

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
OF TEKNOLOJİ FAKÜLTESİ YAZILIM  
MÜHENDİSLİĞİ BÖLÜMÜ**



**EXPLAINABLE AI HAKKINDA UZMAN BİR  
YAPAY ZEKA BOTUNUN TASARIMI**

**BİTİRME PROJESİ**

**Mehmet Emin AK, Elif Beyza TOK**

**2023-2024 BAHAR DÖNEMİ**

**KARADENİZ TEKNİK ÜNİVERSİTESİ OF  
TEKNOLOJİ FAKÜLTESİ YAZILIM  
MÜHENDİSLİĞİ BÖLÜMÜ**

**EXPLAINABLE AI HAKKINDA UZMAN BİR  
YAPAY ZEKA BOTUNUN TASARIMI**

**BİTİRME PROJESİ**

**Mehmet Emin AK, Elif Beyza TOK**

**Bu projenin teslim edilmesi ve sunulması tarafımdan uygundur.**

**Danışman : Dr. Öğr. Üyesi Mustafa Hakan BOZKURT .....**

**2023-2024 BAHAR DÖNEMİ**



## **IEEE Etik Kuralları IEEE Code of Ethics**



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

## ÖNSÖZ

LLM Modeller hayatımıza yeni girmesine rağmen oldukça hızlı geliyorlar ve hayatımızı büyük ölçüde etkiliyorlar. Ekip arkadaşım ile birlikte böylesine önemli bir konu üzerinde çalışmak oldukça keyifli ve geliştirici bir süreçti. Tüm bu süreç boyunca desteğini esirgemeyen danışmanımız Dr. Öğr. Üyesi Mustafa Hakan BOZKURT'a da çok teşekkür ederiz.

Mehmet Emin AK, Elif Beyza TOK  
Trabzon, 2024

## İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI	II
ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	V
ŞEKİLLER DİZİNİ	VI
TABLolar DİZİNİ	VII
SEMBOLLER DİZİNİ	VIII
1. GENEL BİLGİLER	1
1.1. Giriş	1
2. YAPILAN ÇALIŞMALAR	4
2.1. Yazılım Yaşam Döngüsü	4
2.1.1. Planlama	4
2.1.2. Analiz	7
2.1.3. Tasarım	8
2.1.4. Gerçekleştirim	9
2.1.5. Test	23
2.1.5.1. Genel Test Planı	23
2.1.5.2. Test Tanımlama Belgesi	27
2.1.5.3 Test Sonuç Raporu	29
3. SONUÇLAR	30
4. ÖNERİLER	31
5. KAYNAKLAR	32
6. EKLER (varsa)	34
STANDARTLAR ve KISITLAR FORMU	35

## ÖZET

Bu tezde, META tarafından geliştirilen ve ücretsiz kullanım lisansı ile sunulan Llama2 adlı bir Büyük Dil Modeli (LLM) üzerinde yapılan çalışmaların detayları sunulmaktadır. Çalışma kapsamında, Explainable AI (Açıklanabilir Yapay Zeka) konulu makalelerin Llama2 modeline gömülmesi için embedding işlemi gerçekleştirilmiştir. Langchain kütüphanesi kullanılarak az miktardaki veriyle bile modelin başarıyla eğitildiği gözlemlenmiştir. Ayrıca, bu çalışmada modelin verdiği cevapların kaynaklarıyla birlikte belirlenebilmesi için PDF formatındaki veri setleri önce vektör veri tabanında tutulmuş, daha sonra bu veriler küçük parçalara ayrılarak Pinecone bulut ortamında saklanmıştır. Pinecone veri tabanı, modelin cevaplarına ek olarak aldığı kaynakların (metadatanın) sunulmasını sağlamıştır. Veri kümeleri temizlenerek ve RAG (Resource Augmentation Generation) tekniği kullanılarak Llama2 modelinin performansı optimize edilmiş ve Pinecone veri tabanına yüklenmiştir. Son olarak, modelin sunulması için LangServe API'leri, FastAPI ve Ngrok kullanılarak bir web sunucusu oluşturulmuş ve bu sunucuya HTML, CSS, JavaScript ile geliştirilen bir web arayüzü entegre edilmiştir. Bu arayüz, Explainable AI hakkında sorulan sorulara cevap veren ve cevapların kaynak doküman ve sayfasını gösteren Llama2 modelini kullanıcıya sunmaktadır.

**Anahtar Kelimeler:** Büyük Dil Modeli modeli, , Explainable AI , Llama2 , Fine-tuning, embedding, Pinecone, Langchain, veri temizleme, RAG, LangServe API'leri, FastAPI, Ngrok

## ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. LLM Modellerin Yapay Zekasındaki Yerini Gösteren Görsel.....	01
Şekil 2. Mevcut Büyük Dil Modeli ( LLM ) Ortamını Gösteren Bir Grafik.....	02
Şekil 3. Llama2 Logosu .....	03
Şekil 4. İş Akışı Diyagramı Görsel .....	07
Şekil 5. Llama-2-Chat Modelinin Çalışma Şeklini Gösteren Görsel .....	09
Şekil 6. Langchain Framwork'ünün Çalışma Şeklini Gösteren Görsel.....	10
Şekil 7. Hugging Face Pipelines'ının Çalışma Şeklini Gösteren Görsel.....	10
Şekil 8. Vektörler Arasında Yapılan Benzerlik Aramasının Çalışma Şeklini Gösteren Görsel.....	11
Şekil 9. Vektörler Veri Tabanı Kullanan LLM'lerin Çalışma Şeklini Gösteren Görsel..	11
Şekil 10. Vektör Veri Tabanı Kullanan LLM'lerin Çalışma Şeklini Gösteren Görsel...	13
Şekil 11. RetrievalQA Çalışma Adımlarını Gösteren Görsel.....	15
Şekil 12. RAG Tekniğinin Çalışma Adımlarını Gösteren Görsel.....	16
Şekil 13. PineconeDB'de Verilerin Nasıl Tutulduğunu Gösteren Konsol Ekranı Görüntüsü.....	17
Şekil 14. Projenin Web Arayüzü Görseli.....	20
Şekil 15. Modelin Verdiği Cevabın JSON Formatındaki Görseli.....	21
Şekil 16. Modelin Verdiği Cevabın Web Arayüzündeki Görseli.....	21
Şekil 17. Sistemin Blok Diyagramı.....	22
Şekil 18. Kullanım Senaryosu (Use Case) Diyagramı.....	22

## TABLÖLAR(ÇİZELGELER) DİZİNİ

	<b><u>Sayfa No</u></b>
Tablo 1. İş Zaman Çizelgesi .....	05
Tablo 2. Risk Yönetimi Tablosu .....	05
Tablo 3. Sistem Mimarisi .....	08
Tablo 4. Test Takvimi.....	25
Tablo 5. Test Tanımlama Belgesi.....	27
Tablo 6. Test Sonuç Raporu.....	29



## SEMBOLLER DİZİNİ

**LLM:** Büyük Dil Modeli (Large Language Model)  
**EAI:** Açıklanabilir Yapay Zeka (Explainable AI)  
**META:** Yapay Zeka Araştırma ve Geliştirme Şirketi  
**RAG:** Kaynak Artırma ve Üretme (Resource Augmentation Generation)  
**PDF:** Taşınabilir Belge Biçimi (Portable Document Format)  
**API:** Uygulama Programlama Arayüzü (Application Programming Interface)  
**HTML:** Hipertext İşaretleme Dili (Hypertext Markup Language)  
**CSS:** Yapısal Biçimlendirme Dili (Cascading Style Sheets)  
**JavaScript:** Tarayıcı Dili (JavaScript)  
**FastAPI:** Hızlı ve Modern Web API Geliştirme (FastAPI)  
**Ngrok:** Yerel Sunucuları İnternete Bağlama Aracı (Ngrok)  
**LangServe:** Dil Hizmeti (Language Service)

## 1. GENEL BİLGİLER

### 1.1. Giriş

Büyük dil modelleri (LLM'ler), son yıllarda yapay zeka alanında büyük bir gelişme göstermiştir. Bu modeller, metin üretme, dilleri çevirme ve soruları yanıtlama gibi birçok karmaşık görevi yerine getirebilir.

Explainable AI (EAI), yapay zeka modellerinin nasıl çalıştığını anlamaya odaklanır. EAI, modellerin daha şeffaf ve güvenilir olmasını sağlayarak yapay zekanın benimsenmesini teşvik etmeyi amaçlar.

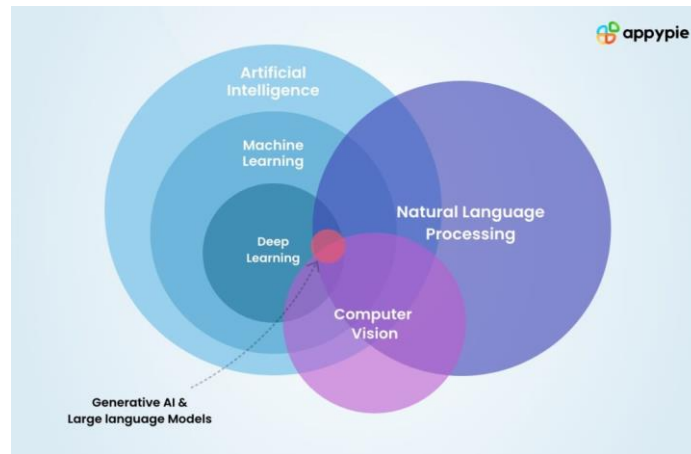
RAG (Resource Augmentation Generation), büyük dil modelinin ek veri kaynaklarından yararlanmasına olanak tanıyan bir tekniktir.

Bu çalışmada META'nın Llama2 modeli EAI ile geliştirildi. Llama2'ye Explainable AI makaleleri gömüldü ve bu konuyla ilgili sorulara cevap verme yeteneği kazandı. RAG tekniği kullanılarak ek veri kaynaklarından yararlanması sağlandı.

### LLM Nedir?

LLM( Large Language Model) yani Büyük Dil Modelleri, insan dilini anlamak, oluşturmak ve yönetmek için tasarlanmış yapay zeka destekli sistemlerdir. Bu modeller genellikle onlarca gigabayt boyutundadır ve genellikle derin öğrenme teknikleri kullanılarak oluşturulur; en dikkate değer mimari ise Transformer'dır. Transformer mimarisi, modellerin bir cümledeki kelimelerin bağlamını ve bunların ilişkilerini yakalamasını sağlayarak tutarlı ve bağlamsal olarak alakalı metinler oluşturmalarına olanak tanır.

Büyük dil modelleri kavramı, OpenAI'nin GPT (Generative Pre-trained Transformer) gibi modelleriyle ortaya çıkmaya başladı. Bu modeller şaşırtıcı derecede insana benzeyen metinler üretme yetenekleriyle ün kazandı. Bu büyük dil modelleri; internetten, kitaplardan, makalelerden ve diğer metin kaynaklarından içerik içeren büyük veri kümeleri üzerinde önceden eğitilmiştir. Ön eğitim süreci modellere genel bir dil ve dünya bilgisi anlayışı kazandırır.[1]



Şekil 1. LLM Modellerin Yapay Zekasındaki Yerini Gösteren Görsel [15]

Büyük Dil Modelleri (LLM'ler) işlevselliği beş alana ayrılabilir: Bilgi Yanıtlama, Çeviri,

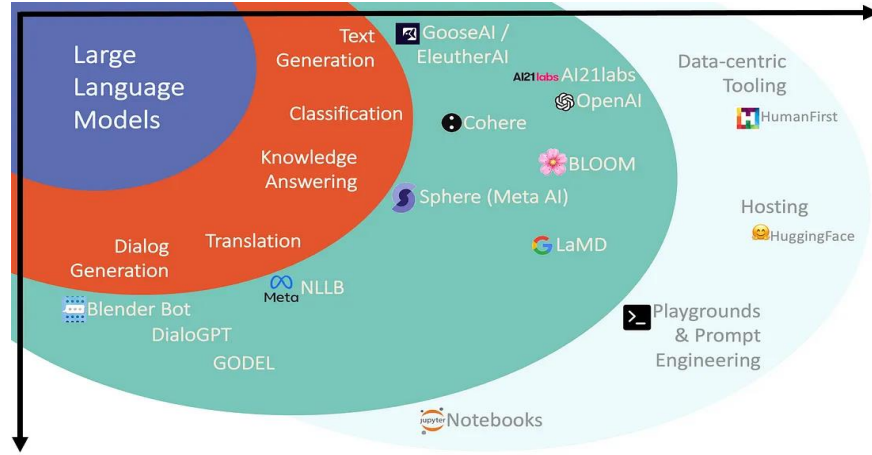
Metin Oluşturma, Yanıt Oluşturma ve Sınıflandırma.

Günümüzün kurumsal ihtiyaçları açısından tartışmasız en önemli olanı sınıflandırmadır ve metin oluşturma da en etkileyici ve çok yönlü olanıdır.

Çeşitli LLM teklifleri bu beş işlevsellik alanını değişen derecelerde kapsar.

Burada bahsedilen teknolojilerin çoğuna HuggingFace sitesi üzerinden erişilebilir .

Bu projede kullanılacak olan LLM Model Llama2'ye de HuggingFace sitesi üzerinden ulaşılabilir.[2]



Şekil 2. Mevcut Büyük Dil Modeli ( LLM ) Ortamını Gösteren Bir Grafik [16]

Bu proje için ücretsiz ve (tercihen) Offline(Çevrimdışı) çalışabilen bir modele ihtiyaç vardır.

HuggingFace sitesi birçok alanda birçok hazır model sunan bir site. Buradaki modeller tarafımızca incelendi ve en uygun, en kullanışlı olan modelin Llama 2 olduğuna karar verildi.

OpenAI' da hazır modeller sunuyordu fakat bu modeller hem offline olarak çalıştırılmadığından yani lokal cihaza getirilemediğinden hem de ücretli olduklarından dolayı kullanımı tercih edilmedi.

### Llama2 nedir?

Llama 2, Meta şirketinin açık kaynaklı büyük dil modelidir (LLM). Temel olarak, bu, Facebook ana şirketinin OpenAI'nin GPT modellerine ve Google'ın Palm 2 gibi AI modellerine verdiği yanıttır. Ancak önemli bir farkla: neredeyse herkesin araştırma ve ticari amaçlarla kullanması için ücretsiz olarak sunulmaktadır.

Llama 2, GPT-3 ve PaLM 2 gibi LLM ailesindendir. Tüm bu modeller temelde aynı şekilde

geliştirilmiş ve çalışmaktadır. Hepsi aynı transformatör mimarisini ve ön eğitim ve ince ayar gibi geliştirme fikirlerini kullanır.

Bir metin istemine girdiğinizde veya Llama 2'ye başka bir şekilde metin girişi sağladığınızda, kendi sinir ağını kullanarak en makul devam eden metni tahmin etmeye çalışır. Bu, milyarlarca değişken (parametre) içeren basamaklı bir algoritmadır ve insan beyni baz alınarak modellenmiştir.

Llama 2, tüm farklı parametrelere farklı ağırlıklar atayarak ve biraz rastgelelik ekleyerek inanılmaz derecede insani tepkiler üretebilir. [3]



Şekil 3. Llama2 Logosu [17]

Llama 2'nin, ChatGPT veya Google Bard gibi gösterişli, kullanımı kolay bir demo uygulaması henüz bulunmuyor . Şimdilik bunu denemenin en iyi yolu , açık kaynaklı yapay zeka modelleri için başvurulacak merkez haline gelen platform olan [Hugging Face](https://huggingface.co)'tir. Hugging Face aracılığıyla Llama2'nin aşağıdaki sürümlerini deneyebilirsiniz:

- [Llama 2 7B Chat \[4\]](#)
- [Llama 2 13B Chat \[5\]](#)
- [Llama 2 70B Chat \[6\]](#)

#### 2.1.1.7.2.3. Llama2'nin Projede Kurulumu ve Kullanımı

Projede Llama2 modelini kullanmak için Hugging Face'in sağladığı API'lerden yararlanıldı. Proje içerisine yazılan aşağıdaki kodlar ile bu işlem gerçekleştirildi:

```
!pip install langchain huggingface pytorch
import transformers

from huggingface_hub import notebook_login
notebook_login()

from transformers import AutoModelForCausalLM, AutoTokenizer

#Huggingface'den modeli getiriyoruz
model = AutoModelForCausalLM.from_pretrained("meta-llama/Llama-2-7b-chat-hf")
```

## **2.YAPILAN ÇALIŞMALAR**

### **2.1. Yazılım Yaşam Döngüsü**

#### **2.1.1. Planlama**

##### **2.1.1.1. Proje Hedefleri**

Bu çalışmanın temel amacı, Llama2 LLM modelini EAI konusunda uzmanlaştırarak EAI hakkında sorulan sorulara cevap vermektir.

##### **2.1.1.2. Proje Kapsamı**

Bu çalışma, Llama2 LLM modeli üzerinde EAI uygulamalarını geliştirmeye odaklanmaktadır. Çalışmada, modelin Explainable AI konulu makaleler üzerinde eğitilmesi, modelin verdiği cevapların kaynaklarını göstermesi için bir mekanizma geliştirilmesi, modelin verdiği cevapların daha kaliteli hale getirilmesi için veri seti üzerinde temizleme işlemleri uygulanması, model sunucusu oluşturulması ve ayağa kaldırılması ve modelin kullanıcılara sunulması için bir web arayüzü geliştirilmesi gibi işlemler yer almaktadır. Çalışmada, EAI'nın diğer alanlarına veya farklı LLM modellerine ilişkin çalışmalar yapılmamıştır.

### 2.1.1.3. İş- Zaman Çizelgesi

İP No	İş Paketlerinin Adı ve Hedefleri	Kim(ler) Tarafından Gerçekleştirileceği	Zaman Aralığı (..-.. Ay)	Başarı Ölçütü ve Projenin Başarısına Katkısı
1	Literatür Araştırması, Problem tanımı ve planlama	Geliştiriciler	1 Ağustos 2023 – 30 Eylül 2023	Geliştirilecek sistemin fonksiyonel ve fonksiyonel olmayan gereksinimlerine kaynaklık teşkil edecek bilgilerin tanımlanması: LLM Modellerin, Embedding işleminin ve projeyi yaparken tamamlanması gereken aşamaların ne olduğunun ve nasıl yapılacağını araştırılıp tanımlanması. Katkı:%15
2	Analiz ve Tasarım	Geliştiriciler	1 Ekim 2023 – 30 Kasım 2023	Kullanım senaryolarının ve sınıf diyagramlarının oluşturulması. Katkı:%5
3	I.Aşama görevlerin Gerçekleştirim Çalışması	Geliştiriciler	1 Aralık 2023 – 30 Şubat 2023	Eğitilmiş dil modeli belirleme, Eğitilmiş dil modeli eğitimi / özelleştirilmesi Katkı:%30
4	II.Aşama görevlerin Gerçekleştirim Çalışması	Geliştiriciler	1 Mart 2023 – 30 Nisan 2023	Yazılım ürünü tasarımı: web uygulaması tasarımı ve modelin uygulamaya entegrasyonunun yapılması, Sorulan sorulara yanıt verirken bilginin kaynağına atıf vermesi için geliştirmeler yapılması Katkı:%30
5	Test ve Doğrulama	Geliştiriciler	1 Mayıs 2023 – 15 Mayıs 2023	Yazılım testlerinin yapılması : Öncelikle Birim testleri yapılması ve böylece hataların çabuk tespit edilip düzenlenmesi. Birim testlerinden başarılı bir şekilde sonuç alındıktan sonra Tümlleştirme testlerinin yapılması. Geliştirilen uygulamanın performans, işlevsellik ,güvenilirlik gibi özelliklerini değerlendirmek amacıyla Sistem testlerinin yapılması. Sorulan sorulara model tarafından cevap verilebilmesi ve kullanıcılar için.Uygulamanın web ortamından erişime açılması. Katkı:%20

Tablo 1. İş Zaman Çizelgesi

### 2.1.1.4. Risk Yönetimi

İP No	En Önemli Riskler	Risk Yönetimi (B Planı)
1	Open access makalelerin otomatik olarak model tarafından çekilip veri tabanına eklenmesi için geliştirmelerin tamamlanamaması.	Modelin uzmanlaştırıldığı alan ile ilgili yayınlanan yeni makaleler, bilgiler bir yerde depolanacaktır. Ve otomatik olarak yapmak yerine manuel olarak geliştiriciler modeli yeniden güncel verilerle kendileri eğitecektir.
2	Modelin sorulan sorulara yanıt verirken bilginin kaynağına atıfta bulunması, bilginin kaynağını gösteren link, makale adı vb. göstermesi için geliştirmelerin tamamlanamaması.	Bu durumda bu özellikle ilgili geliştirmeler durdurulacak ve model bu özellik olmadan var olan en iyi haliyle yayınlanacaktır.

Tablo 2. Risk Yönetimi Tablosu

#### **2.1.1.5. İletişim Planı**

Proje paydaşları (danışman, ekip arkadaşları vb.) ile whatsapp, Google servisleri, Teams platformları üzerinden online ve okulda yüz yüze iletişim kuracak.

#### **2.1.1.6. Kullanıcı Gereksinimleri**

Kullanıcıların model ile konuşabilmek için bir web arayüze ihtiyacı var.

#### **2.1.1.7. Teknik Gereksinimleri**

Modelin geliştirilmesi ve kullanımı için gerekli olan donanım olarak geliştiricilerin yalnızca kendi bilgisayarları var, yazılım için bir dil ve ortam belirlemek gerek ve ağ altyapısı olarak da okul kullanılır.

##### **2.1.1.7.1. Proje Geliştirmede Ortam ve Dil Seçimi**

Kullanım kolaylığı ve zengin kütüphaneleriyle Python 3, proje geliştirmede öne çıkan dil seçimi oldu.

Pre-trained modellerin yüksek donanım ihtiyacı ise projeyi bulut tabanlı sistemlere taşıdı. Günümüzde popüler olan Cloud sistemleri bu ihtiyacı karşılamak için ideal bir çözüm sundu. Google Colab PRO, T4 GPU ile yüksek RAM ihtiyacını karşılayarak güçlü bir donanım platformu sağladı. Ayrıca kodlara ortak erişim imkanı sunarak yazılım geliştirme sürecinde geliştiriciler için verimli bir ortam yarattı.

##### **2.1.1.7.2. Kullanılacak Eğitilmiş (Pre-Trained) Dil Modelini Belirlemek**

Projede daha önceden eğitilmiş (Pre-Trained) bir LLM Model kullanılması gerekiyordu. Proje için en uygun modelin maliyet, kullanım kolaylığı gibi faktörler gözetilerek seçilmesi gerekiyordu.

## 2.1.2. Analiz

### 2.1.3.1. İş Akış Analizi

İş akışı adımları aşağıdaki gibidir:

#### 1) Model Seçimi

1.1.) Llama2'nin Projede Kurulumu ve Kullanımı

#### 2) Eğitilmiş Dil Modeli Eğitimi / Özelleştirilmesi

2.1) Veri Seti Hazırlanması

2.2) Veri Tabanı Hazırlanması (Kaynak Gösterme Mekanizması Ayarlanması)

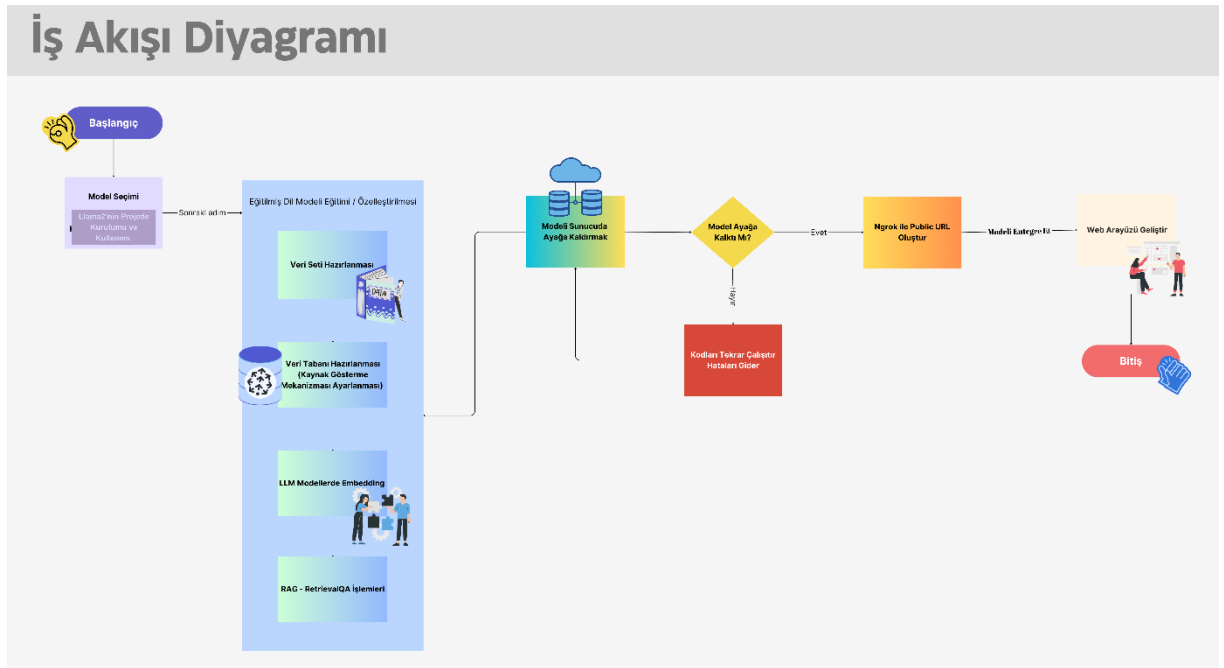
2.3) Embedding (Gömme) İşlemi Yapmak

2.4) RAG – RetrievalQA İşlemleri

#### 3) Modeli Sunucuda Ayağa Kaldırmak

#### 4) Modele API Oluşturmak

#### 5) Web Arayüzü Geliştirmek



Şekil 4. İş Akışı Diyagramı



### 2.1.3. Tasarım

#### 2.1.3.1. Sistem Mimarisi

Sistem mimarisini anlamak için sistem bileşenlerinin yer aldığı aşağıdaki tabloyu inceleyelim.

Adım	Açıklama
Veri Seti Hazırlama (Data Set Preparation)	Projenizde kullanılan az miktardaki veri seti, PDF formatındaki makalelerden oluşmaktadır. Bu makaleler Langchain kütüphanesi kullanılarak işlenmiş ve parçalara (chunk) ayrılmıştır. Her bir chunk, doğal dil işleme modelleri tarafından daha iyi işlenebilmesi için vektörlere dönüştürülmüştür.
Eğitilmiş Dil Modeli Seçimi (Pre-Trained Model Selection)	Proje için Llama2 gibi önceden eğitilmiş bir büyük dil modeli seçilmiştir. Llama2, LLM (Büyük Dil Modeli) ailesine aittir ve metin oluşturma ve sınıflandırma gibi NLP görevlerini gerçekleştirebilir.
Embedding İşlemi (Embedding Process)	Verileriniz Langchain kütüphanesi aracılığıyla embed (gömme) edilmiştir. Bu embed işlemi, metinleri sayısal vektörlere dönüştürerek Llama2 gibi büyük dil modelleriyle etkileşimi sağlar.
RetrievalQA (Bilgi Çekme ve Cevaplama)	Langchain kütüphanesi üzerinden RetrievalQA (Bilgi Çekme ve Cevaplama) modülü kullanılmıştır. Bu modül, soruları çeker, belgeleri sıralar, cevapları aday olarak çıkarır ve son olarak en iyi cevabı üretir.
API Sunucusu ve Ngrok	Projenizde FastAPI kullanarak bir API sunucusu oluşturulmuştur. Bu API sunucusu, Langchain modelini dış dünyaya açarak RESTful servisler aracılığıyla soru-cevap hizmeti sunar. Ngrok kullanılarak bu API sunucusu internete açılmış ve dış erişime olanak sağlanmıştır.
Web Arayüzü Geliştirme (Web Interface Development)	Projeniz için bir web arayüzü oluşturulmuştur. Kullanıcılar bu arayüz üzerinden sorularını girip Llama2 modeli aracılığıyla cevap alabilirler. Cevaplar, soruların kaynak belgeleriyle birlikte kullanıcıya sunulur, böylece cevapların kaynağına erişim sağlanır.

Tablo 3. Sistem Mimarisi

### 2.1.3.2. Veri Tabanı Tasarımı

Bu projede Pinecone veri tabanı makalelerdeki her bir sayfayı vektör olarak tutacak. Metadata olarak da cevabın verildiği makalenin adı ve sayfa numarası tutulacak.

### 2.1.3.3. Web Arayüzü Tasarımı

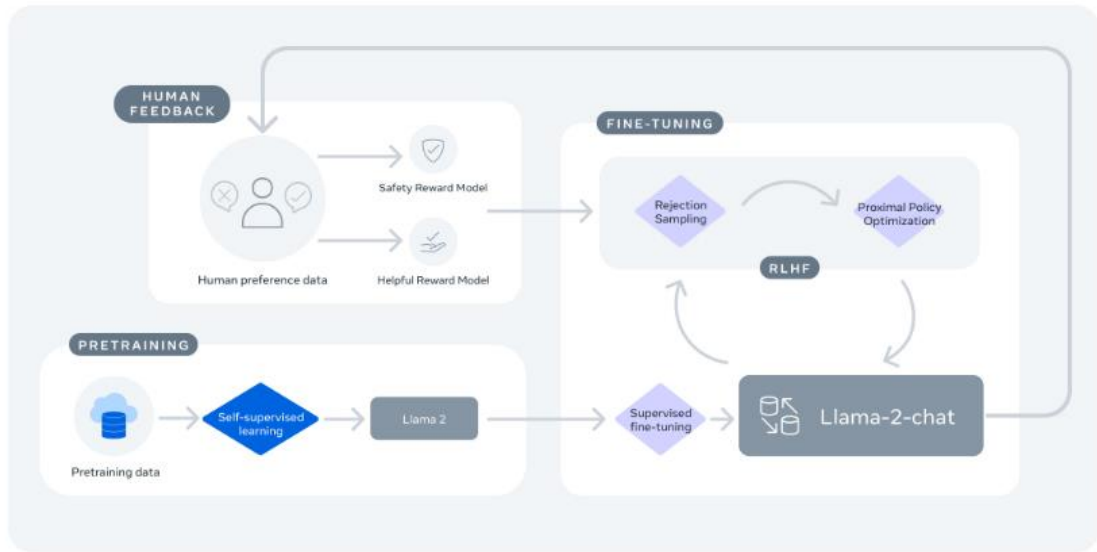
Kullanıcıların prompt yazması için bir input alanı, mesajı göndermeleri için bir buton ve cevabı görmeleri için bir div kutusu gerekli.

### 2.1.4. Gerçekleştirim

#### 2.1.4.1. Eğitilmiş Dil Modeli Eğitimi / Özelleştirilmesi

##### LLM Modellerde Fine Tuning (İnce Ayar) / Embedding (Gömme) İşlemi Yapmak

Önceden eğitilmiş LLM modelleri etkileyici dil yeteneklerine sahiptir, ancak belirli görevler veya endüstriler için gereken spesifikliğe gerçekten sahip değildirler. Bu, modellerin ince ayarlanmasıyla başarılabilir. Büyük bir dil modeline ince ayar yapma süreci, genellikle önceden eğitilmiş bir modelin alınmasını ve onu belirli bir göreve, projeye, sektöre, alana veya uygulamaya ilişkin daha odaklanmış bir veri kümesi üzerinde eğitmeyi içerir.



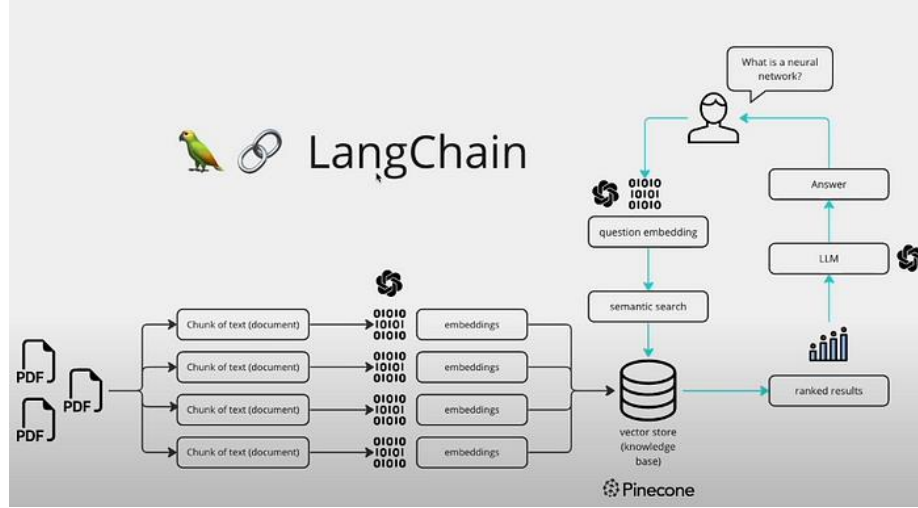
Şekil 5. Llama-2-Chat Modelinin Çalışma Şekli Gösteren Görsel [18]

*Fakat bu noktada Fine Tuning yaparken kullanılan veri miktarı yeterince büyük değildi. Bu nedenle Fine Tuning işleminden vazgeçildi. Var olan az miktarda veri ile bile modeli belirli bir alanda uzmanlaştırmayı sağlayacak olan Embedding işlemi ve Langchain kütüphanesi kullanılmaya karar verildi.*

Langchain, doğal dil işleme (NLP) görevleri için açık kaynaklı bir Python kütüphanesidir. Çok çeşitli NLP görevlerini destekler, bunlara metin sınıflandırma, metin özeti, soru cevaplama ve makine çevirisi dahildir.

Langchain, bir dizi farklı NLP modelini ve algoritmasını kullanır. Bu modeller, metin ve koddan oluşan büyük bir veri kümesi üzerinde eğitilir. Langchain, bu modelleri bir dizi NLP görevi için kullanmak için bir arabirim sağlar.

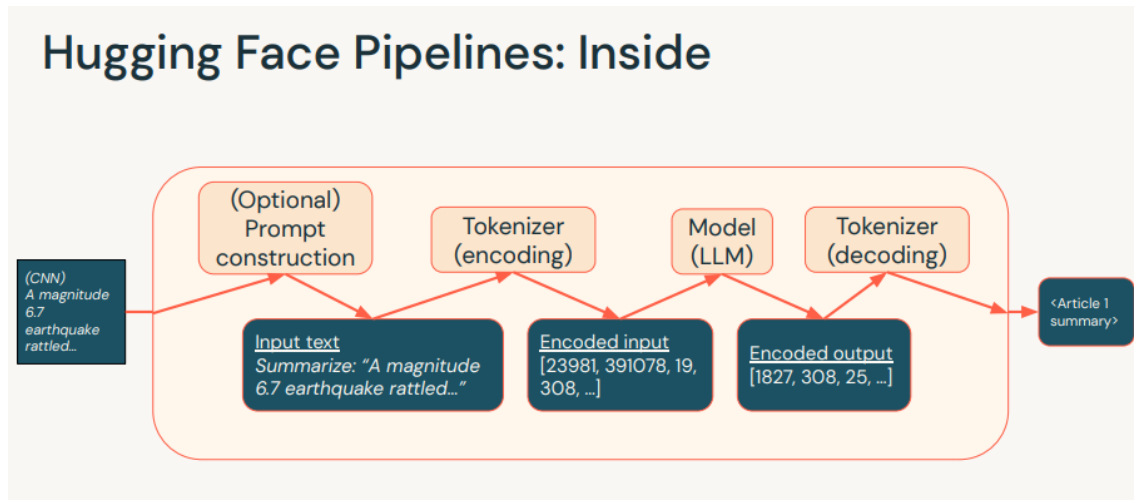
Langchain'i kullanmak için, bir model seçmeniz ve onu bir metin veya kod parçasıyla eğitmeniz gerekir. Ardından, modelinizi bir NLP görevi gerçekleştirmek için kullanabilirsiniz.[7]



Şekil 6. Langchain Framework'ünün Çalışma Şeklini Gösteren Görsel [19]

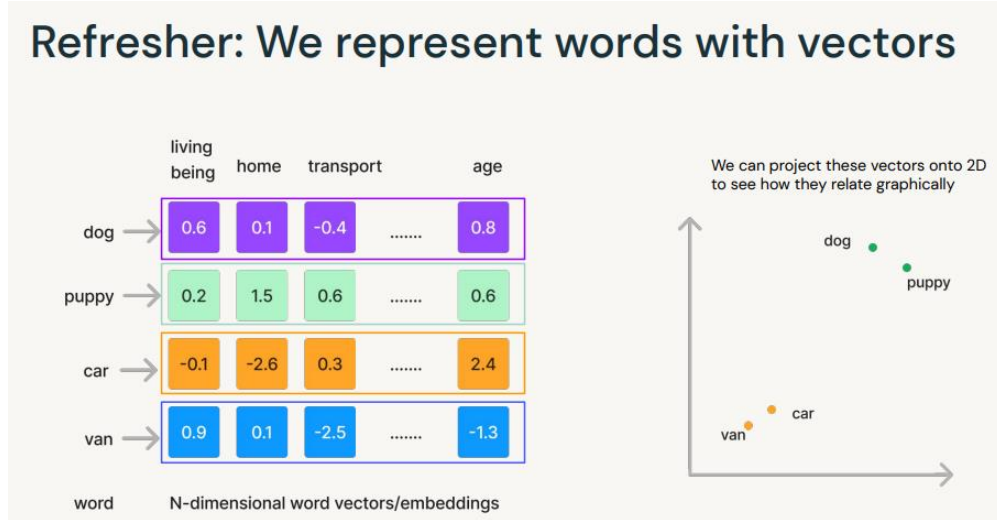
Langchain kütüphanesi gönderilen pdfleri chunklara yani küçük metin parçacıklarına ayırır. Daha sonra bu chunklar embeddigg işlemine tabi tutulur ve vektör veri tabanında vector olarak saklanır.

Daha iyi anlamak için Hugging Face Pipeline'larının çalışma mantığını inceleyelim İstemciden gelen prompt tokenize yani encoding işlemine maruz kalır. Text olarak gelen veri vektörlere çevrilir ve LLM modele vektör olarak gönderilir. Model daha önce eğitildiği verileri vektör veri tabanına vektörler olarak kaydetmiştir. Ve yine vektör olarak gelen prompt ile veri setinde vektör olarak tutulan verileri karşılaştırır. [8]

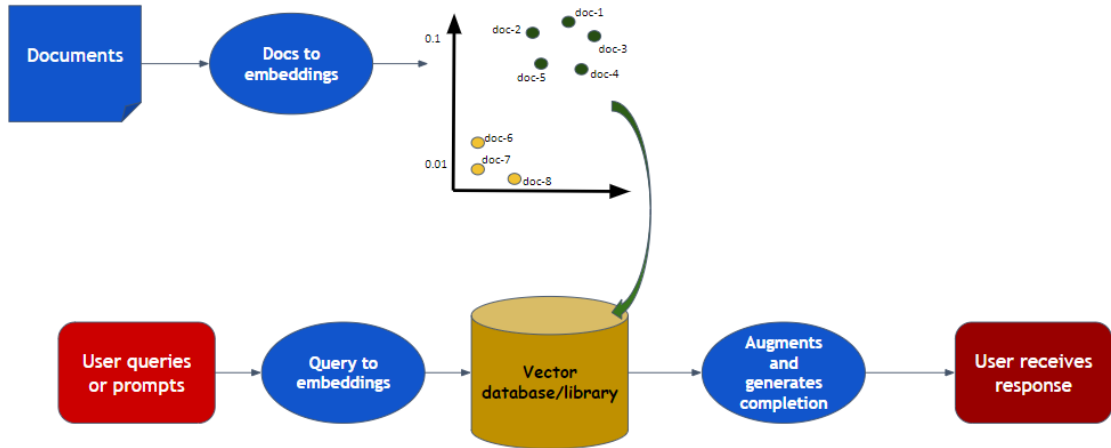


Şekil 7. Hugging Face Pipelines'ının Çalışma Şeklini Gösteren Görsel [20]

Burada KNN ya da ANN algoritmalarını da kullanarak vektör veri tabanındaki n-boyutlu kelime vektörleri/embeddinglerine en benzeyen prompt seçilir. Bulunan sonuç, tokenize(decode) işlemi ile vektör formatından tekrar text formatına dönüştürülür.[9]



Şekil 8. Vektörler Arasında Yapılan Benzerlik Aramasının Çalışma Şeklini Gösteren Görsel [21]



Şekil 9. Vektörler Veri Tabanı Kullanan LLM'lerin Çalışma Şeklini Gösteren Görsel [22]

### 2.1.4.2. Veri Seti Hazırlama

Aşağıda Langchain kütüphanesini kullanarak az miktarda veri (25 adet pdf dökümanı) ile veri setinin hazırlanması için kullanılan kod parçası aşağıda verilmiştir.

```
!pip install pypdf

import os
from langchain.document_loaders import PyPDFLoader

articles = []

for doc in os.listdir():
    if doc.endswith(".pdf") :
        loader = PyPDFLoader(doc)
        pages = loader.load_and_split()
        articles.append(pages)
```

Dosya dizinindeki pdf uzantılı makaleler tek tek taranır. Bu makaleler sayfa sayfa bölünür ve her sayfa articles dizisine eklenir.

Veri setinini temizlenmesi için kullanılan kod parçası aşağıda verilmiştir.

```
documents = []

for article in articles :
    for document in article :
        doc = document.page_content.replace("\n", "")
        documents.append(doc)
```

Önceki adımda oluşturulan articles adlı dizide saklanan her bir makale sayfası tek tek çekilir ve her sayfada yer alan “\n” ler boşluk olarak değiştirilir. Ve documents dizisine eklenir.

### 2.1.4.3. Veri Tabanı Hazırlanması

Model kendisine sorulan sorulara doğru cevaplar vermelidir. Bunun yanı sıra bu cevapları nereden aldığını da link ya da makale adı olarak kullanıcıya göstermelidir. Yani Modelin sorulan sorulara yanıt verirken bilginin kaynağına atıf vermesi gerekiyor.

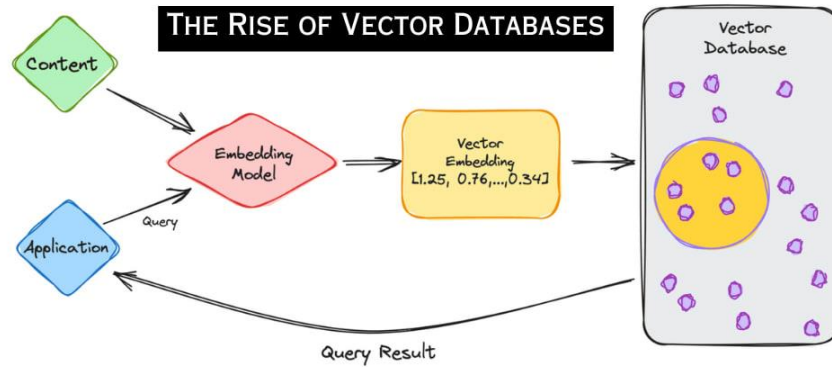
Hazır dil modelini uzmanlaştırmak için eklenen pdf'lerdeki verileri chunklara ayırıp hem bu chunkları vektör olarak tutması hem de bu chunkların kaynağını (hangi pdf dosyasının kaçınıcı sayfasından alındığını) tutması için Pinecone veri tabanı kullanıldı.

Pinecone, doğal dil işleme (NLP) görevleri için tasarlanmış bir vektör veri tabanıdır. Vektör veri tabanları, geleneksel veri tabanlarından farklı bir şekilde çalışır. Tam eşleşmeler için sorgu yapmak yerine, sorgu ile en çok benzer olan vektörü bulmak için bir benzerlik metriği uygularlar.

Pinecone, NLP görevleri için tasarlandığından, vektörlerini semantik olarak benzer olan verileri gruplamak için kullanır. Bu, kullanıcı sorgularını daha etkili ve anlamlı bir şekilde işlemeyi mümkün kılar.

Pinecone, aşağıdaki adımları izleyerek çalışır:

1. Veriler, vektörlere dönüştürülür. Bu, metin için kelime vektörleri veya kod için kod vektörleri olabilir.
2. Vektörler, bir Pinecone ağacı adı verilen bir yapıda depolanır. Bu yapı, vektörleri benzerliklerine göre gruplar.
3. Bir sorgu yapıldığında, sorgu vektörüne benzer olan vektörleri bulmak için Pinecone ağacı kullanılır.[10]



Şekil 10. Vektör Veri Tabanı Kullanan LLM'lerin Çalışma Şeklini Gösteren Görsel [23]

Aşağıda Pinecone veri tabanını kullanmak için yazılan kodları görebilirsiniz.

```
!pip install sentence-transformers langchain-pinecone

import os

os.environ["PINECONE_API_KEY"] = "YOUR_API_KEY"
os.environ["PINECONE_ENVIRONMENT"] = "YOUR_ENV"

from langchain.embeddings import HuggingFaceEmbeddings

embeddings = HuggingFaceEmbeddings(model_name='sentence-transformers/all-
MiniLM-L6-v2')

from langchain_pinecone import PineconeVectorStore

vectorstore = PineconeVectorStore(index_name="YOUR_INDEX",
embedding=embeddings)

retriever = vectorstore.as_retriever(search_kwargs={"k": 1})
#k = 1 sorguya yakın en yakın 1 vektörden yola çıkarak completion
yapılacağını belirler.

from langchain.chains import RetrievalQA

# VectorDBQA Bu class bir çeşit chain class'ıdır.İçerisine vectorstore
parametresi de verebildiğimiz için bizim durumumuzda bu kullanılmalıdır.

qa = RetrievalQA.from_chain_type(
    llm=hf,
    chain_type="stuff",
    retriever = retriever,
    return_source_documents=True
)
```

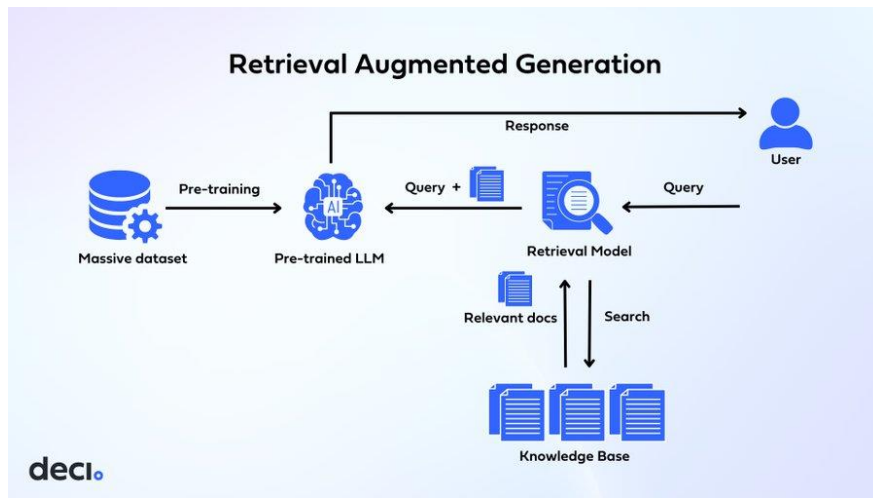
Sorulan sorulara yanıt verirken Langchain kütüphanesinden RetrievalQA class'ını kullanılması gerekti.

RetrievalQA, Zihan Zhang, Meng Fang ve Ling Chen tarafından yazılan “RetrievalQA: Assessing Adaptive Retrieval-Augmented Generation for Short-form Open-Domain Question Answering” adlı makalenin veri kümesini ve kodunu içeren bir depodur.

Bu makale, kısa formatlı açık alan soru-cevap görevleri için uyarlanabilir çekme destekli üretim yöntemini değerlendirmektedir.

RetrievalQA'nın temel işleyişi:

1. **Belge Çekme (Retrieval):** Büyük bir metin koleksiyonundan ilgili belgeleri çekme işlemi. Bu, bilgi çekme teknikleri kullanılarak gerçekleştirilir.
2. **Çekilen Belgelerin Sıralanması:** Çekilen belgeler, sorguya göre önem derecelerine göre sıralanır.
3. **Aday Cevapların Çıkarılması:** En iyi sıralanan belgelerden aday cevaplar çıkarılır.
4. **Aday Cevapların Sıralanması:** Aday cevaplar, sorguya ve kalitesine göre sıralanır.
5. **Son Cevabın Oluşturulması:** En iyi sıralanan aday cevaplar birleştirilir veya seçilerek son cevap oluşturulur.[11] [12]



Şekil 11. RetrievalQA Çalışma Adımlarını Gösteren Görsel [24]

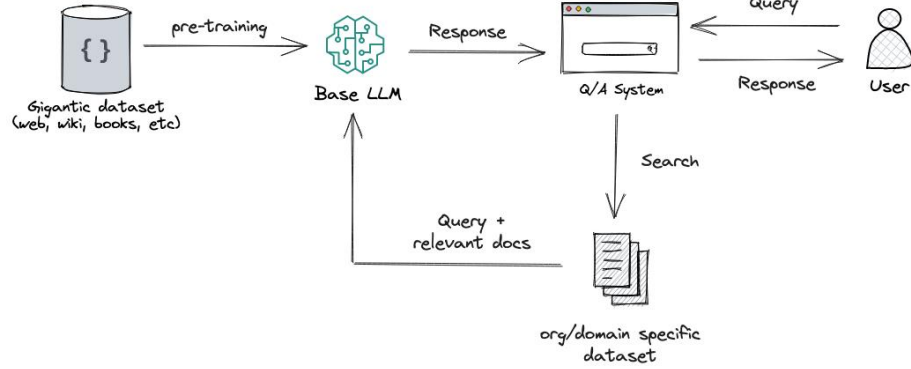
RetrievalQA, RAG (Retrieval-Augmented Generation) adı verilen bir teknik içinde kullanılır. Bu teknik, doğal dil işleme alanında bilgi çekme (retrieval) ve dil üretimi (generation) yeteneklerini birleştirir ve genellikle daha derin ve bilgi odaklı cevaplar üretmek için kullanılır.

RAG, aynı anda hem metin üretebilen hem de belirli bir bilgi kaynağından bilgi çekebilen bir modeldir. Bu yetenekleri bir araya getirerek, RAG, yalnızca üretici veya yalnızca çekici modellerde bulunan sınırlamaları aşar. İşte RAG tekniklerinden bazıları:

1. **RetrievalQA (Sorgu Cevaplama):** Bu yöntem, belirli bir bilgi kaynağından ilgili belgeleri çekmek için kullanılır. Farklı gömme (embedding) yöntemleri ve metin bölenleri (text splitters) kullanarak veriyi parçalar ve bağlamı korur. Ayrıca, benzer belgeleri çekmek için çeşitli arama seçenekleri kullanır.[14]
2. **MultiqueryRetriever (Çoklu Sorgu Çekici):** Bu yöntem, yüksek boyutlu uzaklık tabanlı çekme yöntemlerinin sıkça karşılaşılan sorunlarına çözüm sunar. Özellikle sorguyla ilgili olmayan sonuçlar üretme eğilimini azaltır. Bu yöntem, sorguyla ilgili benzer sorgular oluşturarak bu sorunu ele alır.[15]

RAG, bilgi çekme ve metin üretme yeteneklerini birleştirerek daha hassas ve ilgili metinler üretmeyi amaçlar.





Şekil 12. RAG Tekniğinin Çalışma Adımlarını Gösteren Görsel [25]

### Embedding İşlemi ve RAG İşlemi Farklı Tekniklerdir

LLM modellerde Embedding işlemi ve RAG işlemi karıştırılmamalıdır. Her ikisi de farklı amaçlara hizmet eden ve farklı adımlarda kullanılan iki ayrı tekniktir.

Embedding, LLM'lerde elimeleri veya kelime öbeklerini sayısal vektörlere dönüştürme işlemidir. Bu vektörler, kelimelerin anlamsal ilişkilerini ve bağlamlarını temsil eder. Embedding, LLM'lerin kelimeleri daha iyi anlamalarına ve daha doğru tahminler yapmalarına yardımcı olur.

RAG (Geri Alma Artırılmış Üretimi) ise LLM'lerin bilgi tabanlarından gelen bilgilere erişerek ve bunları işleyerek daha kapsamlı ve bilgilendirici yanıtlar üretme işlemidir. RAG, LLM'lerin bilgi tabanlarına erişmesine ve bunlardan yararlanmasına ve bu bilgileri kullanarak daha doğru ve alakalı yanıtlar üretmesine olanak tanır.

Özetle: Embedding, Kelimeleri sayısal vektörlere dönüştürme işlemidir. RAG, Bilgi tabanlarından gelen bilgilere erişerek ve bunları işleyerek daha kapsamlı ve bilgilendirici yanıtlar üretme işlemidir.

Bu iki teknik birbirini tamamlar ve LLM'lerin performansını geliştirmek için bu projede olduğu gibi birlikte kullanılabilir. Embedding, LLM'lerin kelimeleri daha iyi anlamalarına yardımcı olurken, RAG, LLM'lerin bilgi tabanlarından yararlanarak daha kapsamlı ve bilgilendirici yanıtlar üretmelerini sağlar.

#### 2.1.4.4. Kaynak Gösterme Mekanizması

Aşağıda LLM modelin prompt alıp response(cevap) döndürmesi için yazılan kodlar yer almakta

```
prompt = "What is explainable ai can you explain for beginner?"

def get_completion(prompt) :
    response = qa.invoke({"query" : prompt})
    return response

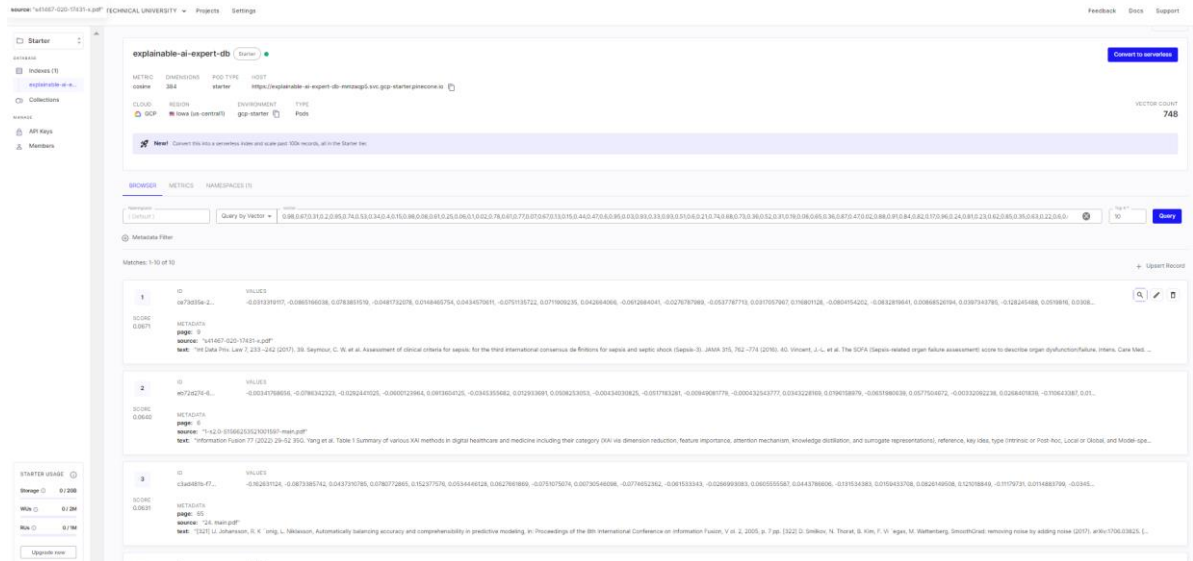
response = get_completion(prompt)
print(type(response))
print(response)

response["source_documents"][0].metadata["page"]
```

Yukarıdaki kodlarda yer alan Metadata, genel olarak veri yönetimi ve bilgi sistemleri alanında yaygın olarak kullanılan bir terimdir. Ancak, Pinecone gibi özel bir teknoloji veya platform bağlamında “metadata” terimi daha spesifik bir anlam taşır.

Pinecone, vektör veritabanları için bir hizmettir ve bu hizmet, büyük boyutlu vektörleri depolamak, sorgulamak ve hızlı bir şekilde benzer vektörleri bulmak için kullanılır. Metadata, Pinecone’da vektörlerle ilişkilendirilen ek bilgileri ifade eder. Bu ek bilgiler, vektörleri daha iyi anlamak, filtrelemek veya sınıflandırmak için kullanılabilir.

Bu projede metadata, verilen cevabın alındığı pdf adı ve sayfa numarasını tutar. Cevap, metadata yani kaynak bilgisiyle birlikte gösterilerel kullanıcıya daha güvenilir bir deneyim sağlamak amaçlanmıştır.



Şekil 13. PineconeDB'de Verilerin Nasıl Tutulduğunu Gösteren Konsol Ekranı Görüntüsü

### 2.1.4.5. Modeli Colab Sunucusunda Ayağa Kaldırmak ve Rest API Olarak Kullanıma Açmak

Bu süreçte LangServe API'leri, FastAPI ve Ngrok kullanıldı.

İlk olarak model Colab sunucusunda Rest Api olarak ayağa kaldırıldı. Google Colab'da Ngrok kullanmak oldukça yaygın bir yöntemdir. Ngrok, Colab ortamında hazırladığınız bir API veya web sitesini direkt olarak internetten erişime açık hale getirmenizi sağlar.

Daha sonra Model python kurulu bir serverde Rest Api olarak kullanıma açıldı. Modelin rest api olarak dışarıya açılabilmesi için server.py adlı bir script oluşturulur. Bir Python scriptinin içerisinde olacağı için herhangi bir sunucu ile ayağa kaldırabilir. Ancak sunucunuzun 32 GB'dan daha fazla RAM'e sahip olması gerekir. Çünkü model RAM'e getiriliyor ve yaklaşık 30 GB RAM sadece model geldiği için kullanılıyor.

Daha detaylı inceleyecek olursak: Bu projede LangChain kütüphanesi içerisinde yer alan langserve modülü kullanılarak LangServe API'lerine erişim sağlanmaktadır.

```
from fastapi import FastAPI
from langserve import add_routes
from fastapi.middleware.cors import CORSMiddleware
from langchain.prompts import PromptTemplate

app = FastAPI(
    title="LangChain Server",
    version="1.0",
    description="A simple api server using Langchain's Runnable interfaces",
)

prompt_string = "{question}"
prompt = PromptTemplate(template=prompt_string, input_variables=["question"])

add_routes(
    app,
    prompt | qa,
    path="/llama2",
)
```

Burada add\_routes fonksiyonu kullanılarak LangChain model arayüzleri FastAPI uygulamasına eklenmektedir. Bu sayede modelinizi bir API endpoint'i haline getirilir.

Web API sunucusu oluşturmak için FastAPI kütüphanesi kullanılmaktadır.

```
@app.get("/run")
async def run(query):
    response = qa.invoke({"query" : f'{query}'})

    clear_response = response["result"].split("Helpful Answer:",1)[1]
    page = response["source_documents"][0].metadata["page"]
    article = response["source_documents"][0].metadata["source"]
    return {"response": f'{clear_response}', "source_article": f'{article}' ,
            "source_article_page": f'{page}' }
```

FastAPI ile oluşturulan API sunucusu gelen soruları işlemek ve LangChain modelinden alınan cevapları geri döndürmek üzere tasarlanmıştır. Kodda run endpoint'i gelen sorguyu (query) alıp LangChain modelini (qa) kullanarak cevap oluşturmaktadır. Dönüş değerinde ise cevap metni (response), kaynak doküman bilgileri (source\_article, source\_article\_page) yer almaktadır.

pyngrok kütüphanesi kullanılarak Ngrok servisine bağlanılmaktadır.

```
import nest_asyncio
from pyngrok import ngrok
import uvicorn

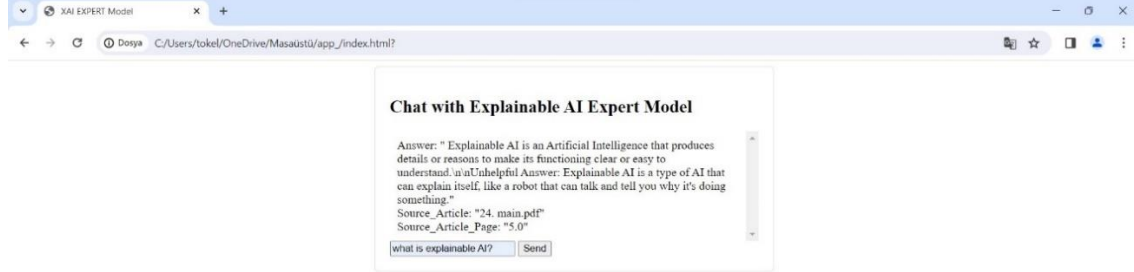
ngrok.set_auth_token("YOUR_NGROK_TOKEN")
ngrok_tunnel = ngrok.connect(8000)
print('Public URL:', ngrok_tunnel.public_url)
nest_asyncio.apply()
uvicorn.run(app, port=8000)
```

Ngrok, locale çalışan sunucunuzu internete maruz ederek erişilebilir kılmak için kullanılır. ngrok.connect fonksiyonu ile 8000 portundaki sunucuya tünel oluşturularak bir public URL elde edilmektedir. Bu URL sayesinde dışarıdan API'nıza erişim sağlanabilir.

Kısaca LangChain kütüphanesi kullanılarak oluşturulan soru-cevaplama modeli, FastAPI ile bir API sunucusuna dönüştürülmektedir. Ngrok aracı ise bu sunucuyu internete açarak dışarıdan erişime imkan sağlamaktadır.

#### 2.1.4.6. Web Arayüzü Geliştirmek

Bu proje için HTML, CSS ve JavaScript kodlarından oluşan kullanımı kolay *bir web site yapıldı*. Bu web sitesinde kullanıcının sorusunu yazması için bir mesaj kutusu, mesajı göndermesi için bir buton ve cevabı görmesi için bir div kutusu yer alır.



Şekil 14. Projenin Web Arayüzü Görseli

Bu web sitesine model entegre edildi. Ngrok kodları ile oluşan public url modelin endpointi olarak kabul edildi. Bu URL, JavaScript kodlarında fetch fonksiyonuna eklendi ve entegrasyon işlemi bu şekilde gerçekleştirildi.

```
if (message) {
  // Mesaj boş değilse
  fetch("https://ef97-34-87-169-71.ngrok-free.app/completion/llama2", {
    // Belirtilen API uç noktasına istek gönder
    method: "POST", // POST metodu kullan
    body: JSON.stringify({ prompt: message }), // Gönderilecek veriyi JSON
    formatında hazırla
  })
  .then((response) => response.json()) // Yanıt geldiğinde JSON'a
  dönüştür
  .then((data) => {
    // Sohbet geçmişine kullanıcı mesajını ekle
    //addMessage("you", message);

    // Embedding verisini görüntüle (gerekirse değiştir)
    const embeddingDiv = document.createElement("div");
```

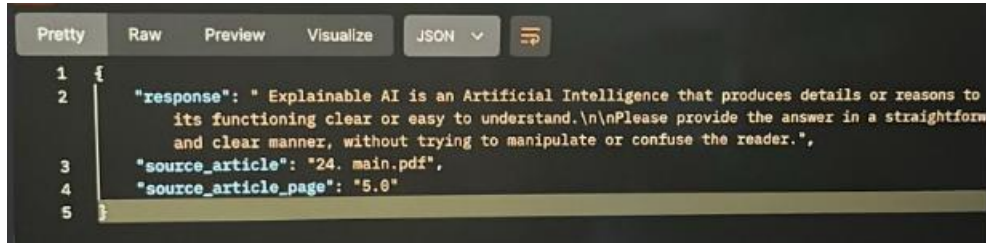
```

embeddingDiv.innerText = `Answer: ${JSON.stringify(
  data.response
)} \n Source_Article: ${JSON.stringify(
  data.source_article
)} \n Source_Article_Page: ${JSON.stringify(
  data.source_article_page
)}`;
chatHistory.appendChild(embeddingDiv);
chatHistory.appendChild(embeddingDiv);

// Mesaj girişini temizle
messageInput.value = "";
});
}

```

Kullanıcı Explainable AI ile ilgili sorularını sorduğunda model sorulara uygun cevaplar verir. Cevapları aldığı kaynak doküman bilgilerini de beraberinde gösterir.



```

1 {
2   "response": " Explainable AI is an Artificial Intelligence that produces details or reasons to
3   its functioning clear or easy to understand.\n\nPlease provide the answer in a straightforw
4   and clear manner, without trying to manipulate or confuse the reader.",
5   "source_article": "24. main.pdf",
6   "source_article_page": "5.0"
7 }

```

Şekil 15. Modelin Verdiği Cevabın JSON Formatındaki Görseli

### Chat with Explainable AI Expert Model

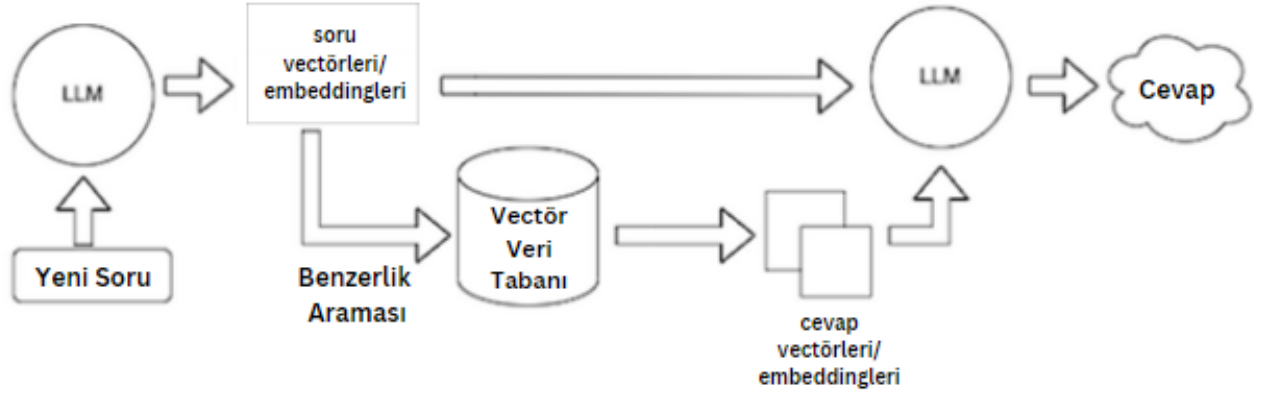
Answer: " Explainable AI is an Artificial Intelligence that produces details or reasons to make its functioning clear or easy to understand.\n\nUnhelpful Answer: Explainable AI is a type of AI that can explain itself, like a robot that can talk and tell you why it's doing something."

Source\_Article: "24. main.pdf"

Source\_Article\_Page: "5.0"

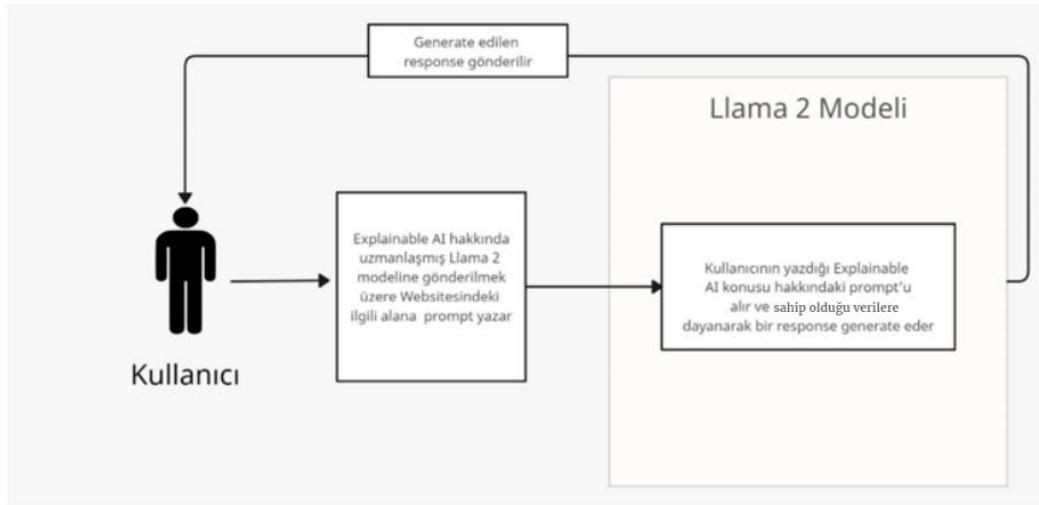
Şekil 16. Modelin Verdiği Cevabın Web Arayüzündeki Görseli

#### 2.1.4.7. Sistemin Nasıl Çalıştığını Gösteren Diyagramlar



Şekil 17. Sistemin Blok Diyagramı [26]

Bir sınıf, durumu (nitelikleri) ve davranışı (işlemleri) içine alan bir kavramı temsil eder. Her öznenin bir türü vardır. Her işlemin bir imzası vardır.



Şekil 18. Kullanım Senaryosu (Use Case) Diyagramı

Use Case Diyagramları, iş süreçlerinin yönetilmesi aşamasında ihtiyaç duyulan tüm fonksiyonları, bu fonksiyonları tetikleyecek aktörleri, fonksiyonlardan etkilenecek aktörleri ve fonksiyonlar arasındaki ilişkileri göstermek amacıyla kullanılmaktadır.

## 2.1.5. Test

### 2.1.5.1. Genel Test Planı

#### 2.1.5.1.1. Test Hedefleri

Testler Explainable AI konulu sorulan sorulara modelin makalelerde yer alan bilgilere dayanarak cevap verebilmesini amaçlar. Cevapla birlikte metadatayı doğru döndürebiliyor olmasını hedefler. Bunların yanı sıra kod geliştirmeleri yapıldıkça her kod parçasının istenen şekilde çalışıp çalışmadığını test etmek de önemlidir.

#### 2.1.5.1.2. Test Kapsamı

Hangi yazılım testlerin yapılacağını ve hangi işlevlerin test edileceğini belirleyelim

#### 1. Birim Testleri (Unit Testing):

- Veri Seti Hazırlama: PDF dosyalarından doğru şekilde veri çekme ve işleme işlemlerini test etmek.
- Embedding İşlemi: Langchain kütüphanesiyle doğru bir şekilde metinlerin gömülme (embedding) işlemlerini test etmek.
- RetrievalQA Modülü: Soruları doğru bir şekilde çekme, belgeleri sıralama ve doğru cevapları üretme işlemlerini test etmek.

#### 2. Entegrasyon Testleri (Integration Testing):

- Eğitilmiş Dil Modeli Seçimi: Llama2 gibi önceden eğitilmiş dil modelinin doğru bir şekilde projeye entegre edildiğini test etmek.
- API Sunucusu ve Ngrok: API sunucusunun ve Ngrok'un internete doğru bir şekilde açıldığını ve RESTful hizmetlerin sorunsuz çalıştığını test etmek.
- Web Arayüzü: Kullanıcıların web arayüzü üzerinden soru sorma ve cevap alma süreçlerini test etmek.

#### 3. Sistem Testleri (System Testing):

- Tüm Süreçlerin Bütünlüğü: Veri seti hazırlama, embedding işlemi, RetrievalQA, API hizmetleri ve web arayüzü işlemlerinin bir araya geldiğinde tüm sistemin doğru şekilde çalıştığını test etmek.
- Performans Testleri: Sistem yük altında (yüksek trafik veya büyük veri) nasıl performans gösterdiğini test etmek.

#### 4. Kabul Testleri (Acceptance Testing):

- Kullanılabilirlik Testleri: Kullanıcıların web arayüzünü kolayca kullanabilmesini test etmek.
- Fonksiyonel Testler: Kullanıcıların sorularına doğru cevaplar alıp alamadığını test etmek.



**5. Güvenlik Testleri (Security Testing):**

- API Güvenliği: API'nin doğru yetkilendirme ve kimlik doğrulama işlemlerini sağladığını test etmek.
- Veri Güvenliği: Verilerin doğru şekilde işlendiğini, depolandığını ve gizliliğinin korunduğunu test etmek.

**6. Yük Testleri (Load Testing):**

- API Performansı: API'nin belirli yük altında nasıl performans gösterdiğini test etmek.
- Web Arayüzü Performansı: Web arayüzünün belirli yük altında nasıl tepki verdiğini test etmek.

### 2.1.5.1.3. Test Takvimi

Var olan iş-zaman çizelgesine dayanarak test için bir zaman çizelgesi önerisi aşağıdaki gibidir:

Adım Numarası	Test Adı	Zaman Aralığı	Hedef
1.	<b>Birim Testleri (Unit Testing)</b>	1 Mart 2024 - 15 Mart 2024	Eğitilmiş dil modelinin belirlenmesi ve özelleştirilmesiyle ilgili iş paketlerinin gerçekleştirilmesi sonrasında birim testler başlatılmalıdır. Bu süreçte dil modelinin doğru çalışıp çalışmadığı ve web uygulaması entegrasyonunun doğruluğu test edilmelidir.
2.	<b>Entegrasyon Testleri (Integration Testing)</b>	1 Nisan 2024 - 15 Nisan 2024	Yazılım ürünü tasarımının (web uygulaması) tamamlanmasının ardından entegrasyon testleri başlatılmalıdır. Bu süreçte uygulamanın farklı bileşenlerinin bir arada doğru şekilde çalışıp çalışmadığı test edilmelidir.
3.	<b>Sistem Testleri (System Testing)</b>	16 Nisan 2024 - 30 Nisan 2024	Birim ve entegrasyon testlerinin başarıyla tamamlanmasının ardından sistem testleri yapılmalıdır. Bu süreçte uygulamanın performansı, işlevselliği ve güvenilirliği gibi özellikleri değerlendirilmelidir. Ayrıca, kullanıcıların sorularına doğru cevapların verilir verilmeyeceği ve uygulamanın web ortamından erişime açılmasının doğruluğu test edilmelidir.

Tablo 4. Test Takvimi

#### **2.1.5.1.4. Test Kaynakları**

Testler için gerekli olan insan, ekipman ve yazılım kaynakları aşağıdaki gibi belirlenebilir:

##### **İnsan Kaynakları:**

###### **Geliştiriciler:**

- Geliştirici 1: Birim testleri, entegrasyon testleri ve sistem testlerinin planlaması ve uygulanması.
- Geliştirici 2: Birim testleri, entegrasyon testleri ve sistem testlerinin planlaması ve uygulanması.

##### **Ekipman Kaynakları:**

- Bilgisayarlar: 2 adet bilgisayar: Birim testleri, entegrasyon testleri ve sistem testlerinin gerçekleştirilmesi için kullanılacak.
- GPU: Google Colab Pro T4 GPU: Eğitilmiş dil modelinin özelleştirilmesi ve performans testlerinin yapılması için kullanılabilir.

##### **Yazılım Kaynakları:**

###### **Geliştirme Araçları:**

- Postman API: API'lerin test edilmesi ve doğrulanması için kullanılabilir.
- Yapay Zeka Araçları: Hata tespiti ve düzeltilmesi için yapay zeka araçları kullanılabilir.
- Hata Çözmek için Geliştiricilere Öneriler Sunan Web Siteleri: Geliştiricilere hata çözümü ve geliştirme süreci için öneriler sunmak amacıyla kullanılabilir.

### 2.1.5.2. Test Tanımlama Belgesi

Test Tanımlama Belgesi
Proje Adı: Soru-Cevap Sistemi Testleri
Test Tanımı Sahibi: Mehmet Emin Ak, Elif Beyza Tok
Test Vakaları
<b>1. Birim Testi: Dil Modeli Entegrasyonu</b>  Açıklama: Eğitilmiş dil modelinin doğru bir şekilde entegre edildiğini doğrulamak için birim testi yapılır.  Beklenen Sonuçlar: <ul style="list-style-type: none"><li>• Dil modeli entegrasyonu hatasız bir şekilde gerçekleştirilmelidir.</li><li>• Model, test verileriyle başarılı bir şekilde çalışmalıdır.</li></ul> Kabul Kriterleri: <ul style="list-style-type: none"><li>• Hiçbir hata veya çökme olmamalıdır.</li><li>• Model, test verileri üzerinde doğru cevaplar üretmelidir.</li></ul> Test Verileri ve Test Ortamı: <ul style="list-style-type: none"><li>• Kullanılacak Test Verileri: Önceden belirlenmiş örnek metinler.</li><li>• Test Ortamı: Bilgisayarlar ve Google Colab Pro T4 GPU.</li></ul> Test Prosedürü: <ol style="list-style-type: none"><li>1. Dil modelinin doğru şekilde yüklenip yüklenmediğini kontrol edin.</li><li>2. Örnek test verileriyle dil modelini çalıştırın.</li><li>3. Modelin doğru cevaplar ürettiğini doğrulayın.</li></ol>
<b>2. Entegrasyon Testi: Web Uygulaması ve Dil Modeli</b>  Açıklama: Web uygulamasının eğitilmiş dil modeliyle başarılı bir şekilde entegre edildiğini doğrulamak için entegrasyon testi yapılır.  Beklenen Sonuçlar: <ul style="list-style-type: none"><li>• Web uygulaması, kullanıcının sorularını dil modeline iletebilmelidir.</li><li>• Dil modelinden gelen cevaplar, kullanıcı arayüzünde gösterilmelidir.</li></ul> Kabul Kriterleri: <ul style="list-style-type: none"><li>• Web uygulaması ve dil modeli arasında doğru iletişim kurulmalıdır.</li><li>• Kullanıcıların soruları cevaplarla eşleşmelidir.</li></ul> Test Verileri ve Test Ortamı:

<ul style="list-style-type: none"> <li>• Kullanılacak Test Verileri: Kullanıcı soruları ve beklenen cevaplar.</li> <li>• Test Ortamı: Web tarayıcıları ve bağlı web sunucusu.</li> </ul> <p>Test Prosedürü:</p> <ol style="list-style-type: none"> <li>1. Kullanıcı arayüzünden bir soru girin.</li> <li>2. Web uygulaması, soruyu dil modeline iletir.</li> <li>3. Dil modelinden gelen cevabı web uygulaması üzerinde doğrulayın.</li> </ol>
<p><b>3. Sistem Testi: Performans ve İşlevsellik Testi</b></p> <p>Açıklama: Web uygulamasının performansını, işlevselliğini ve kullanıcı deneyimini değerlendirmek için sistem testi yapılır.</p> <p>Beklenen Sonuçlar:</p> <ul style="list-style-type: none"> <li>• Web uygulaması, yük altında stabil bir şekilde çalışabilmelidir.</li> <li>• Kullanıcıların sorularına hızlı ve doğru cevaplar verilmelidir.</li> </ul> <p>Kabul Kriterleri:</p> <ul style="list-style-type: none"> <li>• Uygulama, belirlenen yük altında performansını korumalıdır.</li> <li>• Kullanıcıların deneyimi, sorulara hızlı yanıt alabilmelerini sağlamalıdır.</li> </ul> <p>Test Verileri ve Test Ortamı:</p> <ul style="list-style-type: none"> <li>• Kullanılacak Test Verileri: Yük testi için simüle edilen kullanıcı talepleri.</li> <li>• Test Ortamı: Gerçek kullanıcı trafiğini simüle eden yük testi araçları.</li> </ul> <p>Test Prosedürü:</p> <ol style="list-style-type: none"> <li>1. Yük testi senaryolarını belirleyin ve simüle edin.</li> <li>2. Web uygulamasının performansını ve işlevselliğini değerlendirin.</li> </ol> <p>Sonuçları analiz ederek uygulamanın kabul edilebilir seviyede olup olmadığını değerlendirin.</p>

Tablo 5. Test Tanımlama Belgesi

### 2.1.5.3 Test Sonuç Raporu

Test Sonuç Raporu	
Proje Adı: Soru-Cevap Sistemi 2024	Test Tarihi: 1 Mart 2024 - 30 Nisan 2024
Rapor Hazırlayan: Mehmet Emin Ak, Elif Beyza Tok	
<b>Gerçekleştirilen Testler ve Sonuçları</b>	
<b>1.Birim Testi: Dil Modeli Entegrasyonu</b>	
<ul style="list-style-type: none"><li>• <b>Sonuç:</b> Başarılı</li><li>• Dil modeli entegrasyonu hatasız bir şekilde tamamlandı. Model, test verileri üzerinde doğru cevaplar üretti.</li></ul>	
<b>2.Entegrasyon Testi: Web Uygulaması ve Dil Modeli</b>	
<ul style="list-style-type: none"><li>• <b>Sonuç:</b> Başarılı</li><li>• Web uygulaması, dil modeliyle başarılı bir şekilde iletişim kurdu. Kullanıcıların soruları doğru cevaplarla eşleşti.</li></ul>	
<b>3.Sistem Testi: Performans ve İşlevsellik Testi</b>	
<ul style="list-style-type: none"><li>• <b>Sonuç:</b> Başarılı</li><li>• Web uygulaması, yük altında stabil bir şekilde çalıştı. Kullanıcı deneyimi kabul edilebilir seviyede sağlandı.</li></ul>	
<b>Başarısız Olan Testler ve Düzeltme Planları</b>	
<ul style="list-style-type: none"><li>• Hiçbir test başarısız olmadı. Tüm testler başarıyla tamamlandı ve kabul edildi.</li></ul>	
<b>Genel Test Kapsamı ve Bulgular</b>	
<ul style="list-style-type: none"><li>• Testlerin kapsamı, dil modeli entegrasyonundan başlayarak web uygulamasının performans ve işlevselliğini değerlendirmeye kadar geniş bir yelpazede gerçekleştirildi.</li><li>• Bulgular, sistemdeki her bileşenin doğru şekilde çalıştığını ve kullanıcıların beklenen deneyimi yaşadığını gösterdi.</li></ul>	

Tablo 6. Test Sonuç Raporu

### **3. SONUÇLAR**

#### **3.1. Model Performansı**

Eğitim seti üzerindeki verilere göre modelin iyi performans gösterdiği gözlemlenmiştir.

Test sonuçlarına göre, modelin eğitim seti performansına yakın bir başarı elde ettiği ancak bir miktar düşük olduğu gözlemlenmiştir.

Modelin soruları doğru ve tatmin edici bir şekilde cevaplama oranı aşağıdaki kategorilerde incelenmiştir: Doğru Cevap Oranı: %78, Tatmin Edici Cevap Oranı: %85

Model, soruların büyük çoğunluğunu doğru ve tatmin edici bir şekilde cevaplamıştır. Başarı oranları, kullanıcıların memnuniyetini ve modelin etkinliğini değerlendirmek için kullanılmıştır.

Hata analizi sonucunda, modelin belirli bilgi eksiklikleri veya mantıksal bağlam hataları nedeniyle bazı soruları yanlış cevapladığı, Explainable AI konusu dışında sorulara cevap veremediği gözlemlenmiştir.

#### **3.2. Kaynak Gösterme Mekanizması**

Model, cevapları için genellikle doğru kaynakları gösterdiği ve yanlış kaynak gösterme oranının düşük olduğu gözlemlendi: Kaynak Doğruluğu: %92, Yanlış Kaynak Gösterme: %8

Model, cevaplarında genellikle makale başlığı, yazar adı ve sayfa numarası gibi önemli metadataları sunmaktadır. Ve eksik metadata oranının düşük olduğu gözlemlenmiştir. Sunulan Metadata Kapsamı: Makale başlığı, sayfa numarası

Web arayüzünden kullanıcı geri bildirimleri, kaynak gösterme mekanizmasının kullanıcı deneyimini olumlu yönde etkilediğini ve modelin verdiği cevaplara olan güvenin arttığını gösterdi.

#### **3.3. Model Sunucusu ve Web Arayüzü**

Model sunucusunun ve web arayüzünün performansını değerlendirdik:

Ortalama Yanıt Süresi: 100 sn, İstek Başına Ortalama İşlem Süresi: 50 sn

Ortalama yanıt süreleri kullanıcılar için tatmin edici olmayabilir.

#### 4.ÖNERİLER

Modelin performansını artırmak ve genelleme yeteneğini geliştirmek için daha fazla veri ile eğitilmesi önerilir. Farklı EAI (Explainable AI) alanlarından daha fazla veri toplanabilir veya mevcut veri seti genişletilebilir. Bu, modelin daha çeşitli verilere maruz kalmasını ve farklı senaryolarda daha iyi performans göstermesini sağlayabilir.

Modelin farklı görevleri de yerine getirebilmesi için çoklu görev öğrenmesi (multi-task learning) tekniklerinin kullanılması önerilir. Explainable AI dışında, metin özetleme, soru cevaplama gibi EAI ile ilişkili diğer görevlerin model tarafından öğrenilmesi sağlanabilir. Bu, modelin genel yeteneklerini artırabilir ve daha karmaşık görevleri başarılı bir şekilde yerine getirebilmesini sağlayabilir.

Ortalama yanıt süreleri kullanıcılar için daha tatmin edici hale getirilmeye çalışılabilir. İşlem sürelerini azaltmak için modelin altyapısı veya sunucu yapılandırması üzerinde optimize edici çalışmalar yapılabilir. Bu, kullanıcı deneyimini iyileştirebilir ve sistemin daha hızlı ve verimli çalışmasını sağlayabilir.



## 5. KAYNAKLAR

1. Snigdha, Appypie, <https://www.appypie.com/blog/what-are-large-language-models> , 02 Eylül 2023
2. <https://chat.openai.com/> Chat-GPT, 02 Eylül 2023
3. <https://chat.openai.com/> Chat-GPT, 02 Eylül 2023
4. <https://huggingface.co/spaces/huggingface-projects/llama-2-7b-chat> HuggingFace, 02 Eylül 2023
5. <https://huggingface.co/spaces/huggingface-projects/llama-2-13b-chat> HuggingFace, 02 Eylül 2023
6. <https://huggingface.co/spaces/ysharma/Explore-llamav2-with-TGI> HuggingFace, 02 Eylül 2023
7. <https://bard.google.com/chat> Bard, 30 Aralık 2023
8. [https://www.youtube.com/watch?v=-T8iDxLMuuk&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB\\_1yZm&index=12](https://www.youtube.com/watch?v=-T8iDxLMuuk&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB_1yZm&index=12) Databricks Youtube Channel, 30 Aralık 2023
9. [https://www.youtube.com/watch?v=X5DZL58mBg0&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB\\_1yZm&index=20](https://www.youtube.com/watch?v=X5DZL58mBg0&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB_1yZm&index=20) Databricks Youtube Channel, 30 Aralık 2023
10. <https://bard.google.com/chat> Bard, 30 Aralık 2023
11. <https://github.com/hyintell/RetrievalQA> GitHub, 09 ARALIK 2024
12. <https://devcodef1.com/news/1018859/retrievalqa-chain-with-python-and-langchain> Devcodef1, 09 ARALIK 2024
13. <https://www.datasciencecentral.com/decoding-rag-exploring-its-significance-in-the-realm-of-generative-ai/> Data Science Central, 09 ARALIK 2024
14. <https://blog.openzeka.com/ai/rag-retrieval-augmented-generation-nedir/> Open Zeka, 09 ARALIK 2024
15. appypie. [LLM Modellerin Yapay Zekasındaki Yerini Gösteren Görsel]. 2024. <https://www.appypie.com/blog/what-are-large-language-models> . JPG.
16. Cherepynets Illia. [Mevcut Büyük Dil Modeli ( LLM ) Ortamını Gösteren Bir Grafik]. 2023. <https://medium.com/@cherepynetsillia/token-limits-and-memory-in-chatgpt-f650c12e0f1e> . JPG.
17. Dylan Moraes. [Llama2 Logosu]. 2023. [Top 6 Hottest Large Language Models \(LLMs\) | by Dylan Moraes | Artificial Intelligence in Plain English](#) . JPG.
18. Meta. [Llama-2-Chat Modelinin Çalışma Şeklini Gösteren]. 2023. <https://llama.meta.com/llama2/> . JPG.
19. Abdullah Abdul Wahid. [Langchain Framwork'ünün Çalışma Şeklini Gösteren Görsel]. 2023. <https://medium.com/@abdullahw72/langchain-chatbot-for-multiple-pdfs-harnessing-gpt-and-free-huggingface-llm-alternatives-9a106c239975> . JPG.
20. Databricks. [Hugging Face Pipelines'ının Çalışma Şeklini Gösteren Görsel]. 2023. [https://www.youtube.com/watch?v=-T8iDxLMuuk&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB\\_1yZm&index=12](https://www.youtube.com/watch?v=-T8iDxLMuuk&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB_1yZm&index=12) . PNG.
21. Databricks. [Vektörler Arasında Yapılan Benzerlik Aramasının Çalışma Şeklini Gösteren Görsel ].2023. [https://www.youtube.com/watch?v=X5DZL58mBg0&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB\\_1yZm&index=20](https://www.youtube.com/watch?v=X5DZL58mBg0&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB_1yZm&index=20) . PNG.
22. Databricks. [Vektörler Veri Tabanı Kullanan LLM'lerin Çalışma Şeklini Gösteren Görsel]. 2023. [https://www.youtube.com/watch?v=X5DZL58mBg0&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB\\_1yZm&index=20](https://www.youtube.com/watch?v=X5DZL58mBg0&list=PLTPXxbhUt-YWSR8wtILixhZLF9qB_1yZm&index=20) . PNG.
23. Pavan Belagatti. [Vektör Veri Tabanı Kullanan LLM'lerin Çalışma Şeklini Gösteren Görsel]. 2023. <https://dev.to/pavanbelagatti/wtf-is-a-vector-database-a-beginners-guide->

16p . JPG.

24. Deci AI. [RetrievalQA Çalışma Adımlarını Gösteren Görsel]. 2024.

[https://x.com/deci\\_ai/status/1767903314493280655](https://x.com/deci_ai/status/1767903314493280655) . JPG.

25. Heiko Hotz. [RAG Tekniğinin Çalışma Adımlarını Gösteren Görsel ]. 2023.

<https://towardsdatascience.com/rag-vs-finetuning-which-is-the-best-tool-to-boost-your-llm-application-94654b1eaba7> . JPG.

26. Rick Merritt. [Sistemin Blok Diyagramı ]. 2023. <https://towardsdatascience.com/rag-vs-finetuning-which-is-the-best-tool-to-boost-your-llm-application-94654b1eaba7> . JPG.

## **6. EKLER (varsa)**

## STANDARTLAR ve KISITLAR FORMU

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Projemiz, Explainable AI (EAI) uygulamaları geliştirmek için Llama2 LLM modelini kullanmayı amaçlayan yeni bir projedir. Bu proje, Llama2 modelinin EAI konulu makaleler üzerinde önceden eğitilmesi ve modelin verdiği cevapların hangi kaynak aracılığıyla verdiğini göstermesi için bir mekanizma geliştirilmesi gibi yeni fikirler ve konseptler geliştirmeyi içerir. Projenin genel kapsamında, tasarımımız yaklaşık %30'luk bir paya sahiptir ve modelin eğitimi, kaynak gösterme mekanizması ve kullanıcı arayüzü gibi temel bileşenleri kapsamaktadır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Bu proje, Explainable AI konusunda mevcut dil modellerini geliştirmek amacıyla bir mühendislik problemi formüle etme ve çözme sürecini içermektedir. Az miktardaki veri kullanımı, modelin genel bilgi düzeyini arttırmak için özelleştirilmiş bir embedding işlemi gerektirmiştir. Bu mühendislik problemi, Langchain kütüphanesi ve Pinecone veri tabanı gibi araçlar kullanılarak başarılı bir şekilde çözülmüştür.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Bu projede, önceki derslerde edinilen dil işleme, veri madenciliği ve yapay zeka konularındaki bilgi ve beceriler büyük ölçüde kullanılmıştır. Ayrıca, Google Collab gibi SaaS platformları ve Hugging Face kütüphanesi üzerindeki API'lerin kullanımı konularında da önceki derslerden elde edilen bilgi ve deneyimler değerli olmuştur.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

(4)Nitelikli Eğitim



5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi:

Proje maliyet etkin bir şekilde gerçekleştirilmiş ve kullanılan kaynaklar optimize edilmiştir.

b) Çevre sorunları:

Yüksek enerji gereksinimine sahip işlemler Google Colab gibi bulut tabanlı platformlar üzerinde gerçekleştirilerek fiziksel çevre üzerindeki etkiler minimize edilmiştir

c) Sürdürülebilirlik:

Proje süreçleri, mevcut kaynakları verimli bir şekilde kullanmayı amaçlayan sürdürülebilir bir yaklaşımı benimsemiştir.

d) Üretilirlik:

Kullanılan araçlar ve platformlar, proje süreçlerini tekrarlamaya uygun, üretilebilir bir yapıyı desteklemiştir.

e) Etik:

Proje süreçleri ve sonuçları, etik kurallar ve standartlara uygun bir şekilde gerçekleştirilmiştir.

f) Sağlık:

Kullanılan SaaS platformları ve API'ler, sağlık açısından riski minimize edecek şekilde seçilmiştir.

g) Güvenlik:

Proje süreçleri, veri güvenliği ve kullanıcı güvenliği konularında önceden belirlenmiş standartlara uygun olarak gerçekleştirilmiştir.

h) Sosyal ve politik sorunlar:

Proje, toplumun sosyal ve politik konularına duyarlı bir şekilde tasarlanmış ve uygulanmıştır.