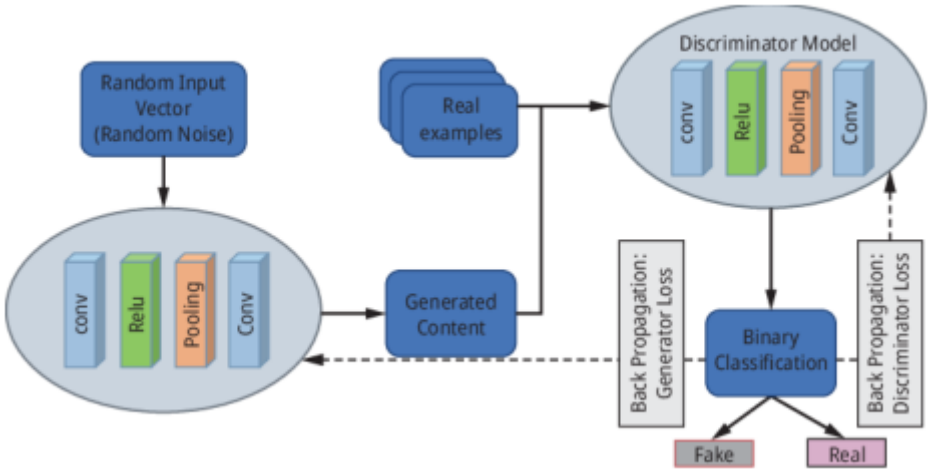
Model generatywnej sztucznej inteligencji i techniki szkoleniowe LLM

Szeroki wachlarz technik szkoleniowych, w tym uczenie głębokie, transfer wiedzy i uczenie wzmacniające, jest wykorzystywany w generatywnej sztucznej inteligencji i LLM do tworzenia realistycznych i wysokiej jakości treści.

Sieci generatywne przeciwstawne (GAN)

GAN to forma modelowania generatywnego, która po otrzymaniu danych treningowych generuje nowe dane, które ściśle odzwierciedlają ich wewnętrzne cechy. Dwie głębokie sieci neuronowe

stanowią szkielet sieci GAN. Działają one w tandemie, wykrywając, replikując i analizując zmienność w zbiorze danych. Architekturę podstawowego modelu GAN przedstawiono na rysunku 3.



Modelowanie generacyjne wykorzystuje nienadzorowane uczenie maszynowe do automatycznego wyszukiwania i uczenia się regularności lub wzorców w danych wejściowych w celu wygenerowania lub wygenerowania nowych przykładów z oryginalnego zbioru danych. Sieci GAN wykorzystują sieci neuronowe do osiągnięcia równowagi między generatorem a dyskryminatorem. Sieć generatora wykorzystuje losowy wektor jako dane wejściowe i próbuje syntetyzować podobne dane, aby wprowadzić dyskryminator w błąd. Dyskryminator wykorzystuje rzeczywiste dane do porównania swoich właściwości z danymi wyjściowymi generatora i sklasyfikowania ich jako rzeczywistych lub fałszywych. Trening adwersarny pozwala obu sieciom stopniowo się rozwijać, a generator nabywa zdolność generowania coraz bardziej realistycznych próbek. Obie sieci współpracują, aby odtworzyć początkowy rozkład, wykorzystując funkcję straty, która dokładnie odzwierciedla rozbieżność między rozkładem rzeczywistym a fałszywym. Głównym celem generatora jest minimalizacja funkcji straty, podczas gdy głównym celem dyskryminatora jest jej maksymalizacja. Wagi generatora są aktualizowane poprzez gradient spadkowy przez dyskryminator i z powrotem do generatora, na podstawie oszacowanej straty, z wykorzystaniem techniki propagacji wstecznej. Generator staje się lepszy w tworzeniu zwodniczych danych fałszywych, podczas gdy dyskryminator lepiej identyfikuje dane rzeczywiste. Podobnie, waga dyskryminatora jest modyfikowana w oparciu o wynik funkcji straty. Funkcje straty dla generatora i dyskryminatora podano odpowiednio w równaniach (3.1) i (3.2):

$$\min_G V(G) = \nabla_{\theta_s} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^i\right)\right)\right) \tag{3.1}$$

$$\max_D V(D) = \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^i\right) + \log\left(1 - D\left(G\left(z^i\right)\right)\right)\right] \tag{3.2}$$

gdzie D odnosi się do prawdopodobieństwa dyskryminatora, że x jest rzeczywiste, a D(G(z)) do prawdopodobieństwa dyskryminatora, że wygenerowana treść G(z) jest rzeczywista. GAN wykorzystuje głównie dwie różne funkcje straty, tj. minimaks i funkcję straty Wassersteina. Minimaks ma swoje korzenie w teorii gier, gdzie gracz stara się zmaksymalizować swoje prawdopodobieństwo i zminimalizować prawdopodobieństwo przeciwnika. Funkcja straty Wassersteina mierzy odległość między rozkładem rzeczywistym a fałszywym w kategoriach wysiłku wymaganego do transformacji, wykorzystując odległość Earth Mover. Gdy wagi do zaktualizowania po obliczeniu funkcji straty są ekstremalnie małe, wyniki są zerowe. Aby temu zaradzić, wprowadzono różne warianty GAN, takie jak

głęboki GAN splotowy, GAN warunkowy, GAN cykliczny, GAN Pix2Pix, GAN skumulowany i GAN waniliowy.

Warunkowy GAN

Warunkowy GAN to technika uczenia nadzorowanego, która wykorzystuje zarówno dane oznaczone, jak i nieoznaczone . Generator i dyskryminator są trenowane z wykorzystaniem warunkowości – etykiet dla każdej instancji. Na przykład podczas tworzenia obrazów, warunkiem może być opisowy podpis reprezentujący zawartość obrazu. Warunkowanie umożliwia generatorowi generowanie danych spełniających kryteria. Dyskryminator wykorzystuje informacje pomocnicze c do odróżnienia wygenerowanych danych D(G(z|c)) od danych rzeczywistych. G(z|c) jest generowane przy użyciu etykiety klasy i wektora ukrytego jako informacji warunkowej, aby dopasować dane rzeczywiste. Wykorzystując zarówno dane oznaczone, jak i nieoznaczone, model może zdobywać wiedzę i zmniejszać zależność od oznaczonych danych treningowych. Jednak takie podejście może skutkować generowaniem przez model danych, które są bardzo zbliżone do danych treningowych, co ogranicza jego zdolność do generalizacji na nowe, nieznane dane.

Głęboka splotowa sieć neuronowa GAN (DCGAN)

Głęboka splotowa sieć neuronowa GAN (DCGAN) to odmiana sieci GAN, w której generator i dyskryminator wykorzystują głęboką splotową sieć neuronową; dlatego nadaje się ona przede wszystkim do obrazów wysokiej jakości. Sekwencja warstw splotowych i transponowanych zwiększa głębię mapy cech i rozdzielczość obrazu, ponieważ generator DCGAN przekształca losowy szum w obraz wysokiej jakości. Rzeczywisty lub utworzony obraz jest wykorzystywany jako dane wejściowe przez dyskryminator, który następnie wykorzystuje warstwy splotowe do ekstrakcji cech z obrazu. Cechy te są następnie przetwarzane w celu określenia prawdopodobieństwa, że obraz jest rzeczywisty. Sieć generatorów wykorzystuje aktywację ReLU (Rectified Linear Unit), z wyjątkiem ostatniej warstwy, gdzie aktywacja tanh jest używana do zapewnienia zakresu wyników od −1 do 1. Podobnie dyskryminator wykorzystuje nieszczelną aktywację ReLU wraz z optymalizatorem Adama ze współczynnikiem uczenia 0,0002.

Pix2Pix GAN

Pix2Pix GAN to specjalny typ sieci CGAN, który koncentruje się na translacji obrazu-obrazu, takiej jak konwersja obrazu o niskiej rozdzielczości na obraz o wysokiej rozdzielczości, obrazu z widzenia dziennego na obraz w nocy itd. Pix2Pix GAN wykorzystuje parę obrazów, np. obraz obiektu o niskiej rozdzielczości jest podawany jako sygnał wejściowy do sieci generatorów, która jest siecią U. Sieć generatorów odbiera obraz wejściowy z określonej domeny i przekształca go w obraz w skali szarości lub obraz kolorowy, w zależności od określonych kryteriów. Dyskryminator ocenia oryginalny obraz i wyjście generatora, wykorzystując technologię PatchGAN, która analizuje obrazy w małych fragmentach, zamiast korzystać z całego obrazu. Pix2Pix wykorzystuje stratę adwersarną, aby zachęcić generator do dokładnego odtworzenia oryginalnego obrazu.

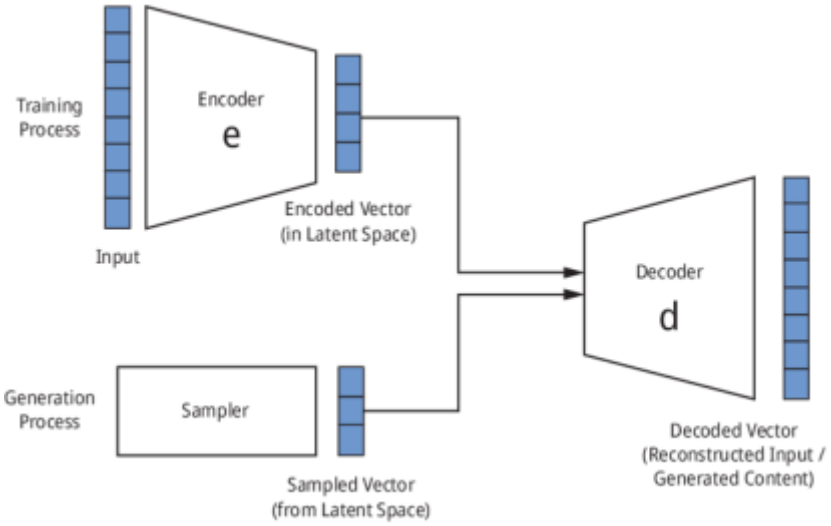
Cycle GAN

CycleGAN to zaktualizowana wersja Pix2Pix GAN, w której translacja obrazu na obraz odbywa się bez sparowanego obrazu. Wykorzystuje ona dwie sieci GAN do uczenia się cech dwóch różnych domen obrazu. Model jest trenowany w celu dokładnego uchwycenia charakterystycznych cech pożądanej domeny i generowania nowych obrazów z danej domeny, które posiadają identyczne cechy. Aby uruchomić model, trenujemy obie sieci GAN niezależnie. Na początek obraz w domenie źródłowej jest używany do trenowania generatora pierwszej sieci GAN w celu wygenerowania obrazu w domenie

docelowej. Na podstawie wygenerowanych obrazów w domenie docelowej, dyskryminator pierwszej sieci GAN jest trenowany w celu wykrywania rzeczywistych obrazów w domenie docelowej. W międzyczasie drugi generator GAN uczy się pobierać obrazy w domenie docelowej i przekształcać je w obrazy w domenie źródłowej. Dwie sieci GAN współpracują ze sobą, tworząc dyskryminator, który potrafi odróżnić wygenerowane obrazy w domenie źródłowej od rzeczywistych. CycleGAN jest definiowany po niezależnym trenowaniu obu sieci GAN.

Autokoder wariacyjny

Rysunek 4 przedstawia podstawowe kroki zawarte w VAE.

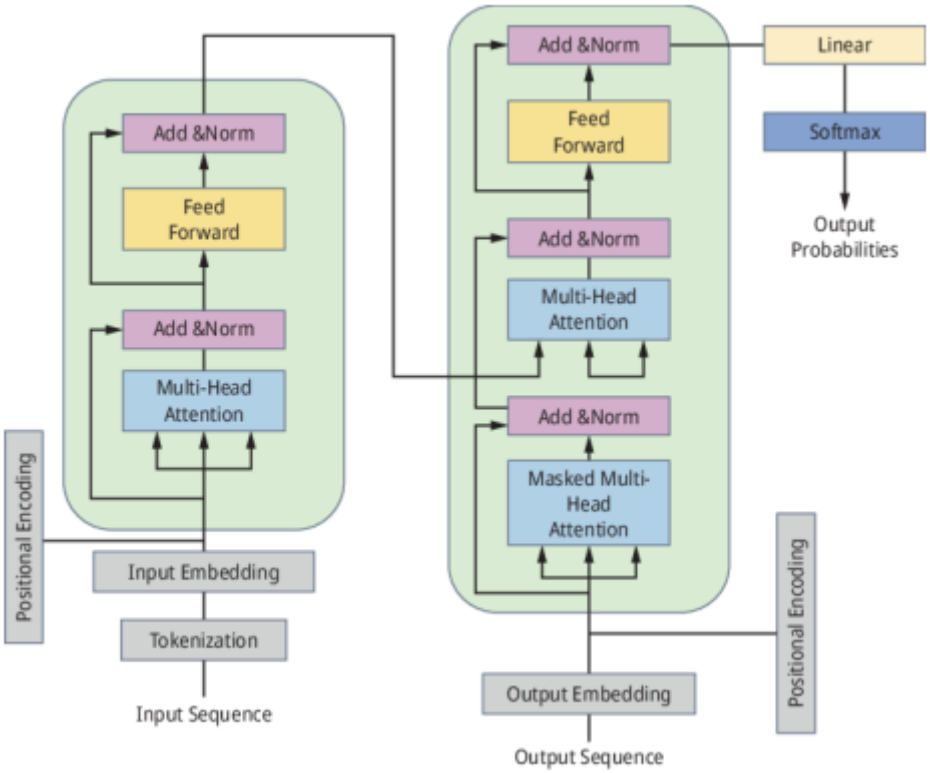


Typ modelu generatywnego zwany VAE został stworzony specjalnie w celu generowania nowych próbek i reprezentowania rozkładu prawdopodobieństwa danego zbioru danych .Generowanie danych jest powszechnym problemem w różnych zastosowaniach, w tym w systemach wizyjnych, przetwarzaniu języka naturalnego (NLP) i medycynie. Polega ono na tworzeniu realistycznych próbek z danych treningowych. Koder, czyli model rozpoznający, i dekoder, czasami nazywany modelem generującym, to dwa połączone, ale oddzielnie sparametryzowane modele, które tworzą VAE. Sieci neuronowe stanowią podstawę architektury VAE, która obejmuje koder lub model rozpoznający do przechwytywania reprezentacji próbek wejściowych oraz dekoder lub model generatora do generowania nowych próbek w oparciu o właściwości uczenia. Koder minimalizuje wektory cech danych wejściowych, aby zmniejszyć wymiarowość. Tradycyjne autoenkodery reprezentują dane wejściowe w sposób unikatowy za pomocą kodowań deterministycznych. Ta sztywność utrudnia im uchwycenie niejednoznaczności i zmienności danych rzeczywistych. VAE rozwiązują ten problem poprzez probabilistyczne modelowanie przestrzeni ukrytej. Odwzorowujemy dane na rozkład prawdopodobieństwa, aby uwzględnić niepewność i zmienność. Modele VAE zachowują wyuczony rozkład danych, generując jednocześnie różnorodne wyniki dzięki elastyczności probabilistycznej przestrzeni utajonej. Kodery VAE tworzą rozkład prawdopodobieństwa w przestrzeni utajonej zamiast regularyzowanego wyniku, aby zminimalizować nadmierne dopasowanie i zapewnić solidne charakterystyki procesu generatywnego. Koder zapewnia probabilistyczne kodowanie utajone, umożliwiając modelowi VAE wyrażenie kilku reprezentacji w przestrzeni utajonej. Dekoder rekonstruuje próbkowany punkt rozkładu utajonego w przestrzeni danych. Aby zminimalizować straty rekonstrukcji, model optymalizuje parametry kodera i dekodera podczas treningu. Celem jest dokładna rekonstrukcja i regularyzacja przestrzeni utajonej, aby zachować rozkład. Strata rekonstrukcji zmusza model do dokładnej rekonstrukcji danych wejściowych, a człon regularyzacji kieruje przestrzeń utajoną

do pożądanego rozkładu, redukując nadmierne dopasowanie i sprzyjając generalizacji. Model VAE uczy się kodowania danych wejściowych do sensownej reprezentacji przestrzeni utajonej poprzez iteracyjne dostosowywanie tych parametrów podczas treningu. Ten ulepszony kod utajony przechwytuje cechy i struktury danych, co zapewnia dokładniejszą rekonstrukcję. Losowe punkty z rozkładu uczącego się można wykorzystać do tworzenia unikatowych próbek w probabilistycznej przestrzeni ukrytej.

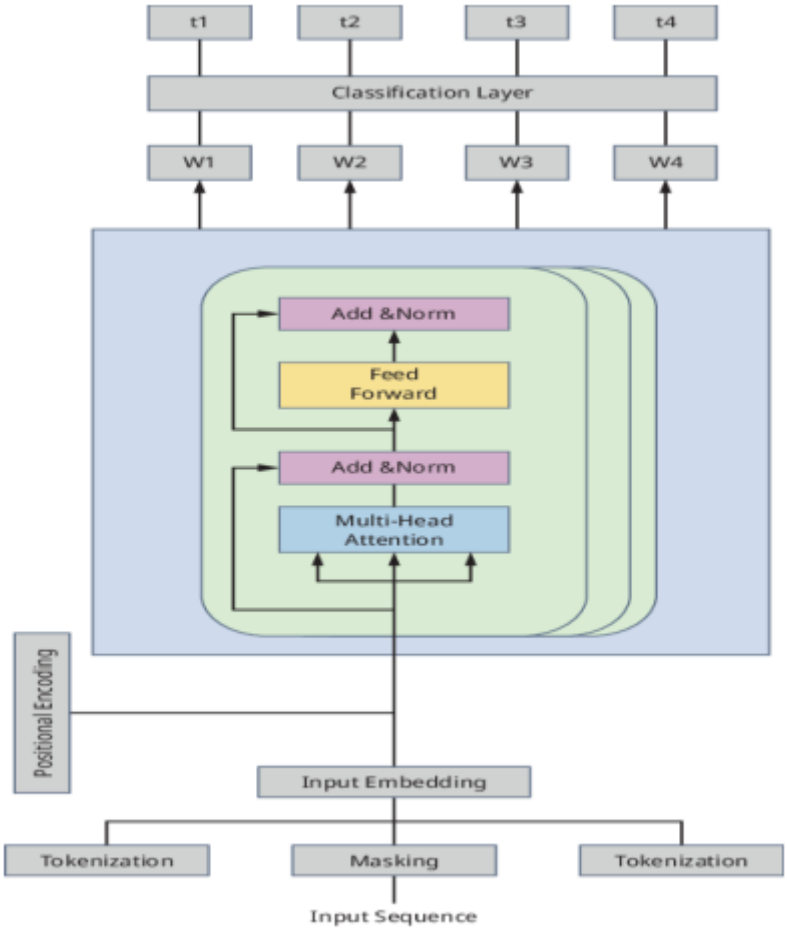
Modele Transformerów

Transformer to model oparty na LLM, który wykorzystuje architekturę sieci neuronowej z mechanizmem samouwagi. Ich zdolność do sekwencyjnego przetwarzania danych sprawia, że idealnie nadają się do generowania tekstu, jego podsumowywania, odpowiadania na pytania, tłumaczeń językowych i rozpoznawania obrazów . Modele Transformerów posiadają zdolność rozumienia powiązań między słowami w zdaniu, ułatwiając w ten sposób przechwytywanie informacji kontekstowych. W przeciwieństwie do konwencjonalnych modeli, które sekwencyjnie obsługują sekwencje, Transformery mogą przetwarzać wszystkie komponenty jednocześnie. Podstawową architekturę modelu Transformera przedstawiono na rysunku 5.



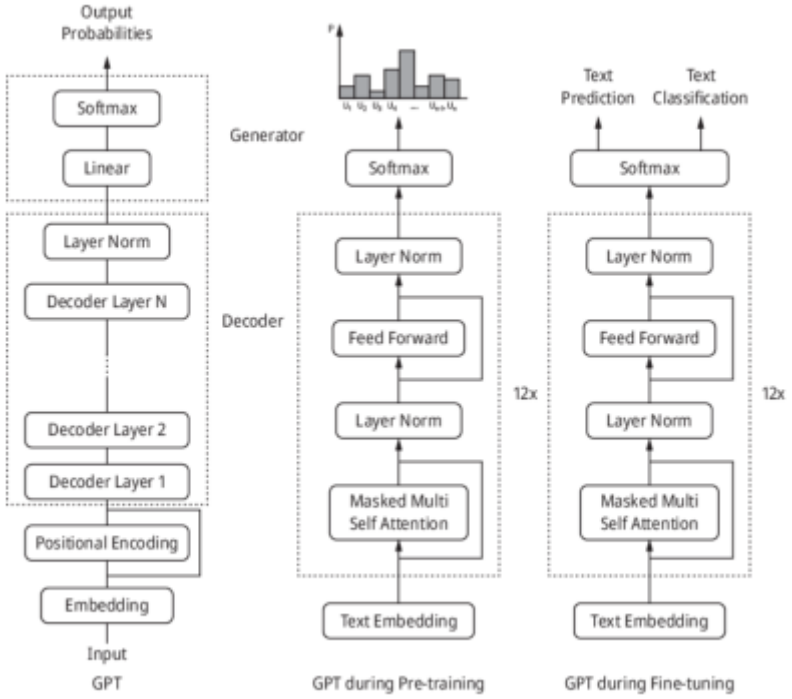
Zazwyczaj większość sieci neuronowych przetwarza tekst wejściowy sukcesywnie, co prowadzi do problemu zanikającego gradientu. Aby rozwiązać ten problem, model transformatora wykorzystuje mechanizmy samouwagi, umożliwiając sieci przetwarzanie całej sekwencji wejściowej jednocześnie, co czyni go efektywnym w obsłudze zadań opartych na przetwarzaniu języka naturalnego. Architektura transformatora składa się z kodera i dekodera. Koder przetwarza przychodzące dane i może wyodrębnić z nich informacje semantyczne i kontekstowe, wykorzystując tokenizację i osadzanie. Sekwencja wejściowa, składająca się z serii słów, jest początkowo tokenizowana na znaczące słowa lub podsłowa. Po tokenizacji, dane wyjściowe tokenizatora są przekształcane na wektory numeryczne poprzez osadzanie, które uwzględnia kontekst każdego tokena. Te osadzania pozwalają modelom matematycznie interpretować teksty i rozumieć bogate niuanse i powiązania języka, a także zapewniają podobne osadzanie podobnych tokenów. Model wykorzystuje osadzania pozycyjne, aby

uchwycić pozycję każdego tokena w sekwencji, gwarantując, że kolejność i względne pozycje tokenów są uwzględniane podczas przetwarzania. Koder przetwarza tokeny za pomocą układów samouwagi i sprzężenia zwrotnego. Sieci sprzężenia zwrotnego wykorzystują wskaźniki samouwagi do poprawy rozumienia słów. Zapewnia to dokładne zbieranie skomplikowanych szczegółów i dostarczanie ich do dekodera. W przypadku danych wyjściowych dekoder jest skonfigurowany podobnie jak koder. Przetwarzanie z samouwagą łączy każde słowo w sekwencji wejściowej, tworząc osadzenia, co pozwala nam nadawać priorytet ważnym słowom. Mechanizm samouwagi łączy każde słowo wejściowe z kolejnym, koncentrując się na najważniejszych terminach. Wartość liczbowa każdego słowa wskazuje jego znaczenie dla pozostałych słów w zdaniu. Dekoder inicjuje swoje pierwsze dane wejściowe za pomocą osadzenia wyjściowego z poprzedniego kroku czasowego oraz przetworzonej sekwencji wejściowej kodera. Ta technika podwójnego wejścia gwarantuje, że dekoder uwzględnia zarówno dane początkowe, jak i dane wyjściowe do tego punktu. Celem jest wygenerowanie spójnej i kontekstowo odpowiedniej sekwencji wyników końcowych. Optymalizację wydajności modelu można osiągnąć poprzez dobór odpowiednich hiperparametrów, takich jak liczba warstw kodera lub dekodera, liczba mechanizmów samouwagi oraz wymiary sieci neuronowej z sprzężeniem zwrotnym. BERT (Bidirectional Encoder Representations from Transformers), generatywny wstępnie wyszkolony transformator (GPT), transformator XL, XLNet i Roberta to przykłady modeli transformatorów.BERT to wstępnie wytrenowany model transformatorowy, zaprojektowany specjalnie do zadań NLP. BERT zawiera standardowy koder, a ponadto maskowany model języka (MLM) i predykcję następnego zdania (NSP), co poprawia wydajność modelu [→17]. Podobnie jak w modelu transformatorowym, sekwencja wejściowa jest parsowana do postaci tokena, a następnie konwertowana na wektor numeryczny za pomocą osadzania, a na koniec wykonywane jest osadzanie pozycyjne w celu uchwycenia kolejności tokenów i względnej pozycji w sekwencji. Typowy model transformatorowy BERT przedstawiono na rysunku 6.



Zazwyczaj model transformatorowy przewiduje kolejne słowo w sekwencji, podążając za określonym kierunkiem. Jednak ta metoda może ograniczać zdolność modelu do uczenia się i rozumienia kontekstu. Aby przezwyciężyć tę wadę, BERT wykorzystuje MLM i NSP. Procedura wstępnego trenowania rozpoczyna się od selektywnego maskowania określonych słów w każdej sekwencji wejściowej. Następnie metoda wykorzystuje techniki oparte na prawdopodobieństwie do przewidywania zamaskowanych słów na podstawie informacji kontekstowych z otaczających je słów. Główny nacisk położony jest na słowo maskowane, co pomaga modelowi zrozumieć kontekst słów. Posiada on warstwę klasyfikacji na warstwie wyjściowej kodera, która wspomaga przewidywanie słów maskowanych. Wymiar słownikowy jest tworzony poprzez pomnożenie wektorów wyjściowych warstwy klasyfikacji przez macierz osadzania, co pomaga dopasować przewidywany wektor do przestrzeni słownikowej. Następnie prawdopodobieństwo słów jest obliczane za pomocą aktywacji SoftMax. Ten etap zapewnia rozkład prawdopodobieństwa w całym słowniku dla każdego miejsca maskowanego. W funkcji straty treningowej uwzględniana jest tylko prognoza wartości maskowanej. Jeśli jej prognozy różnią się od rzeczywistych wartości słów maskowanych, model jest karany. Ponieważ BERT przewiduje tylko wartości maskowane, konwergencja modelu spowalnia, co zwiększa świadomość kontekstu, aby zrekompensować wolniejszą konwergencję. NSP określa, czy druga fraza jest kontynuacją, czy też łączy pierwszą, konwertując token wyjściowy na wektory i używając SoftMax do obliczenia prawdopodobieństwa połączenia zdań. Zarówno MLM, jak i NSP są trenowane jednocześnie w celu zminimalizowania połączonej funkcji straty. Takie podejście skutkuje solidnym modelem, który może skutecznie interpretować kontekst. Generatoratywny, wstępnie wytrenowany transformator, często znany jako GPT, to oparty na transformatorach model LLM, który wykorzystuje sieci neuronowe do tworzenia tekstów i materiałów o charakterze ludzkim (obrazów, muzyki i innych), a także odpowiada na pytania w sposób konwersacyjny. Oceniają one żądania języka naturalnego, czyli

podpowiedzi, i szacują najlepszą prawdopodobną odpowiedź na podstawie swojej znajomości języka. Modele GPT wykorzystują w tym celu setki miliardów parametrów trenowanych na ogromnych zbiorach danych lingwistycznych. Potrafią uwzględniać kontekst wejściowy i dynamicznie reagować na różne sekcje danych wejściowych, co pozwala im generować dłuższe odpowiedzi, a nie tylko kolejne słowo w serii. Przekształcony model GPT przedstawiono na rysunku 7.

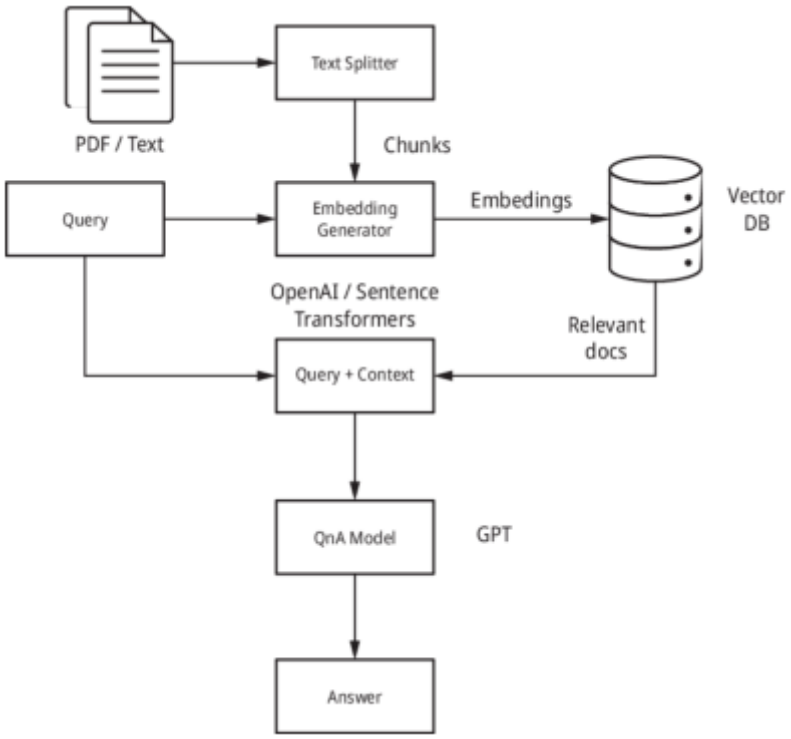


Model GPT składa się z trzech głównych komponentów: generatora, wstępnie wytrenowanego i transformatorów. Cecha generatora podkreśla zdolność modelu do generowania tekstu poprzez rozumienie i reagowanie na dostarczoną próbkę tekstu. Przed opracowaniem modeli GPT słowa były ekstrahowane lub przekształcane z danych wejściowych w celu wygenerowania tekstu wyjściowego. Modele GPT były w stanie tworzyć bardziej spójną i ludzką prozę niż poprzednie modele, dzięki swojej mocy generatywnej. Do trenowania modeli GPT wykorzystywane jest autoregresyjne modelowanie języka. Biorąc pod uwagę sekwencję słów jako dane wejściowe, model wykorzystuje rozkłady prawdopodobieństwa do prognozowania najbardziej prawdopodobnego słowa lub frazy, aby zidentyfikować najbardziej odpowiednie słowo jako następne. Transformatory to typowy projekt sieci neuronowej dla tekstu o zmiennej długości. GPT ma architekturę wyłącznie dekoderową, w której najważniejsza jest samouwaga. Każde słowo w sekwencji jest porównywane z innymi w zdaniu. Na podstawie kontekstu, model przewiduje kolejne najbardziej prawdopodobne słowo, wykorzystując zdekodowany tekst wejściowy. Faza wstępnego trenowania buduje modele uczenia maszynowego bez nadzoru, wykorzystując dużą ilość danych. GPT jest trenowany na dużym korpusie danych tekstowych z różnych źródeł. Dzięki temu model może znajdować wzorce i korelacje danych bez bezpośredniego sterowania. Model może wykorzystać swoją wiedzę o strukturze i atrybutach językowych do wykonywania zadań, takich jak odpowiadanie na pytania i podsumowywanie. Trening polega na osadzeniu i podziale sekwencji wejściowej na podciągi o długości k. Następnie model jest proszony o przewidzenie następnego tokena dla każdego podciągu, wykorzystując wyjściowy rozkład prawdopodobieństwa dla wszystkich tokenów słownikowych. Ten rozkład pokazuje prawdopodobieństwo, że każdy token jest właściwym podciągiem następującym po tokenie. Po wstępnym wytrenowaniu GPT może zostać użyty do generowania tekstu. GPT jest algorytmem ARM, co oznacza, że wykorzystuje wcześniej przewidziane tokeny jako dane wejściowe do przewidywania

następnego tokena. W każdej iteracji GPT pobiera początkową sekwencję i przewiduje dla niej kolejny, najbardziej prawdopodobny token. Następnie sekwencja i przewidywany token są łączone i wykorzystywane jako dane wejściowe do przewidywania następnego tokena. Proces jest kontynuowany do momentu przewidzenia tokena [koniec] lub osiągnięcia maksymalnego rozmiaru danych wejściowych. Po przechwyceniu lingwistyki języka przez GPT, musi on zostać dostrojony do nadzorowanego zadania. Akceptuje on zestaw danych z etykietami, przy czym każdy przykład zawiera sekwencję wejściową x i pasującą do niej etykietę y, które muszą zostać przewidziane. Każdy przykład jest przesyłany przez model, który generuje ukryte reprezentacje h na ostatniej warstwie. Wygenerowane wektory są następnie przesyłane do dodatkowej warstwy liniowej z parametrami W, które można nauczyć, a następnie do warstwy softmax. Ogólnie rzecz biorąc, skuteczność GPT wynika z zastosowania silnej architektury transformatorowej, wstępnego trenowania z wykorzystaniem maskowanego modelowania języka oraz jednokierunkowej konstrukcji zaprojektowanej do tworzenia tekstu. Te komponenty umożliwiają GPT generowanie innowacyjnej i gramatycznie akceptowalnej prozy, co czyni go ważnym narzędziem w różnorodnych zastosowaniach NLP.

LangChain

LangChain to framework do tworzenia aplikacji z wykorzystaniem LLM. Łączy się on z modułem LLM w celu generowania odpowiedzi, gdy użytkownik wyśle żądanie. Może również integrować wiedzę i dane z innych źródeł, takich jak dokument lub baza danych, aby oferować dokładniejsze i kontekstowo adekwatne odpowiedzi. Zwiększa on inteligencję aplikacji i zdolność do obsługi skomplikowanych i niestandardowych zapytań. Podstawową architekturę modelu LangChain dla systemu zapytań/odpowiedzi zilustrowano na rysunku 8.

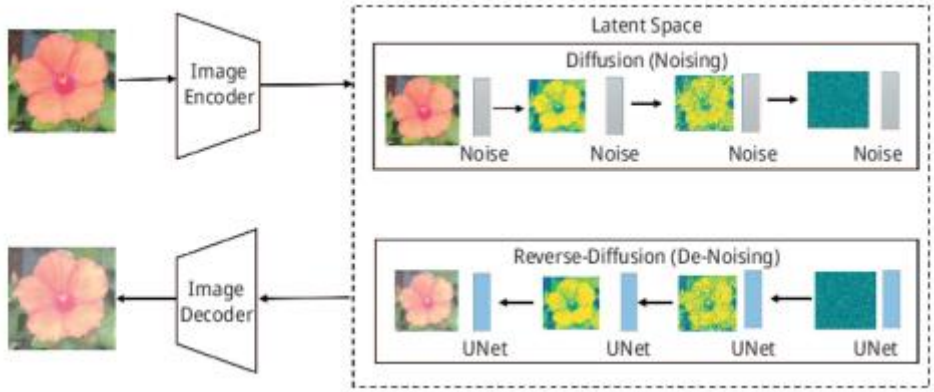


Model LangChain obejmuje LLM, szablony monitów, indeksy, parsery, magazyny wektorów i agentów. LangChain opiera się na modelu LLM jako głównym źródle generowania odpowiedzi, które mogą być używane do generowania tekstu, odpowiadania na zapytania, tłumaczenia języka, podsumowywania i

wielu innych zadań. Szablony monitów służą przede wszystkim do formatowania zapytań użytkownika w sposób zrozumiały dla LLM. Opisują one kontekst istotny dla danych wprowadzanych przez użytkownika lub zadania LLM. Bazy danych zwane indeksami zawierają szczegółowe informacje na temat zbioru treningowego LLM. Dokumenty, powiązania i informacje zawarte w dokumentach mogą być zawarte w tych danych. Parser formatuje wygenerowaną odpowiedź, strukturyzując ją poprzez ograniczenie jej do pożądanych informacji. Magazyny wektorowe przechowują matematyczne reprezentacje słów i zdań. Agenci podają uzasadnienie pytań, dzielą je na mniejsze zadania i kierują łańcuch do niezbędnych zadań. Początkowo dokumenty i powiązane źródła są konwertowane na reprezentację wektorową za pomocą osadzania. Do generowania osadzania wykorzystywane są techniki takie jak transformator, BERT lub inne modele oparte na transformacjach. Osadzanie to po prostu numeryczna reprezentacja danych tekstowych, która poprawia wydajność podczas przechowywania i pobierania. Osadzenia są zapisywane w bazie danych wektorowej, takiej jak Pinecone lub Chroma DB. LangChain umożliwia stosowanie różnych strategii pobierania po umieszczeniu danych w bazie danych. Należą do nich podstawowe wyszukiwanie semantyczne, wyszukiwanie dokumentów nadrzędnych, wyszukiwanie z własnym zapytaniem, wyszukiwanie zespołowe i inne. Podczas wyszukiwania, system wyszukiwania przyznaje każdemu dokumentowi punktację lub ocenę na podstawie jego trafności w kontekście zapytania. Następnie dokumenty są sortowane w kolejności malejącej, a te najbardziej trafne pojawiają się na górze listy. Na podstawie zapytania użytkownika, ocena ta pozwala systemowi szybko zlokalizować i uzyskać najbardziej trafne informacje. System rankingowy wykorzystuje kilka zmiennych, w tym aktywność użytkownika, popularność dokumentu, dopasowanie słów kluczowych i inne sygnały trafności, aby określić kolejność wyświetlania wyników. Na koniec, wykorzystując reprezentację kontekstową, model LLM generuje odpowiedź na podstawie tych wyników.

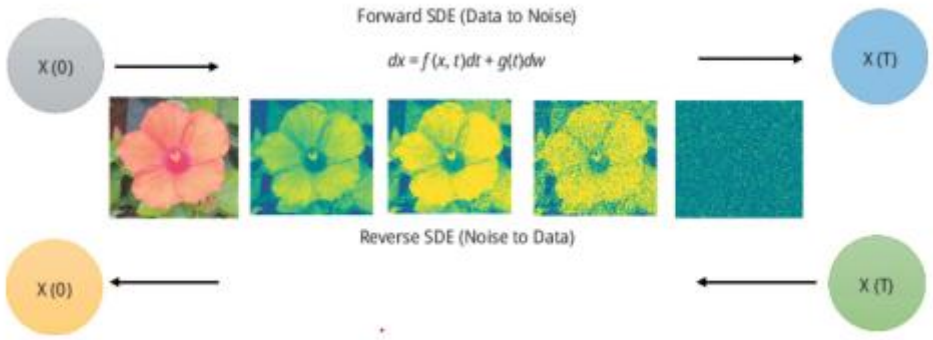
Model dyfuzji

Modele dyfuzji to rodzaj generatywnego modelu sztucznej inteligencji (AI), który umożliwia generowanie nowej treści, takiej jak tekst i obrazy. Modele dyfuzji działają poprzez usuwanie danych treningowych poprzez stopniowe dodawanie szumu gaussowskiego, a następnie uczenie się odzyskiwania danych poprzez odwrócenie procesu szumu. Model dyfuzji to model zmiennych ukrytych, który odwzorowuje przestrzeń utajoną za pomocą stałego łańcucha Markowa. Łańcuch ten stopniowo dodaje szum do danych, aby uzyskać oszacowany wynik a posteriori. Celem dyfuzji jest nauczenie modelu procesu odwrotnego generowania obrazu poprzez przechodzenie wstecz wzdłuż łańcucha. Rysunek 9 ilustruje ogólny model generowania treści oparty na dyfuzji.



Proces dyfuzji, który wyjaśnia ruch cząstek z obszaru gęstego do obszaru o mniejszej gęstości, stanowi podstawę modeli dyfuzyjnych. Model dyfuzyjny obejmuje dwa procesy: wstępne trenowanie modelu

dyfuzyjnego oraz przeprowadzanie wnioskowania na modelu dyfuzyjnym. Do trenowania modelu dyfuzyjnego, w oparciu o cel modelu, który wymaga tysięcy podobnych obrazów, wymagany jest duży zbiór danych obrazów. Zaczynając od oryginalnego obrazu, proces trenowania przesyła go przez koder obrazu, który konwertuje go z formatu wielowymiarowego na niskowymiarowy, usuwając szum i rejestrując istotne szczegóły. Po odebraniu obrazu koder tworzy wektor liczb reprezentujących jego cechy. Obraz jest teraz przechowywany w wielowymiarowej przestrzeni zwanej przestrzenią utajoną, gdzie każdy punkt reprezentuje inną cechę. Kolejny etap nazywa się dyfuzją, w którym oryginalny obraz jest wielokrotnie poddawany działaniu szumu w celu jego zniekształcenia. Termin „szum" opisuje dowolne wariacje lub zniekształcenia stosowane do obrazu, stopniowo zaciemniające oryginalne szczegóły, aż do ich utraty. Proces ten trwa do momentu, aż obraz zamieni się w całkowity szum, co pomaga modelowi nauczyć się procesu odwrotnego. Rysunek 10 ilustruje proces odszumiania i odszumiania obrazów podczas fazy wstępnego treningu.

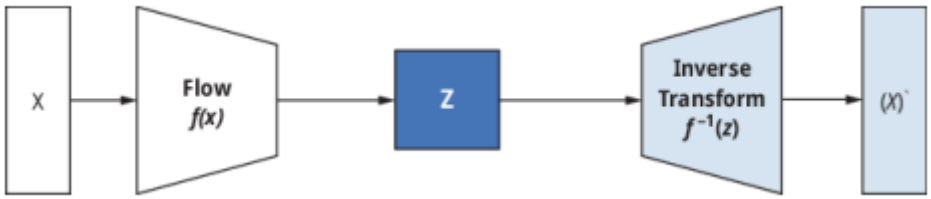


Zniekształcony obraz poddawany jest procesowi odszumiania w celu rekonstrukcji oryginalnych cech. Poprzez identyfikację kluczowych cech obrazów i wielokrotne cofanie zniekształceń zastosowanych do nich podczas pierwotnego procesu dyfuzji, model uczy się mapować zaszumione obrazy na punkty w tej przestrzeni ukrytej. Architektura UNet jest wykorzystywana do przeprowadzenia procesu odszumiania. Dekoder odbiera odszumiony wektor i konwertuje go z reprezentacji numerycznej na rzeczywisty obraz. Model jest trenowany iteracyjnie poprzez dostosowywanie jego parametrów w celu ulepszenia wyników predykcji z zaszumionych obrazów. Po pomyślnym zakończeniu wstępnego treningu model jest gotowy do generowania nowych obrazów. Na koniec model dyfuzji otrzymuje komunikat tekstowy, który jest szczegółowym opisem obrazu, który ma zostać utworzony na podstawie generowanych obrazów syntetycznych. Komunikat jest przekazywany do kodera tekstu w celu przekształcenia go w wektor numeryczny, który oddaje jego semantyczne znaczenie. Zakodowany wektor jest zniekształcany poprzez dodawanie szumu w procesie dyfuzji. Na koniec, proces odszumiania stopniowo usuwa szum, aby wygenerować wektor bez szumu, który jest dostarczany do dekodera w celu rekonstrukcji obrazu. Kluczowymi mechanizmami modelu dyfuzji są stochastyczne równania różniczkowe (SDE), modelowanie generatywne oparte na wynikach (SGM) oraz odszumiające modele probabilistyczne dyfuzji (DDPM). Model ten odgrywa kluczową rolę w procesie generowania danych. Model matematyczny SDE, szczegółowy proces dodawania szumu do oryginalnego obrazu, przebiega w sposób przyrostowy. Ta struktura ma kluczowe znaczenie, ponieważ umożliwia modelom dyfuzji interakcję z wieloma typami danych i aplikacji, umożliwiając ich dostosowanie do różnorodnych zadań generowania. SGM to procesy, w których model uczy się rozumieć i odszumiać obraz, co pomaga w generowaniu realistycznych wyników z zaszumionych danych. DDPM to modele dyfuzji, które probabilistycznie usuwają szum w celu rekonstrukcji oryginalnych danych. Podczas szkolenia uczą się, jak szum kumuluje się w danych w czasie i jak odwrócić ten proces, aby odzyskać dane oryginalne. Wiąże się to z wykorzystaniem rachunku prawdopodobieństwa do tworzenia uzasadnionych założeń dotyczących wyglądu danych przed wprowadzeniem szumu. Ta metoda ma kluczowe znaczenie dla

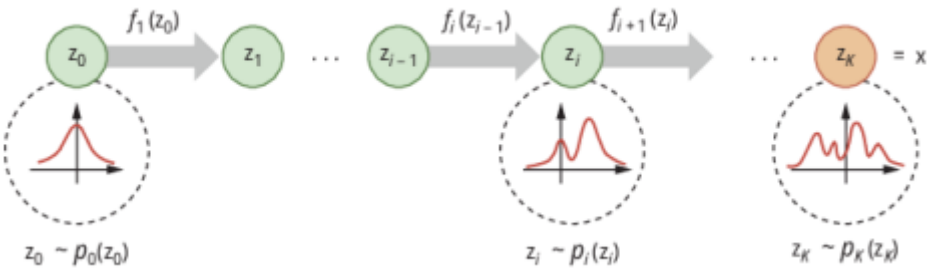
zdolności modelu do efektywnej rekonstrukcji danych, zapewniając, że wyniki są nie tylko wolne od szumu, ale także ściśle powiązane z danymi oryginalnymi.

Modele oparte na przepływie

Model oparty na przepływie to generatywny model oparty na sztucznej inteligencji, który wykorzystuje sekwencję transformacji odwrotnych do generowania obrazu. Model generatywny oparty na przepływie składa się z serii funkcji transformacji odwrotnej, które bezpośrednio uczą się rozkładu danych , jak pokazano na rysunku 11.



GAN i VAE działają wyjątkowo dobrze w przypadku złożonych rozkładów danych; jednak oba modele nie umożliwiają dokładnej oceny i wnioskowania na temat rozkładu prawdopodobieństwa. Oba te problemy zostały rozwiązane poprzez normalizację przepływów poprzez funkcje transformacji odwrotnej. Przepływy normalizujące modelują rzeczywisty rozkład danych i dostarczają nam precyzyjnego prawdopodobieństwa danych; stąd modele oparte na przepływach wykorzystują ujemny logarytm wiarygodności jako funkcję straty. Przekształca ona rozkład prosty w rozkład zespolony poprzez zastosowanie sekwencji transformacji odwrotnych. Najczęściej stosowanymi typami przepływów normalizujących są transformacja logarytmiczna (fθ(z) = log(z)), transformacja afiniczna (fθ(z) = Az + b) i transformacja wykładnicza (fθ(z) = ez). Funkcje transformacji odwrotnej są używane przez generatywne modele oparte na przepływie do tłumaczenia danych wejściowych x na ukryte reprezentacje Z, co zilustrowano na Rysunku 12.



W tym przypadku z musi pasować kształtem do x. W tym przypadku funkcje odwracalne oznaczają, że mamy pasującą reprezentację ukrytą (z) dla każdego punktu danych (x), umożliwiając bezstratną rekonstrukcję (z do x). Podczas treningu modele oparte na przepływie mapują obrazy wejściowe do przestrzeni utajonej z, a następnie tworzą obrazy poprzez próbkowanie przestrzeni utajonej z i zastosowanie jej do odwrotnych funkcji transformacji f. Obraz wejściowy jest rzutowany na przestrzeń utajoną, przestrzeń wielowymiarową reprezentującą podstawowe cechy. Używając sekwencji odwracalnych transformacji, przestrzeń utajona jest konwertowana na przestrzeń danych przy użyciu wybranej funkcji transformacji. Funkcja straty jest obliczana w celu trenowania modelu przy użyciu typowej techniki uczenia maszynowego. Po wytrenowaniu modelu generatywnego można go użyć do generowania nowych danych.

Metryki oceny

Ocena ma kluczowe znaczenie dla zrozumienia możliwości i ograniczeń modeli. Osiąga się ją poprzez wykorzystanie odpowiednich metryk do porównania jakości wygenerowanego materiału z rzeczywistą treścią. Dostosowanie lub dostrojenie hiperparametrów modelu szkoleniowego poprawia jego wydajność. W zależności od kontekstu aplikacji, zaproponowano różne zestawy metryk do oceny, takie jak precyzja, przypominanie, dokładność, metoda ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation), dwujęzyczny student oceny (BLEU), konsternacja, wynik BERT, METEOR (Metric for Evaluation of Translation with Explicit Ordering), MoverScore i NIST.

Metryki ewaluacji

Ewaluacja ma kluczowe znaczenie dla zrozumienia możliwości i ograniczeń modeli. Osiąga się ją poprzez wykorzystanie odpowiednich metryk do porównania jakości wygenerowanego materiału z rzeczywistą treścią. Dostosowanie lub dostrojenie hiperparametrów modelu szkoleniowego poprawia jego wydajność. W zależności od kontekstu aplikacji, zaproponowano różne zestawy metryk do oceny, takie jak precyzja, przypominanie, dokładność, Recall-Oriented Understudy for Gisting Evaluation (ROUGE)-L, dwujęzyczny student ewaluacji (BLEU), konsternacja, wynik BERT, METEOR (Metric for Evaluation of Translation with Explicit Ordering), MoverScore i NIST [→25, →26]. W ocenie generatywnych modeli LLM sztucznej inteligencji, pomiary statystyczne, takie jak przypominanie, precyzja i wynik F1, są obliczane za pomocą tabeli kontyngencji w tabeli , która jest przydatna do określenia zdolności modelu do generowania poprawnych i kontekstowo odpowiednich odpowiedzi.

	Relevant	**Irrelevant**
Retrieved	*r*	*n*
Unretrieved	*R*	*N*

Te wskaźniki techniczne podkreślają zdolność modelu do interpretowania danych przychodzących i dostarczania wartościowych wyników z wysokim poziomem dokładności. Dokładność to miara liczby treści poprawnie sklasyfikowanych przez model:

Dokładność = r+N/(r+n+R+N)

Miara precyzji skupia się na pozytywnych wynikach modelu, czyli liczbie pobranych istotnych treści w stosunku do całkowitej liczby istotnych treści:

Precyzja = r/(r+R)

Wskaźnik zapamiętania to miara istotnych informacji odzyskanych w wyniku procesu wyszukiwania poprzez unikanie informacji nieistotnych:

Przypomnij sobie = r/(r+n)

Podobnie, wynik F1 ma na celu znalezienie równowagi między precyzją a odwołaniem, co jest idealnym miernikiem w przypadku niezrównoważonego zbioru danych:

Wynik F1 = 2×Precyzja×Odwołanie / (Precyzja + Odwołanie)

Wynik Początkowy (IS)

Wynik Początkowy (IS) to statystyka służąca do oceny jakości obrazów generowanych przez sieć GAN lub inne modele generatywne. Ma on na celu uchwycenie ostrości, czyli zbliżenia do rzeczywistości, oraz różnorodności generowanych obrazów [→25, →26]. Jest to miara entropii warunkowej obliczana na podstawie wygenerowanych obrazów wzdłuż odchylenia standardowego:

IS(G) = exp(\euro$_{x\sim pg}$D$_{KL}$(p(y x) p(y))),

gdzie € to średnia obliczona dla wygenerowanego obrazu, a D_{KL} (p(y|x)) to dywergencja Kullbacka–Leiblera między warunkowym rozkładem prawdopodobieństwa przewidywanych etykiet klas. Wyższy IS wskazuje na generowanie obrazów lepszej jakości.

Odległość rozpoczęcia Frecheta

Odległość rozpoczęcia Frecheta (FID) to miara statystyczna oparta na IS, służąca do oceny jakości wygenerowanego obrazu z uwzględnieniem rzeczywistego rozkładu danych [→25, →26]. Niech G i T reprezentują próbki wygenerowane i testowe; podobnie FG i FT odnoszą się do ich reprezentacji cech. Niech (μ_G, ΣG) i (μ_T, Σ_T) oznaczają średnią i kowariancję dwóch rozkładów, gdzie FID jest miarą odległości Wassersteina-2 między tymi dwoma rozkładami:

$$\mathrm{FID} = ||\mu_T - \mu_G||^2 + \mathrm{Tr}(\Sigma_T + \Sigma_G - 2\left(\Sigma_T \Sigma_G)^{\frac{1}{2}}\right)$$

CLIP

CLIP to statystyczna miara relatywizmu wygenerowana w odniesieniu do podanego opisu tekstowego. Jest to miara współczynnika podobieństwa cosinusowego między dwoma wektorami osadzania: vi reprezentuje wektor osadzania obrazu, a vt odnosi się do wektora osadzania tekstu w tej samej przestrzeni ukrytej:

Podobieństwo cosinusowe (vi, vt) = (vi. vt)/|‖vi‖ ‖vt‖|

Perpleksja

Perpleksja to miara prawdopodobieństwa, najlepiej stosowana w zadaniach przetwarzania języka naturalnego (NLP), w celu oceny skuteczności przewidywania próbki przez model prawdopodobieństwa na podstawie jej rozkładu [→25–27]. W przypadku odgadnięcia kolejnego słowa w sekwencji, niska perpleksja wskazuje na trafność wyników przewidywania:

Prepleksja = $(1/p(x_1, x_2,\ldots, x_n))^1/n$

gdzie $p(x_1, x_2,\ldots, x_n)$ odnosi się do prawdopodobieństwa dla całej sekwencji słów przypisanej przez model i odnosi się do całkowitej liczby słów w sekwencji.

Wynik BLEU

BLEU to precyzyjna miara, która oblicza liczbę n-gramowych nakładek między tekstem referencyjnym a tekstem utworzonym. Wynik uwzględnia również karę za zwięzłość, która jest naliczana, gdy tekst wygenerowany maszynowo jest znacznie krótszy niż tekst referencyjny [→28]. Wyższy wynik BLEU wskazuje na większe podobieństwo między dwoma tekstami:

$$\mathrm{BLEU} = \min(1, \mathrm{output} - \mathrm{length}/\mathrm{reference} - \mathrm{length})\left(\prod_{i=1}^{4} \mathrm{precision}_i\right)^{\frac{1}{4}}$$

ROUGE

ROUGE to miara nakładania się tekstu referencyjnego i wygenerowanego. ROUGE jest mierzony w trzech wariantach, takich jak ROUGE-n, ROUGE-1 i ROUGE-s [→29]:

ROUGE-n – wykrywanie liczby nakładających się fragmentów

ROUGE-1 – wykrywanie najdłuższego wspólnego podciągu w dwóch tekstach

ROUGE-s – koncentracja na gramach pomijających

ROUGE = Liczba n−gramów znalezionych w odniesieniu i modelu / Liczba n−gramów znalezionych w odniesieniu

METEOR

METEOR to metryka skupiająca się na jakości generowanego tekstu w oparciu o dopasowanie w tekście referencyjnym i modelowym [→28]. Miarę oblicza się za pomocą średniej harmonicznej precyzji i odwołania do unigramu, z większym naciskiem na odwołanie niż na precyzję:

F_{mean} = 10PR/R+9P

gdzie p odnosi się do precyzji unigramu, czyli stosunku liczby zmapowanych unigramów do całkowitej liczby unigramów w tłumaczeniu systemowym, a R odnosi się do odwołania, czyli stosunku liczby zmapowanych unigramów do całkowitej liczby unigramów w odniesieniu. Aby uwzględnić dłuższe dopasowanie, obliczana jest kara:

$$\text{Penalty} = 0.5 \times \left(\frac{\#\text{chuncks}}{\#\text{unigrams_matched}}\right)^3$$

Ostatecznie wynik METEOR zostanie podany jako

Wynik = F_{mean} × (1 − Kara)

BERT

BLEU i ROUGE to indeksy oparte na podobieństwie, których celem jest precyzyjne dopasowanie, idealne dla modeli generowania tekstu. Jednak w przypadku generatywnej sztucznej inteligencji, gdy wynik jest podobny do korpusu, a jednocześnie pozostaje znaczący, metryki te okazały się nieskuteczne . BERT koncentruje się na podobieństwie w oparciu o osadzenie kontekstowe, które jest idealne w przypadku generatywnych modeli opartych na tekście:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{x_j \in x} x_i^T x_j r^2$$

$$P_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{x_j \in x} x_i^T x_j r^2$$

$$F_{\text{BERT}} = 2 \times \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}$$

Wynik GPT

Stosunkowo nową statystyką służącą do oceny modeli LLM jest wynik GPT. Wynik GPT ocenia jakość tekstu, wykorzystując wrodzone umiejętności LLM [→30]. Wykorzystuje on sam LLM do oceny wygenerowanego tekstu zgodnie z ustalonymi wcześniej wytycznymi lub wymaganiami wejściowymi:

$$\text{GPT score}(h\ d, a, S) = \sum_{t=1}^{m} w_t \log p\big(h_t|\ h_{<t}, T(d, a, s), 0\big)$$

gdzie w_t to waga tokena w danej pozycji, h to tekst do oceny, d to opis zadania, a to definicja aspektu, a S to informacje kontekstowe.

Współczynnik podobieństwa Levenshteina

Współczynnik podobieństwa Levenshteina (LSR) to ponownie wskaźnik oparty na podobieństwie, mierzący minimalną liczbę edycji wymaganych do przekształcenia jednego tekstu w inny:

LSR = 1 − odległość Levenshteina (LD) / długość (najdłuższy ciąg)

gdzie LD odnosi się do minimalnej liczby edycji wymaganych do

przekształcenia tekstu.

MoverScore

MoverScore to metryka oparta na jakości, obliczana na podstawie semantyki utworzonego tekstu. MoverScore ocenia stopień podobieństwa semantycznego między wygenerowanym tekstem a jednym lub kilkoma tekstami referencyjnymi – a nie tylko powierzchowne nakładanie się słów. MoverScore opiera się na dwóch głównych komponentach, a mianowicie osadzeniu kontekstowym i odległości Earth Mover [→31]:

$$\mathrm{EMD}(P, Q) = \min_{\gamma} \Sigma_i \Sigma_j \gamma(i,\ j)\ *\ d(p_i, q_j)$$

gdzie P i Q odnoszą się do rozkładów prawdopodobieństwa, y(i,j) odnosi się do ilości masy, która ma zostać przeniesiona z elementu i w P do elementu j w Q, a d(pi, qi) jest miarą odległości między nimi. Końcowy wynik MoverScore wygenerowanego tekstu jest obliczany poprzez agregację wszystkich odległości Earth Mover w ciągu odniesienia. Dostarczono szereg miar oceny, które służą do gruntownego przetestowania wydajności modeli LLM i generatywnej sztucznej inteligencji, oceniając skuteczność i zdolność przetwarzania języka w udzielaniu odpowiedzi na pytania użytkowników [→32].

Wnioski

Zagłębiliśmy się w tętniący życiem świat generatywnej sztucznej inteligencji oraz proces szkolenia i oceny modeli LLM. Przeanalizowaliśmy różne metodologie szkolenia, które zostały wykorzystane do wdrożenia tych modeli. Techniki generatywnej sztucznej inteligencji są szeroko stosowane w wielu zastosowaniach, takich jak synteza obrazu, translacja obrazu i ulepszanie obrazu, ze względu na ich zdolność do dostarczania bardziej spójnych i kontekstowo istotnych treści. Generatywne modele sztucznej inteligencji, takie jak CycleGAN, warunkowe sieci GAN i Pix2PixGAN, rewolucjonizują dziedzinę widzenia komputerowego. CycleGAN okazał się skuteczniejszy w tłumaczeniu obrazów bez parowania bez potrzeby danych treningowych. Podobnie, warunkowe sieci GAN generowały obrazy w oparciu o precyzyjne warunki wejściowe, a Pix2PixGAN generuje obrazy o wysokiej wierności, odkrywając bezpośrednią korelację między obrazami wejściowymi i wyjściowymi. Model oparty na transformacjach rewolucjonizuje dziedzinę przetwarzania języka naturalnego (NLP) pod względem rozumienia immanentnych cech języka, wychwytywania zależności dalekiego zasięgu i generowania bardziej spójnej treści za pomocą mechanizmu samouwagi. Dzięki wstępnemu szkoleniu na rozległych zbiorach danych tekstowych, a następnie dostrajaniu do konkretnych zadań, LLM-y wykazały się wyjątkowymi zdolnościami w wielu obszarach przetwarzania języka naturalnego (NLP), takich jak tłumaczenie języka, streszczanie tekstu, odpowiadanie na pytania i analiza sentymentu. Na koniec przyjrzeliśmy się różnym wskaźnikom ilościowym i jakościowym, takim jak precyzja, IS, MoverScore, BLEU, CLIP i perplexity, które weryfikują wydajność modelu i przygotowują grunt pod udoskonalenia.