Dokumentacja projektu Llama Chatbot

Łukasz Czyż

Daniel Gut

Miłosz Kilman

Michał Serkuczewski

Piotr Marczak

Wprowadzenie

Rozwój nowoczesnych technologii i narzędzi przetwarzania języka naturalnego pozwala na tworzenie zaawansowanych aplikacji, takich jak chatboty. Naszym celem było stworzenie chatbota z wykorzystaniem modeli LLama, który będzie w stanie odpowiadać na pytania dotyczące materiałów edukacyjnych z wykładów na naszej uczelni. W tym projekcie, chatbot wykorzystuje lokalnie uruchomiony model LLama-2-13B-Chat do przetwarzania danych w formacie PDF.

Cel projektu

Celem projektu było stworzenie aplikacji chatbota, który umożliwi na łatwy dostęp do wiedzy zawartej w materiałach edukacyjnych z wykładów. Potrzeba takiej aplikacji wynika z chęci poprawy efektywności nauki a przede wszystkim szybkiego i łatwego dostępu do informacji. Szczególnie przydatne może okazać się podczas przygotowań do egzaminów. Obecnie dostępne rozwiązania mogą nie spełniać w pełni potrzeb studentów, dlatego opracowanie dedykowanego narzędzia z wykorzystaniem zaawansowanych modeli językowych, takich jak LLama, było naturalnym krokiem.

Wybrana technologia

- **FastAPI**:

Wybraliśmy FastAPI jako framework do budowy backendu aplikacji ze względu na jego wysoką wydajność, nowoczesną architekturę oraz łatwość integracji z systemami bazującymi na modelach ML. FastAPI pozwala na szybkie tworzenie API opartego na standardach OpenAPI i JSON Schema, co ułatwia budowanie skalowalnych aplikacji.

- **LLama-2-13B-Chat**:

Wybraliśmy model LLama-2-13B-Chat, udostępniony przez TheBloke, z uwagi na jego zdolność do generowania wysokiej jakości odpowiedzi na pytania. Model ten jest odpowiednio zoptymalizowany do uruchamiania lokalnie, na sprzęcie z ograniczonymi zasobami. Modele Llama potrzebuję wysokiej mocy obliczeniowej. Wybrany model jest możliwy do uruchomienia na sprzęcie, gdzie dostępne jest 16 GB RAM.

- **Baza wektorowa Chroma**:

Służy do efektywnego przeszukiwania dokumentów i zarządzania informacjami w kontekście dużych zbiorów danych tekstowych. Jest to istotne przy ekstrakcji odpowiednich fragmentów z materiałów PDF przed przekazaniem ich do modelu LLama.

- **Streamlit**:

Streamlit został wybrany jako narzędzie do budowy interfejsu użytkownika (UI) aplikacji ze względu na jego prostotę i szybkość wdrożenia. Streamlit umożliwia tworzenie interaktywnych i dynamicznych aplikacji webowych w Pythonie bez konieczności posiadania zaawansowanej wiedzy z zakresu front-end developmentu. Dzięki Streamlit można łatwo integrować komponenty wizualne z modelem LLama oraz bazą wektorową, co umożliwia użytkownikom intuicyjne przeszukiwanie dokumentów i uzyskiwanie odpowiedzi w czasie rzeczywistym.

Metoda

- Architektura:

Nasza aplikacja opiera się na architekturze klient-serwer. Backend oparty na FastAPI zajmuje się przetwarzaniem zapytań, integracją z modelem LLama oraz zarządzaniem bazą wektorową.

- Przetwarzanie PDF:

Dokumenty PDF są wstępnie przetwarzane, aby wyodrębnić tekst i metadane, które następnie są indeksowane w bazie wektorowej.

- Model LLama-2-13B-Chat:

Model LLama jest używany do generowania odpowiedzi na podstawie zadanych pytań. Korzysta z metod zaawansowanego przetwarzania języka naturalnego (NLP) do analizowania tekstu i generowania odpowiedzi.

- Integracja z modelem:

Backend aplikacji przesyła wyodrębnione fragmenty tekstu z materiałów PDF do modelu LLama-2-13B-Chat, który następnie generuje odpowiedzi na podstawie kontekstu pytania.

Parametry modelu LLama-2-13B-Chat

Model ten charakteryzuje się 13 miliardami parametrów. Jest to model optymalizowany do zadań konwersacyjnych. Dokładność odpowiedzi oraz zdolność do generowania kontekstu są dostosowane do wymagań aplikacji edukacyjnej. Model w formacie llama-2-13b-chat.Q2_K.gguf zapewnia wystarczającą precyzję przy optymalnym wykorzystaniu pamięci.

Opis funkcjonalności

- **Interfejs użytkownika:**

Aplikacja oferuje prosty interfejs użytkownika, który został wykonany przy pomocy biblioteki streamlit. Pozwala na zadawanie pytań oraz otrzymywanie odpowiedzi w czasie rzeczywistym.

- **Przetwarzanie dokumentów PDF:**

Aplikacja automatycznie przetwarza i indeksuje pliki PDF, wyodrębniając z nich tekst. Wyodrębnione dane są przechowywane w bazie wektorowej, co umożliwia szybkie przeszukiwanie.

- **Generowanie odpowiedzi:**

Na podstawie zadanego pytania, aplikacja przeszukuje bazę danych, aby znaleźć odpowiednie fragmenty tekstu, a następnie przesyła je do modelu LLama, który generuje odpowiedź.

- **Optymalizacja:**

Aplikacja została zoptymalizowana pod kątem wydajności, aby działać na maszynach z ograniczoną ilością pamięci RAM, co było kluczowe ze względu na lokalne uruchamianie modelu LLama.