

# LangChain ve OpenAI ile Soru-Cevap Sistemi Oluşturmak

---

## Giriş

Yapay zekâ tabanlı soru-cevap sistemleri, bilgiye erişimi hızlandırmak ve kullanıcı deneyimini zenginleştirmek amacıyla modern yazılım mimarilerinin merkezinde yer almaktadır. Bu sistemler, kullanıcıların doğal dilde yönelttiği sorulara doğru, tutarlı ve bağlamsal olarak uygun yanıtlar üretebilmek için büyük dil modellerinin (LLM – Large Language Model) gücünden yararlanır.

Bu bağlamda LangChain ve OpenAI ekosistemleri, geliştiricilere modüler, ölçeklenebilir ve kolay uygulanabilir bir altyapı sunarak bilgiye dayalı yapay zekâ sistemleri kurmayı mümkün kılmaktadır.

## Sistem Mimarisi

LangChain ve OpenAI kullanılarak geliştirilen bir soru-cevap sistemi genel olarak şu bileşenlerden oluşur:

1. Veri Kaynağı
2. Metin İşleme
3. Vektörleştirme
4. Vektör Veritabanı
5. Retriever Modülü
6. Dil Modeli Entegrasyonu
7. Zincirleme Yapı (Chain Orchestration)

## Retrieval-Augmented Generation (RAG)

RAG yaklaşımı, büyük dil modellerinin bilgiye dayalı şekilde yanıt üretmesini sağlar. Modelin ezberlediği bilgiler yerine, harici kaynaklardan dinamik olarak bilgi çekmesi sağlanır. Bu mimaride doğruluk ve kaynak güvenilirliği en önemli avantajlardır.

## LangChain'in Rolü

LangChain, LLM uygulamaları için bir yapı iskelesi (framework) görevi görür. Geliştiricilere bileşen soyutlama, zincirleme işlem mantığı, bellek yönetimi, agent desteği ve geniş entegrasyon ekosistemi gibi önemli avantajlar sunar.

## OpenAI Entegrasyonu

OpenAI, sistemde iki ana işlevi üstlenir: embeddings ve dil modeli. Embeddings işlemi metinleri sayısal vektörlere dönüştürürken, dil modeli bağlama dayalı olarak anlamlı cevaplar üretir.

## Örnek Uygulama Akışı

Aşağıdaki Python örneği, LangChain ve OpenAI kullanarak temel bir soru-cevap sistemi oluşturma sürecini göstermektedir.

```
from langchain.llms import OpenAI
from langchain.chains import RetrievalQA
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.embeddings import OpenAIEmbeddings
from langchain.vectorstores import FAISS

docs = ["Yapay zekâ, insan zekâsını taklit eden sistemlerin geliştirilmesidir."]
splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=50)
chunks = splitter.split_text(docs[0])

embedding = OpenAIEmbeddings()
db = FAISS.from_texts(chunks, embedding)
retriever = db.as_retriever()
llm = OpenAI(temperature=0)

qa_chain = RetrievalQA.from_chain_type(llm=llm, retriever=retriever)
print(qa_chain.run("Yapay zekâ nedir?"))
```

## Performans, Güvenlik ve Ölçeklenebilirlik

Gerçek dünyada kullanılan sistemlerde performans, güvenlik, maliyet ve sürdürülebilirlik konuları büyük önem taşır. Özellikle vektör arama, API anahtar güvenliği ve model maliyetleri dikkatle yönetilmelidir.

## Sonuç

LangChain ve OpenAI'nin birleşimi, bilgiye dayalı akıllı sistemlerin geliştirilmesini son derece kolaylaştırmaktadır. Bu teknolojiler bir araya geldiğinde güvenilir, kaynak tabanlı ve doğal dilde etkileşim kurabilen profesyonel sistemler üretmek mümkündür.