Chat GPT

Przegląd istniejących rozwiązań

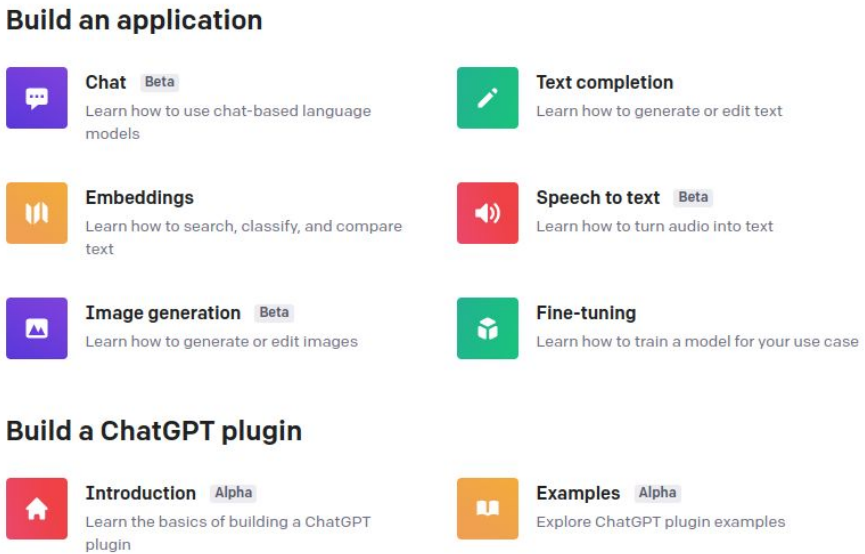
Bartosz Mikulski

OpenAI Platform

Nie tylko Chat GPT

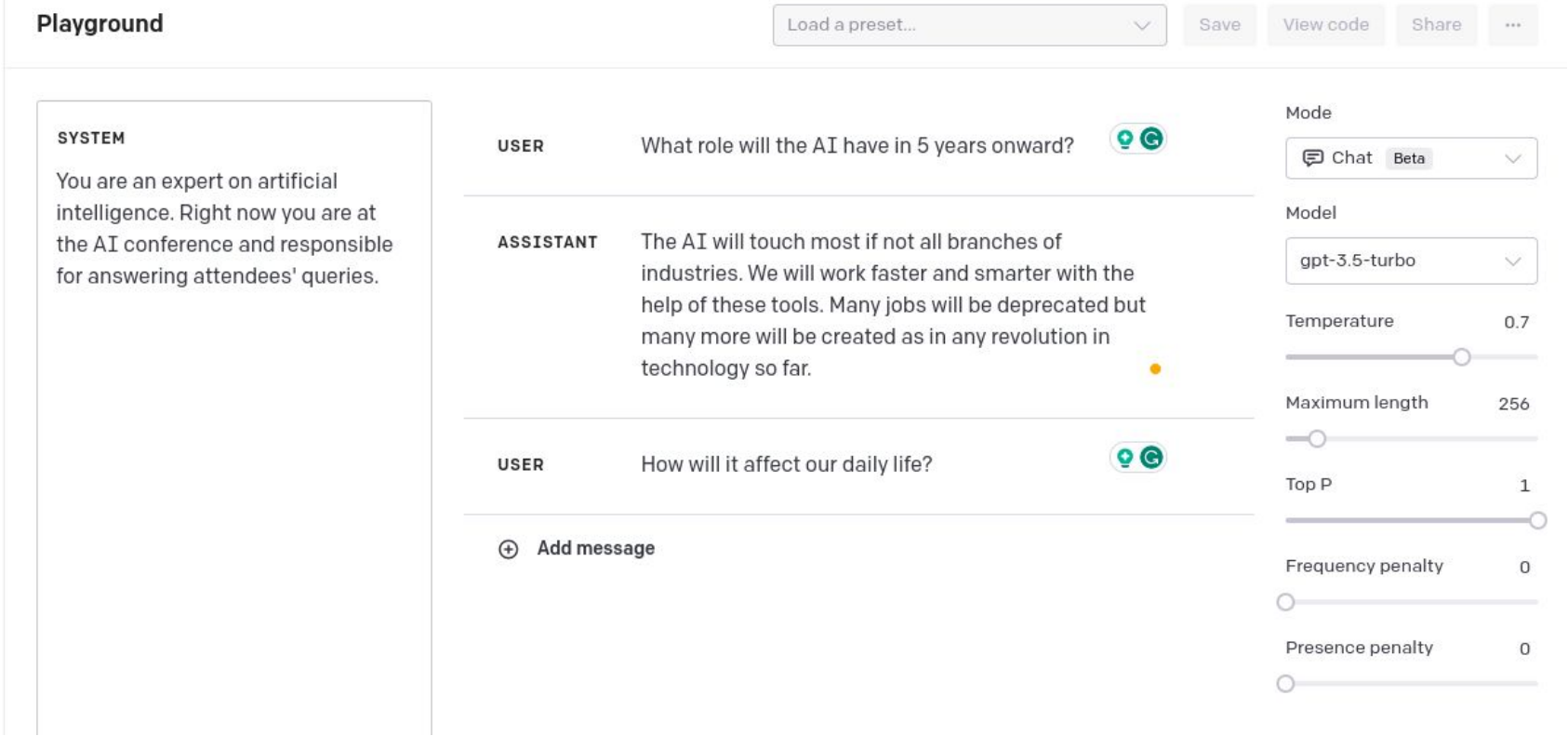
ChatGPT to tylko jedno z narzędzi dostępne od **OpenAI**. Większość ludzi używa go z poziomu przeglądarki internetowej w celu zadawania pytań (Promptów) do modelu.

Jednakże **OpenAI** udostępnia dużo innych produktów, które można wykorzystać do budowania aplikacji opartych o AI.



https://platform.openai.com/

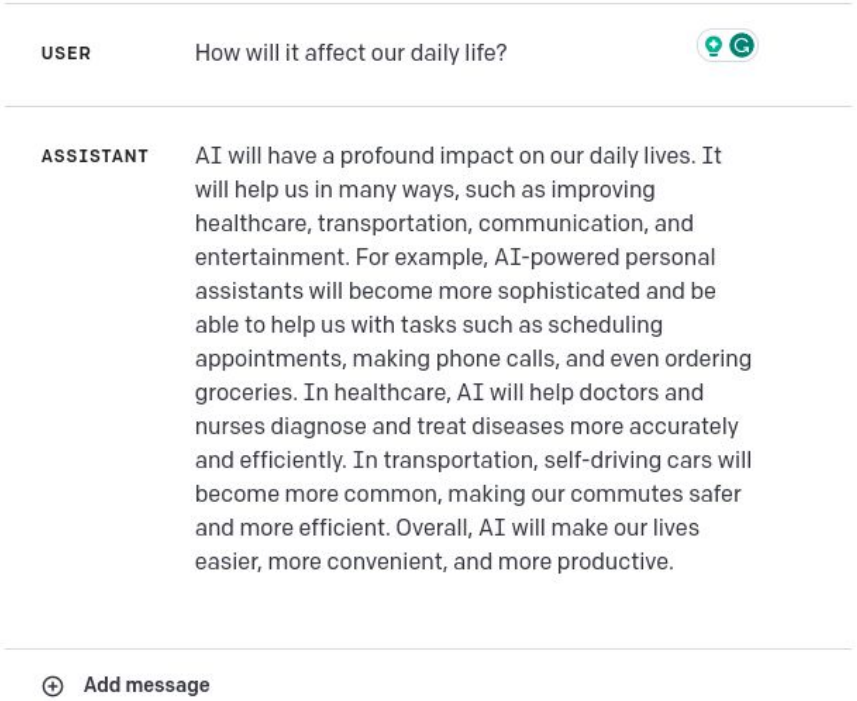
Chat API


Playground
Load a preset...
Save
View code
Share
SYSTEM
You are an expert on artificial intelligence. Right now you are at the AI conference and responsible for answering attendees' queries.
USER
What role will the AI have in 5 years onward?
ASSISTANT
The AI will touch most if not all branches of industries. We will work faster and smarter with the help of these tools. Many jobs will be deprecated but many more will be created as in any revolution in technology so far.
USER
How will it affect our daily life?
Add message
Mode
Chat
Beta
Model
gpt-3.5-turbo
Temperature
0.7
Maximum length
256
Top P
1
Frequency penalty
0
Presence penalty
0

Chat API

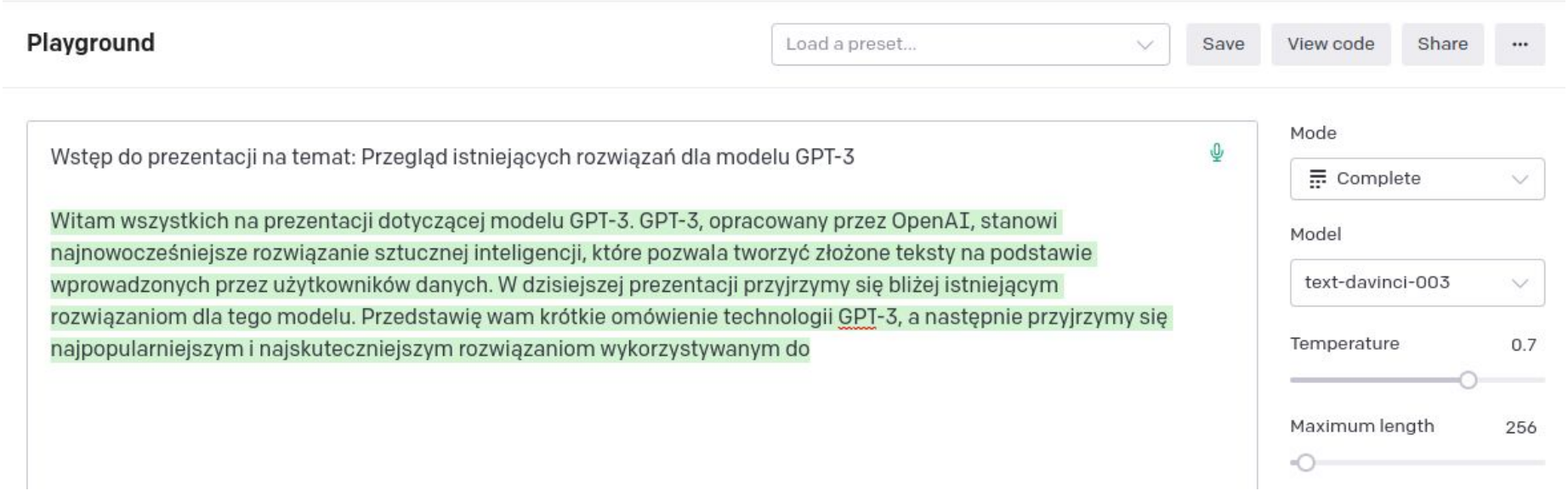
Standardowo jak w aplikacji ChatGPT tutaj model też odpowiada na nasze zapytania. Jednakże możemy zamodelować początek rozmowy, wraz z opisem asystenta.

Umożliwia to tworzenie aplikacji która będzie mogła odpowiadać za nas, biorąc pod uwagę dotychczasową rozmowę, np. między klientem, a konsultantem.



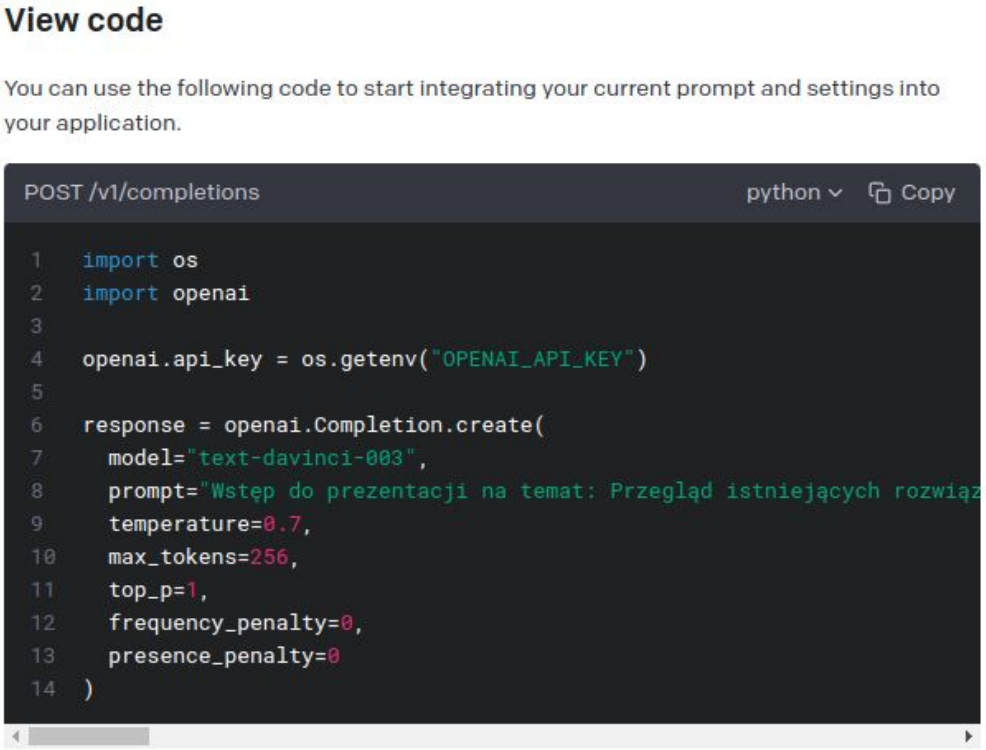
Complete API

Dedykowany model do uzupełniania podanego tekstu.



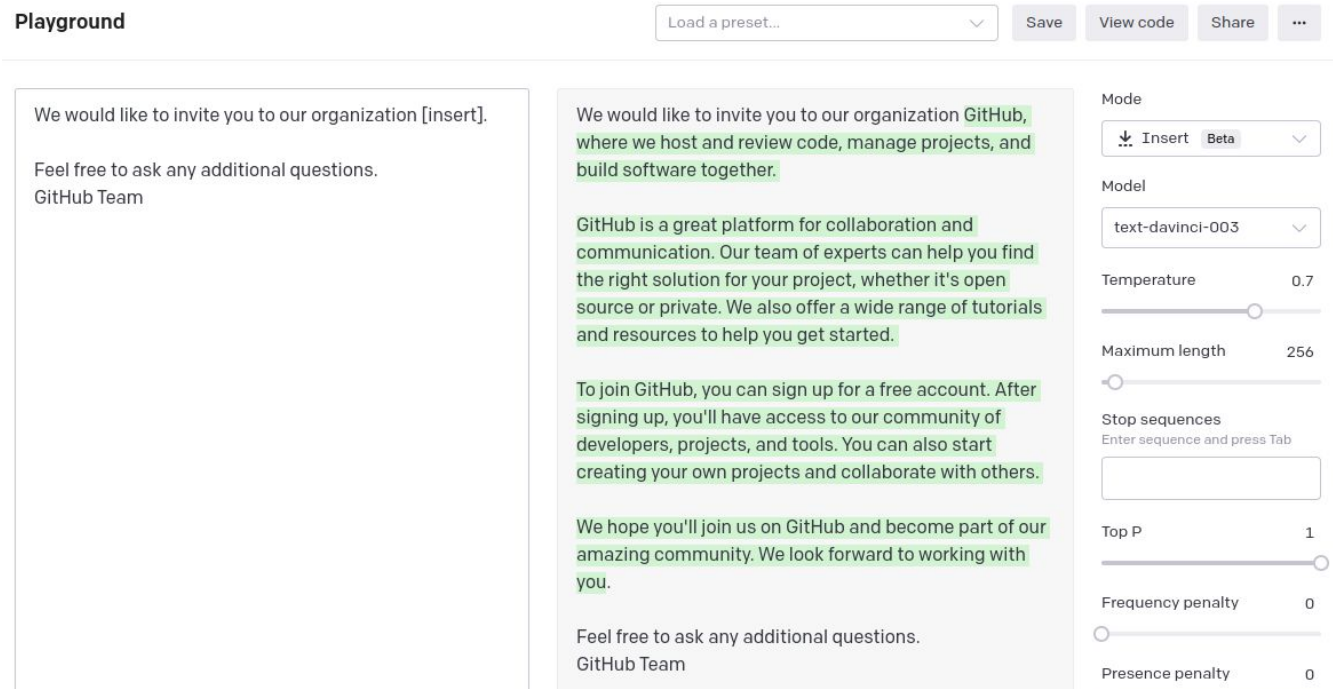
Complete API

Wszystkie zapytania które wykonaliśmy w piaskownicy, można przekonwertować do kodu w Pythonie, do bezpośredniego użycia w naszej aplikacji.



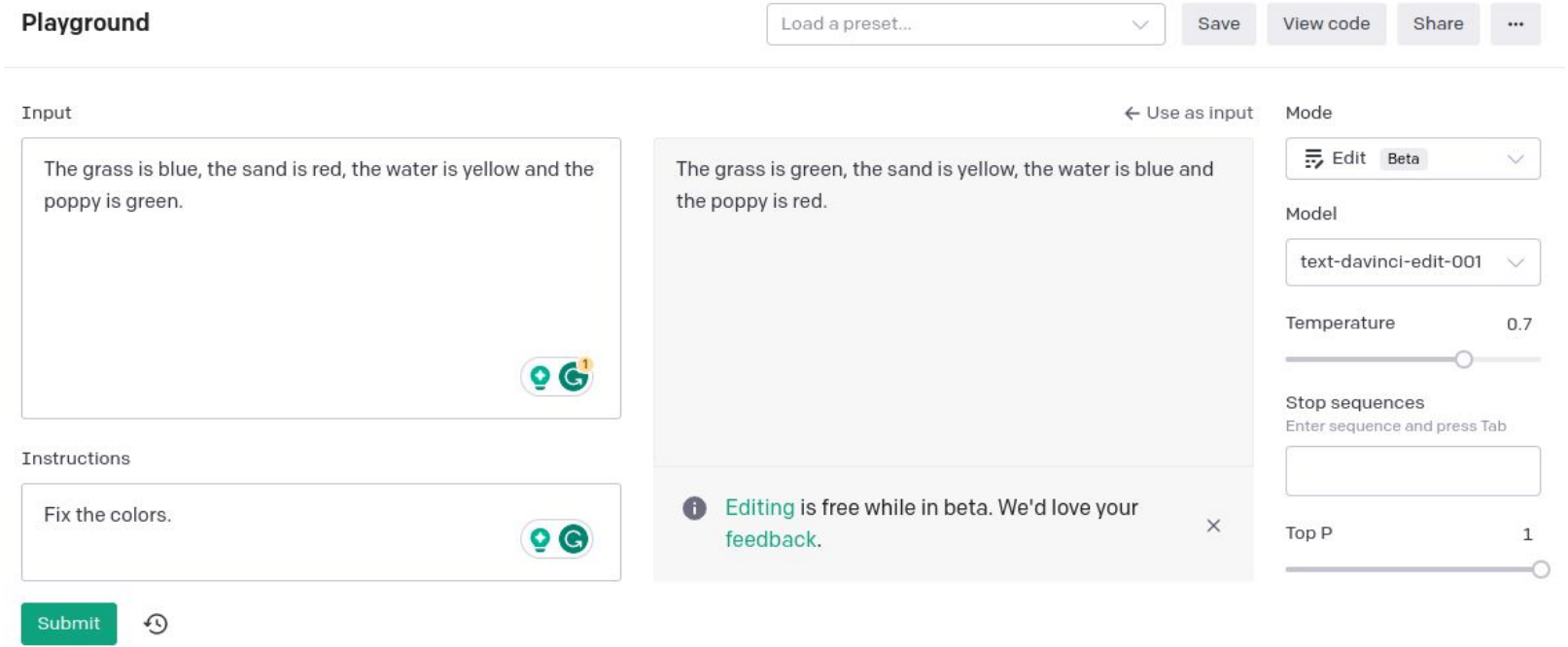
Insert API

Dedykowany model do generowania tekstu w wybranym miejscu.



Edit API

Dedykowany model do wykonywania edycji na danym wejściu.



Zastosowania

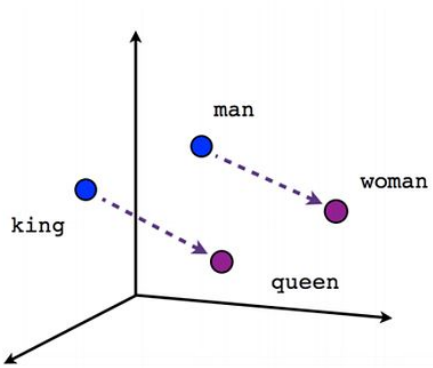
Przedstawione API (jak i cała rodzina GPT) nadaje się do poniższych zastosowań, które mogą mieć zastosowanie w wielu aplikacjach komercyjnych.

- Generowanie podsumowania długiego tekstu
- Poprawa błędów gramatycznych
- Wyjaśnianie skomplikowanych pojęć
- Generowanie szkieletu tekstu
- Generowanie listy pomysłów
- Debugowanie kodu aplikacji
- Szybkie znajdowanie prostych informacji

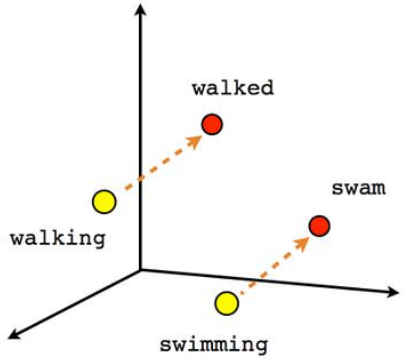
Embeddings API

Embedding to wektor obliczony na podstawie wejściowego tekstu. Takie wektory mają konkretne właściwości:

- podobne pojęcia znajdują się blisko siebie w przestrzeni
- powiązane pojęcia mają podobną odległość i kierunek do innych powiązanych pojęć
- można zmierzyć podobieństwo dwóch wektorów (tekstów) używając zwykłej odległości kosinusowej



Male-Female



Verb tense

https://arxiv.org/abs/1301.3781

Zastosowania

- Wyszukiwanie - rezultaty oceniane przez dopasowanie do zapytania
- Grupowanie - test jest grupowany podobieństwem
- Rekomendacje - przedmioty z podobnymi opisami są polecane
- Wykrywanie anomalii - wykrywanie tekstów które są mało powiązane z resztą korpusu
- Miary różnorodności - mierzenie rozkładów miar podobieństwa w korpusie
- Klasyfikacja - tekst jest klasyfikowany na podstawie największego podobieństwa do etykiety

https://www.pinecone.io/

Z tego powodu bardzo popularne stały się wektorowe bazy danych takie jak Pinecone, które są zoptymalizowane pod przechowywanie obliczonych wektorów i uproszczenia wykonywanie operacji opisanych po lewej stronie.

https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

Obecne zastosowania

LangChain

LingChain to framework do budowania aplikacji opartych o LLM (w tym platformę Open.AI, ale nie tylko). Upraszcza i strukturyzuje pracę z modelami za pomocą uniwersalnych modułów, takich jak:

- LLM i propmty
- Łańcuchy
- Augmentacje danych
- Agenci
- Pamięć

Przykłady rozwiązań które można zbudować z wykorzystaniem LangChain:

- Personalny asystent
- Odpowiadanie na pytania z użyciem zewnętrznych danych
- Odpytywanie API
- Podsumowywanie długich dokumentów

https://docs.langchain.com/docs/use-cases/qa-docs | https://github.com/hwchase17/langchain

LangChain - Odpowiadanie na pytania

Bardzo często pojawia się problem automatycznego odpowiadania na pytania na które odpowiedzi znajdują się się w istniejących zasobach.

Może to być Wikipedia, albo dokumentacja projektu. Znalezienie odpowiedzi czasami zajmuje dużo czasu. Niestety modele LLM nie są zwykle uczone na danych wewnątrz firmowych (no chyba, że są hostowane na GitHub ;))

LangChain pozwala nam zbudować aplikację, która będzie w stanie odpowiadać na pytania naszego zespołu i klientów bez trenowania modelu.

Przygotowanie danych

Dla każdego dokumentu liczymy embeddingi słów i zapisujemy dokumenty i embeddingi w wektorowej bazie danych.

Potok danych

Dla przychodzącego zapytania liczymy embedding i szukamy najbardziej podobnych dokumentów w bazie (np.: top 5)

Umieszczamy zawartość tych dokumentów w prompcie razem z pytaniem i odpytujemy LLM.

LangChain - Podsumowanie długiego tekstu

LLM są świetne do generowania podsumowania tekstu. Jednakże mają ograniczenia, np.: GPT 3.5 ma maksymalne okno percepcyjne o długości 4096 tokenów (token =/= słowo).

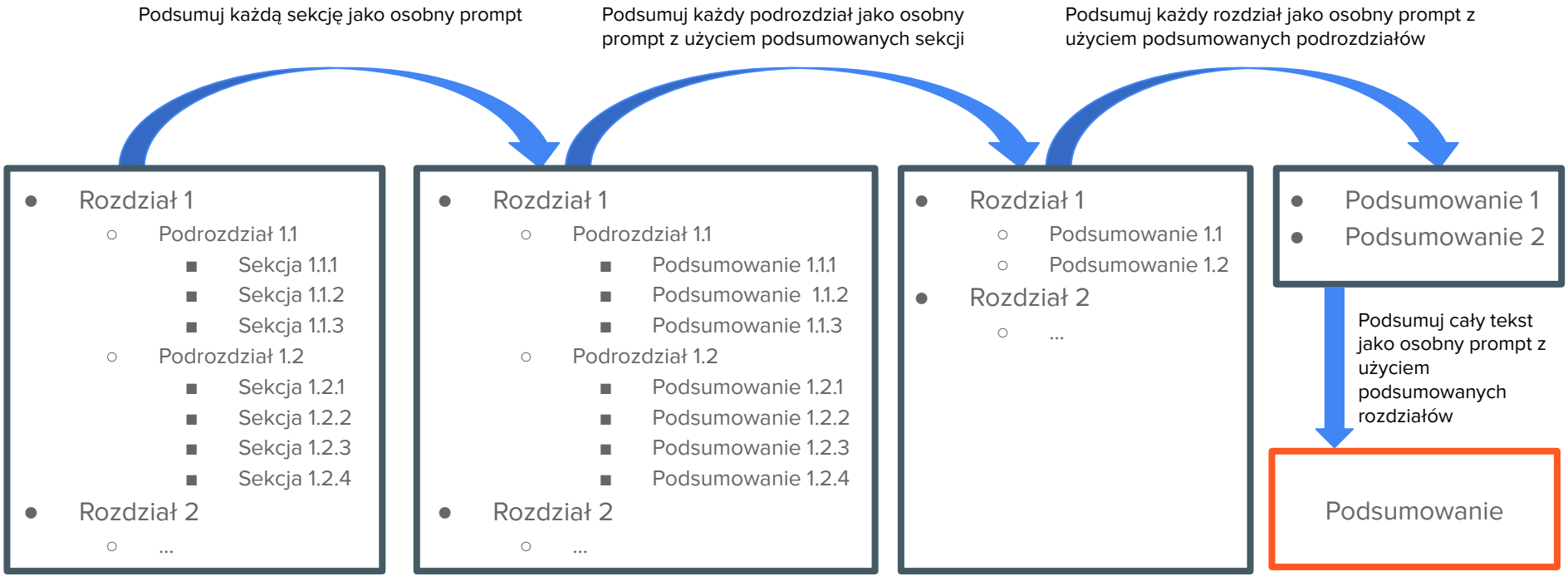
Co zrobić kiedy mamy bardzo długi tekst (np.: 10 stron A4), który chcemy podsumować?

Zwykły model LLM podsumuje tylko ostatnie 4096 tokenów co nie jest poprawną odpowiedzią.

Rozwiązaniem jest podzielenie tekstu na kawałki. Dobrą abstrakcją jest wizualizacja podziału jak podział na rozdziały, podrozdziały, sekcje itp.

Model odpytujemy rekurencyjnie od wyjaśnienia najmniejszych bloków, aż do wyjaśnienia całego tekstu na podstawie pomniejszych podsumowań.

LangChain - Podsumowanie długiego tekstu



LangChain - przykład kodu dla API

```
!pip install arxiv
```

```
from langchain.chat_models import ChatOpenAI
from langchain.agents import load_tools, initialize_agent, AgentType

llm = ChatOpenAI(temperature=0.0)
tools = load_tools(
    ["arxiv"],
)

agent_chain = initialize_agent(
    tools,
    llm,
    agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
    verbose=True,
)
```

```
agent_chain.run(
    "What's the paper 1605.08386 about?",
)
```

```
> Entering new AgentExecutor chain...
I need to use Arxiv to search for the paper.
Action: Arxiv
Action Input: "1605.08386"
Observation: Published: 2016-05-26
Title: Heat-bath random walks with Markov bases
Authors: Caprice Stanley, Tobias Windisch
Summary: Graphs on lattice points are studied whose edges come from a finite set of
allowed moves of arbitrary length. We show that the diameter of these graphs on
fibers of a fixed integer matrix can be bounded from above by a constant. We
then study the mixing behaviour of heat-bath random walks on these graphs. We
also state explicit conditions on the set of moves so that the heat-bath random
walk, a generalization of the Glauber dynamics, is an expander in fixed
dimension.
Thought:The paper is about heat-bath random walks with Markov bases on graphs of lattice points.
Final Answer: The paper 1605.08386 is about heat-bath random walks with Markov bases on graphs of
lattice points.

> Finished chain.
```

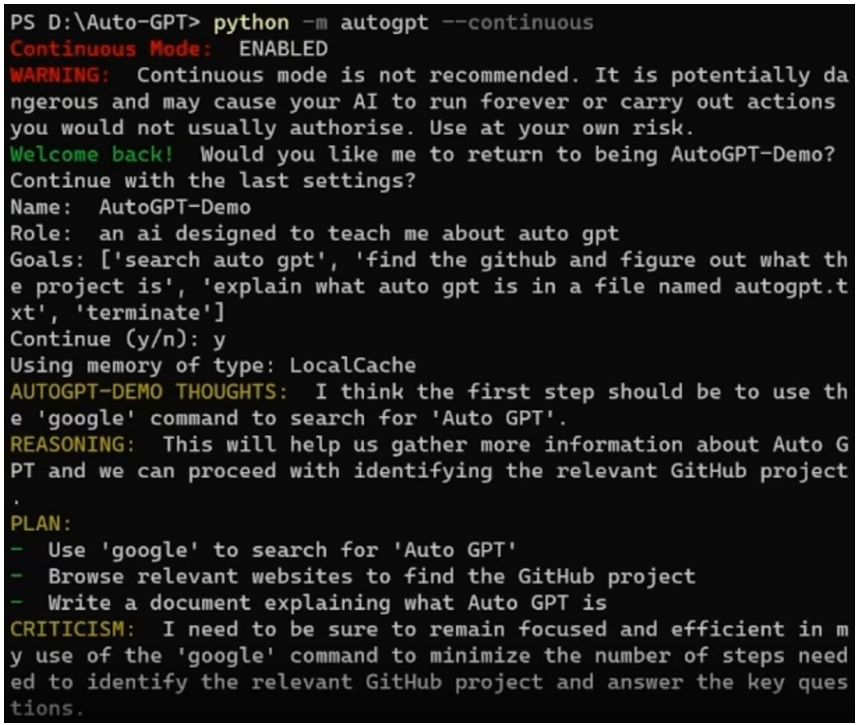
```
'The paper 1605.08386 is about heat-bath random walks with Markov bases on graphs of lattice
points.'
```

https://python.langchain.com/en/latest/modules/agents/tools/examples/arxiv.html

Auto GPT

Autonomiczny GPT - eksperyment i technologiczne demo pokazujące jak LLM mogą rozwiązywać skomplikowane problemy bez pomocy z zewnątrz. Aplikacja posiada dostęp do internetu i różnych narzędzi, które może wykorzystać aby osiągnąć dany cel.

Auto GPT rozbija skomplikowany cel na szereg mniejszych, które jest w stanie wykonać z użyciem dostępnych mu narzędzi.



https://github.com/Significant-Gravitas/Auto-GPT

Stripe

Lepsze zrozumienie biznesu klienta

W celu wsparcia klientów Stripe przegląda ich strony internetowe i robi podsumowanie ich biznesu.

Okazuje się, że GPT jest w stanie generować lepsze podsumowanie biznesu klienta, niż ludzie.

Odpowiadanie na problemy

Zespół do spraw pomocy, często musiał odpowiadać na oczywiste pytania, na które odpowiedzi były zawarte w dokumentacji. Znalezienie lub nakierowanie klienta na odpowiedź trwało bardzo długo.

GPT jest w stanie automatycznie odpowiadać na takie pytania po podłączeniu dokumentacji do modelu.

https://openai.com/customer-stories/stripe

Be My Eyes

Od 2012 roku Be My Eyes tworzy technologię dla społeczności ponad 250 milionów osób niewidomych lub niedowidzących. Duński startup łączy osoby niewidome lub niedowidzące z wolontariuszami, którzy pomagają im w setkach codziennych zadań, takich jak identyfikacja produktu lub nawigacja po lotnisku.

Dzięki nowym możliwościom wprowadzania danych wizualnych GPT-4 (w wersji zapoznawczej), Be My Eyes rozpoczęło opracowywanie Virtual Volunteer™ opartego na GPT-4 w aplikacji Be My Eyes, która może generować ten sam poziom kontekstu i zrozumienia, co ochotnik-człowiek.

https://openai.com/customer-stories/be-my-eyes

Koniec

Dziękuje za uwagę

Kontakt

- LinkedIn: https://www.linkedin.com/in/bartosz-mikulski/
- GitHub: https://github.com/BartMiki
- Kontakt: bartosz.mikulski.praca@gmail.com