İTÜ MTH409: AI Chatbot Dersi 2025 Güz Dönemi

Proje Raporu

Geliştirici :Öykü Deniz Zengin

Basic RAG ile AI Chatbot Geliştirme

Bu projede RAG yöntemi kullanılarak bir chatbot geliştirilmesi amaçlanmıştır. Veri seti olarak "The GALE ENCYCLOPEDIA of MEDICINE SECOND EDITION" [1] isimli kitap kullanılmıştır.

Tıbbi Asistan Chatbot

Kullanıcıyı tıbbi hastalıkları açıklama, tedavi ve belirtileri hakkında bilgilendiren bir chatbot geliştirme amaçlanmıştır.

Proje Mimarisi

- **Veri Hazırlığı**
  Veri Yükleme: PDF dosyamızı pypdf kütüphanesi ile okutulur ve metne dönüştürülür.
  Metin Bölümleme: 'Text Splitter' ile 1000 karakterlik küçük parçalara metin ayrılır (chunking).
  Vektörleştirme: HuggingFace'in "Sentence Transformers" modeli kullanarak chunkslar sayısal vektörlere dönüştürülür.
  Vektör Veritabanı: Oluşturulan vektörler, anlamsal arama yapılması ve LLM'e iletilmesi için bulut ortamında tutulmak üzere Pinecone Veritabanı kullanılır.
- **Sohbet Akışı**
  Sırasıyla kullanıcı ve chatbot arasında bu adamlar gerçekleşir:
  1.Kullanıcı Arayüzü: Flask ile hazırlanmış web arayüzü ile kullanıcı sorusunu iletir.
  2.Anlamsal Arama (Similarity Search): Kullanıcı sorusu vektöre çevrilerek Pinecone veritabanından "en çok benzeyen doküman parçaları" araması yapılır.
  3.Prompt Oluşturma: Getirilen dokümanlar ve kullanıcı sorusu birleştirilerek LangChain kullanılarak bir prompt (istem) haline getirilir.
  4.Yanıtın Oluşması: Oluşan prompt LLM modeline iletilerek, LLM modelinin ilgili bağlam içerisinde soruya bir cevap üretir.
  5.Sonuç: Flask arayüzünden kullanıcıya oluşan yanıt iletilir.

Kullanılan Araçlar

LangChain: Kullanılan tüm araçları birbirine bağlayarak projenin iskeletini oluşturur.

PyPDF: Elimizdeki PDF dosyasının okuyup metne çevirilir.

HuggingFace: Metin parçalarını embedding modeli ile sayısal vektörlere dönüştürür.

Google Gemini: Doğal dil modeli kullanıcının yanıtını oluşturur.

Groq Llama3: Bir diğer LLM modeli karşılaştırma yapılması adına kullanılır.

Pinecone: Metinden çevrilen vektörler Pinecone veritabanında bulut tabanlı bir ortamda bulunur. Hızlı ve kolay erişim sağlanır.

Flask: Uygulamanın kullanım kolaylığı açısından bir web tarayıcısında frontend iletişimini kurarak uygulamayı çalıştırır.

LLM Modelleri

- **Google Gemini**
  Google AI Studio'dan API key alarak "2.5-flash-lite" Gemini modeli kullanıldı.
- **Groq Llama3**
  Hızlı bir şekilde LLM'leri çalıştıran Groq sağlayıcısı ile Llama 3 modeli kullanıldı.

Modellerin Karşılaştırılması

İki modelin karşılaştırılması için RAGAS Framework'ü kullanılarak içerisindeki Faithfulness ve Answer Relevancy metrikleri ile hesaplamalar yapılmıştır.

Beş sorudan oluşan test seti ile Gemini modelini jüri olarak kullanıp değerlendirme işlemi gerçekleştirilir.

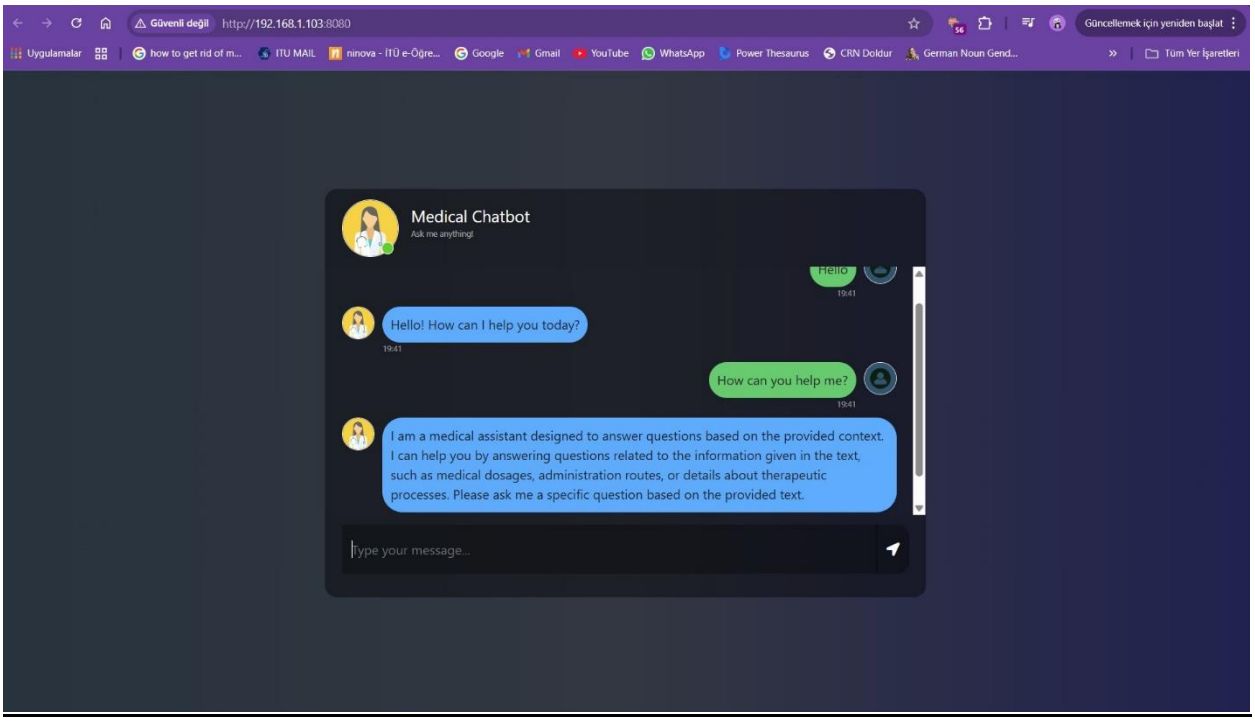
Model	Faithfulness	Answer_relevancy
Gemini	1	0.886
Llama3	0.871	0.918

Her iki modelin de 5'er sorudan gelen skorların ortalaması alınarak genel puanlar hesaplanmıştır.

Kullanılan API Key'ler

- **Pinecone API Key**
  Pinecone vektör veritabanımız için.
- **Google API Key**
  Gemini modeli olan chatbotumuz için.
- **Groq API Key**
  LLama modeli olan chatbotumuz için.
- **Google API Key (RAGAS)**
  İki modelimiz için ayrı API Key'ler ile jüri oluşturduk.

Proje Arayüzü



Proje Çıktıları, Dikkat Edilmesi Gereken Unsurlar ve Yorumlar

Paket versiyon çakışmaları

Langchain kullanımında paket yapısı güncellenebilme durumu olduğundan pip install komutunu ortamınızda çalıştırıp yeni bir paket yüklediğinizde versiyon çatışmaları sebebiyle projeniz çalışmayabilir. Oldukça fazla kez karşıma çıkan bir problem oldu.
İki LLM modeli için ayrı sanal ortamlar kurarak gerekli paketleri indirdim ve daha temiz ve garanti bir çalışma şekli elde edebildim.
Langchain dokümantasyon sayfalarından güncel olarak paketi import etmek için hangi komut gerektiğini sıklıkla kontrol etmem gerekti.

API Key Free Trial Usage Limit

Eğer LLM kullanımlarında ücretsiz sürümlerden yararlanıyorsanız, uzun bir süre modeli deneyerek çalışmanızı yürütürseniz belirli bir zaman içerisindeki istek sınırına ulaşmanız olasıdır. Bu projede ücretsiz bir şekilde Google AI Studio ve Groq LLM Inference kullanarak API Key'ler oluşturuldu.
Ancak yukarıda daha önce de belirtildiği gibi Google AI Studio'da proje başına olan kullanım limitini özellikle RAGAS işlemini yaparken aşmanız çok olasıdır, bu nedenle ayrı projeler ile farklı API Key'ler oluşturmanızı öneririm.

Veri Seti Yetersizliği

Projeye geliştirme önerisi ve eleştiri olarak daha kapsamlı bir veri setiyle çalışabildiği takdirde daha esnek ve verimli bir chatbot geliştirilebiliceğini düşünüyorum.
Bazı sohbetlerde similarity search yanlış çalışabiliyor ve chatbotun kullanıcıyı yanlış anladığı ve yanlış bağlamı getirdiği durumlar oluşabildiğini gözlemledim.
Bununla beraber son olarak PDF'imizde Latince Tıbbi rahatsızlık isimleri geçmesi ve kaynağın İngilizce olması kullanıcının başka dillerde bir sorgu oluşturduğunda chatbotun anlamlı cevap verememesine sebebiyet verebildiğini gözlemledim.

Özet

Bu projenin AI Chatbot yapımının temellerini kavramak ve konseptlere giriş yapmak için verimli bir başlangıç olduğunu düşünüyorum.
Basic bir RAG projesiyle nasıl chatbot yapılabileceğiyle ilgili pratik yapabileceğiniz, temelde basit gibi olsa da tüm projeyi inşaa ederken minik detaylarla zorlanarak daha çok bilgilenebileceğiniz bir deneyim sunuyor.