

Akıllı Yönlendirmeli Agentic AI Chatbot

Hazırlayan: İrem KUMLU

İçindekiler

1. Giriş.....	3
o Projenin Amacı ve Kapsamı	
o Temel Yaklaşım: Akıllı Yönlendirme	
2. Temel Fikir ve Mimari.....	3
o Supervisor-Agent Mimarisi Açıklaması	
▪ Supervisor (Denetleyici) Rolü	
▪ Alt Agent'lar (Uzmanlar): Resmi Gazete, Haber/Genel Bilgi, Fallback	
o LangGraph Kullanımı	
3. İş Akışı: Kullanıcıdan Cevaba.....	3
o Adım Adım Süreç: Soru Alma, Analiz, Yönlendirme, İşlem, Cevap Sunma	
o Agent'ların Çalışma Mekanizmaları (RAG ve Araç Kullanımı)	
4. Kullanılan Ana Teknolojiler ve Veri Kaynakları.....	4
o Ana Kütüphaneler (LangChain, LangGraph)	
o Yapay Zeka Modelleri (Gemini, Embedding)	
o Veri Depolama (ChromaDB)	
o Veri Kaynakları (PDF'ler, Wikipedia, Tavily)	
o Arayüz ve Dağıtım (Streamlit, Docker)	
5. Kullanıcı Arayüzü ve Çalıştırma Kolaylığı.....	5
o Streamlit Arayüzünün Özellikleri	
o Docker ile Kolay Kurulum ve Çalıştırma	
6. Sonuç ve Değerlendirme.....	5
o Projenin Başarısı ve Kazanımları	
o Genel Değerlendirme	

1. Giriş

Bu projenin temel amacı, kullanıcılar tarafından sorulan soruların içeriğini anlayarak, bu soruları en doğru bilgi kaynağıyla eşleştiren ve anlamlı cevaplar üreten bir yapay zeka sohbet botu (chatbot) geliştirmektir. Sistem, bir sorunun Türkiye Cumhuriyeti Resmi Gazetesi ile mi ilgili, yoksa güncel haberler veya genel ansiklopedik bilgilerle mi ilgili olduğunu ayırt edebilmektedir. Bu ayırım yapıldıktan sonra, soruyu cevaplamak için en uygun yöntem (belge arama veya dış araçları kullanma) otomatik olarak seçilir. Projenin temelini, görevleri merkezi bir şekilde yöneten ve uzmanlaşmış alt birimlere dağıtan modern bir "Supervisor-Agent" yapay zeka mimarisi oluşturmaktadır. Kullanıcılar sistemle basit bir web arayüzü üzerinden etkileşim kurabilir ve tüm sistem, kolay kurulum ve çalıştırma için paketlenmiştir.

2. Temel Fikir ve Mimari

Proje, günümüz yapay zeka uygulamalarında sıklıkla kullanılan modüler bir yaklaşım olan **Supervisor-Agent mimarisi** üzerine kurulmuştur. Bu mimarinin ana fikri şudur:

- **Supervisor (Denetleyici):** Sistemin beyni gibidir. Gelen her soruyu ilk olarak o karşılar ve sorunun ne hakkında olduğunu anlamaya çalışır. Ardından, bu soruyu en iyi hangi uzmanın cevaplayabileceğine karar verir ve görevi o uzmana devreder. Bu projede supervisor, bir yapay zeka dil modeli (Google Gemini Flash) kullanarak soruyu sınıflandırır.
- **Alt Agent'lar (Uzmanlar):** Her biri belirli bir görev türünde uzmanlaşmış birimlerdir:
 1. **Resmi Gazete Agent'ı:** Sadece Resmi Gazete'de yayınlanmış veya yayınlanması muhtemel konularla (kanunlar, yönetmelikler, devlet destekleri, resmi ilanlar vb.) ilgili soruları cevaplamak üzere tasarlanmıştır. Bu agent, özel olarak hazırlanmış bir belge veritabanında arama yaparak çalışır.
 2. **Haber/Genel Bilgi Agent'ı:** Güncel olaylar, son dakika haberleri, "Kimdir?", "Nedir?" gibi genel kültür soruları veya ansiklopedik bilgi gerektiren konularla ilgilenir. Bu agent, cevabı bulmak için Wikipedia veya web arama motorları (Tavily) gibi dış kaynakları kullanır.
 3. **Fallback Agent:** Yukarıdaki iki uzmanın alanına girmeyen veya cevaplamakta zorlandığı sorular olduğunda devreye girer. Kibarca sorunun cevaplanamadığını belirtir.

Bu mimari, sistemi daha yönetilebilir, esnek ve geliştirilebilir kılar. Her agent kendi görevine odaklanır ve gerektiğinde sadece ilgili agent üzerinde değişiklik yapmak yeterli olur. Bu yapıyı kurmak ve yönetmek için **LangGraph** kütüphanesi kullanılmıştır. LangGraph, supervisor'ın kararından sonra iş akışının doğru agent'a yönlendirilmesini ve süreç boyunca bilginin (örneğin kullanıcının sorusu) agent'lar arasında taşınmasını sağlar.

3. İş Akışı: Kullanıcıdan Cevaba

Sistemin çalışma adımları şu şekildedir:

1. **Soru Alma:** Kullanıcı, web arayüzü (Streamlit) üzerinden sorusunu yazar.
2. **Supervisor Analizi:** Soru, merkezi denetleyici olan Supervisor Agent'a iletilir. Supervisor, yapay zeka (LLM) kullanarak soruyu analiz eder ve üç kategoriden birine atar: "Resmi Gazete", "Haber/Genel Bilgi" veya "İlgisiz".
3. **Yönlendirme:** Supervisor'ın kararına göre, görev ilgili Alt Agent'a yönlendirilir.
4. **Agent İşlemi:**

- **Resmi Gazete Agent'ı seçildiyse:** Agent, önceden işlenmiş ve vektör temsillerine dönüştürülmüş Resmi Gazete belgelerinin saklandığı **ChromaDB** adlı özel veritabanında anlamsal bir arama yapar. Soruyla en ilgili belge parçalarını bulur. Ardından, bu belgeleri ve orijinal soruyu yapay zekaya (LLM) vererek, sadece bu belgelere dayalı bir cevap oluşturmasını ister (Bu yöntem RAG - Retrieval-Augmented Generation denir).
 - **Haber/Genel Bilgi Agent'ı seçildiyse:** Agent, sorunun türüne göre önce **Wikipedia**'yı veya daha güncel bilgiler için **Tavily** web arama motorunu kullanır. Bu araçlardan gelen bilgileri alır ve yine yapay zeka (LLM) kullanarak kullanıcıya sunulacak nihai cevabı düzenler ve özetler.
 - **Fallback Agent seçildiyse:** Agent, önceden tanımlanmış "Üzgünüm, bu soruya yanıt veremiyorum" benzeri bir mesajı hazırlar.
5. **Cevap Sunma:** Seçilen agent tarafından üretilen cevap, hangi kaynaktan geldiği bilgisiyle birlikte kullanıcıya web arayüzünde gösterilir.

4. Kullanılan Ana Teknolojiler ve Veri Kaynakları

Projenin temel taşlarını oluşturan teknolojiler ve kullanılan veriler şunlardır:

- **Ana Kütüphaneler:**
 - **LangChain:** Yapay zeka uygulamaları geliştirmek için genel bir çerçeve; agent'ları, LLM'lerle etkileşimi ve araçları (Wikipedia, Tavily) yönetmek için kullanıldı.
 - **LangGraph:** Supervisor-Agent gibi karmaşık iş akışlarını modellemek ve yönetmek için kullanıldı.
- **Yapay Zeka Modeli:**
 - **Google Gemini (Flash modeli):** Soruları sınıflandırmak (supervisor), belgelerden cevap üretmek (RAG) ve araçlardan gelen bilgileri işlemek için kullanılan ana dil modeli.
 - **Google Embedding Modeli:** Resmi Gazete metinlerini bilgisayarın anlayabileceği sayısal vektörlere dönüştürmek için kullanıldı.
- **Veri Depolama:**
 - **ChromaDB:** Resmi Gazete metinlerinin vektörlerini saklamak ve hızlı arama yapmak için kullanılan yerel bir vektör veritabanı.
- **Veri Kaynakları:**
 - **Resmi Gazete PDF'leri:** data klasöründe bulunan örnek PDF dosyaları, RAG sisteminin bilgi tabanını oluşturdu.
 - **Wikipedia & Tavily Search:** Haber/Genel Bilgi Agent'ının kullandığı harici bilgi kaynakları.
- **Arayüz ve Dağıtım:**
 - **Streamlit:** Kullanıcıların kolayca soru sorup cevap alabileceği interaktif web arayüzünü oluşturmak için.
 - **Docker & Docker Compose:** Tüm projenin bağımlılıklarıyla birlikte paketlenerek herhangi bir bilgisayarda tek bir komutla kolayca çalıştırılabilmesini sağlamak için.

Bu, "konteynerleştirme" olarak bilinir ve projenin kurulumunu ve paylaşımını çok basitleştirir.

5. Kullanıcı Arayüzü ve Çalıştırma Kolaylığı

Kullanıcılar için **Streamlit** tabanlı basit ve anlaşılır bir sohbet arayüzü geliştirilmiştir. Kullanıcılar sorularını metin kutusuna yazabilir ve sistemin ürettiği cevapları sohbet formatında görebilirler. Cevabın hangi agent (ve dolayısıyla hangi kaynak) tarafından üretildiği de belirtilir.

Docker kullanımı sayesinde, projenin çalıştırılması teknik bilgisi olmayan kişiler için bile oldukça kolaydır. Gerekli tüm kurulumlar Docker tarafından otomatik olarak yapılır ve kullanıcıların sadece docker-compose up gibi basit bir komut çalıştırması yeterlidir. API anahtarları gibi hassas bilgiler de güvenli bir şekilde .env dosyası üzerinden yönetilir.

6. Sonuç ve Değerlendirme

Bu proje, kullanıcı sorularını akıllıca analiz edip doğru bilgi kaynağına yönlendiren ve bu kaynakları kullanarak cevap üreten supervisor-agent tabanlı bir chatbot sistemini başarıyla hayata geçirmiştir. LangGraph ile modellenen mimari, sistemin modülerliğini ve esnekliğini sağlarken, RAG ve harici araçların kullanımı farklı türdeki sorulara cevap verme yeteneğini kazandırmıştır. Streamlit arayüzü ve Docker ile paketleme, projenin kullanımını ve dağıtımını kolaylaştırmıştır.

Sistem genel olarak beklentileri karşılamakla birlikte, supervisor'ın yönlendirme doğruluğu veya RAG sisteminin bulduğu bilgilerin kalitesi gibi alanlarda her zaman iyileştirme potansiyeli bulunmaktadır.