Retriever: Arama Amaçlı RAG Modülü

1st Tamer Bayar
220205027
220205027@ostimteknik.edu.tr

I. Özet

Bu çalışma, hassas alanlarda dahi kullanılabilecek, yüksek isabetli bir Hibrit Semantik Arama ve Bilgi Erişim (Hybrid Semantic Retrieval) sistemi sunmaktadır. Sistemin etkililiğini kanıtlamak için anlamsal olarak hassas ve zorlu terimlere sahip olan hukuk alanı seçilmiştir, istenilirse farklı alanlarda arama yapmak için kolayca düzenlenilebilir. Geleneksel anahtar kelime tabanlı arama yöntemlerinin yetersiz kaldığı ve güncel üretken yapay zeka modellerinin yavaş ve hukuk gibi kritik alanlarda halüsinasyon riski taşıdığı göz önünde bulundurularak; LLM kullanmayan, anlamsal benzerlik ve anahtar kelime eşleşmesinden yararlanan hibrit bir puanlama algoritması kullanılmıştır.

Çalışma kapsamında örnek veri seti olarak, 2080 adet hukuk soru-cevap çifti Sentence-Transformers mimarisi kullanılarak gömülmüş ve FAISS(Facebook AI Similarity Search) kütüphanesi ile optimize edilmiştir. Sistemin özgün yanı hibrit skorlama algoritmasıdır. Bu algoritma, anlamsal olarak yakın fakat istenmeyen eşleşmeleri filtreleyerek doğruluğu artırmaktadır. Ayrıca anlamsal arama yapması sayesinde farklı dillerdeki sonuçları da bulabilmektedir. Sonuç olarak sistemin düşük gecikme süresiyle yüksek doğrulukta sonuçlar ürettiğini ve kapsamlı bir RAG mimarisi için güvenilir bir doğru bilgiyi getirme (retrieval) katmanı oluşturduğunu göstermektedir.

II. Amaç

Bu projenin temel amacı, seçilen hukuk gibi teknik ve anlamsal derinliği yüksek olan bir alanda, kullanıcıların sorularına cevap olması için benzer yazılara hızla ulaştırabilen bir bilgi erişim sistemi geliştirmektir. Proje hedefleri:

- Kelime bazlı değil anlam bazlı arama: Günümüzde halen bazı orta ölçekli web sitelerinde kullanılmakta olan kelime benzerliği ile arama, çoğu zaman istenen sonucu verememektedir. Kullanıcının aradığı sonuç ile birkaç harf uyuşmazlığı olduğu için istenen sonuç sunulamamakta ve kullanıcıların deneyimini kötüleştirmektedir. Yeni kullanıcılar, aradığı sonucu bulmak için başka web sitelerine yönelebilmekte veya profesyoneller aradıkları kaynaklara ulaşmakta zorlanabilmektedir. Bu çalışma, hibrit arama yöntemi kullanarak her türden arama ihtiyacı için hızlı ve doğru sonuç vermeyi amaçlamakta.
- Bilgiyi doğru getirme: Güncel üretken yapay zeka modellerinin en büyük problemi olan bilgi uydurma riskini tamamen ortadan kaldırmak.
- Hibrit skorlama: Sadece vektör benzerliğine güvenmek yerine, benzerlik puanını anahtar kelime eşleşme oranı ile güncelleyen özgün bir algoritma. Bu sayede anlamsal olarak yakın görünse de hukuki sonucu tamamen değiştirebilecek farklılıkları (farklı kanun maddeleri veya kavramlar) ayırt edebilmek.
- Milyonlarca satırlık veri setlerinde dahi arama süresini minimumda tutmak için FAISS kütüphanesini kullanarak, donanım kaynaklarını en verimli şekilde kullanan bir altyapı oluşturmak.
- Gelecekte potansiyel olarak geliştirilebilecek tam kapsamlı bir arama asistanı projesi için, en kritik aşama olan retrieval katmanını, modüler ve test edilebilir bir yapıda kurmak.

III. Literatür Taraması

A. *Kelime Bazlı Arama (Lexical Search)*

Geleneksel sistemler (örneğin SQL LIKE sorguları veya Elasticsearch BM25 algoritması), dökümanları sadece karakter dizisi benzerliğine göre sıralar. Bu yöntemlerin en büyük sorunu, kullanıcının belirttiği metinle hedef metin arasında bir harf farkı olması veya eş anlamlı kelimeler kullanılması durumunda sonuç üretememesidir. Örneğin Hukuk gibi hassas terimlerin olduğu bir alanda, ”nikah” araması yapan bir kullanıcıya ”evlilik” kullanılmış olan sonuçlar verilemez.

B. *Anlamsal Arama ve Embedding Modelleri*

Daha güncel gelişmelerle, metinleri vektörlere çeviren (Word2Vec, ardından BERT ve türevleri) modeller geliştirilmiştir. Reimers ve Gurevych (2019) tarafından sunulan Sentence-BERT mimarisi, cümlelerin anlamlarını çıkarmayı sağlayan bir sistemdir. Bu çalışmada kullanılan MiniLM tabanlı modeller, metinleri 384 veya daha fazla boyutta, farklı kelimeler kullanılsa bile anlam benzerliğini kosinüs benzerliği kullanarak hesaplayabilmektedir.

C. *RAG ve Güvenilirlik Problemi*

Güncel RAG mimarisi, döküman erişimi ve metin üretimini birleştiren bir yöntemdir. Fakat hassas alanlarda, arama sonuçlarını LLM’den geçirmenin güvenilirliği hala bir tartışma konusudur. Bu nedenle ”Retrieval-only” veya ”Hybrid Retrieval” olarak adlandırılan ve sadece doğrulanmış bilgiyi getirmeye odaklanan sistemler şimdilik tercih edilmekte.

IV. Metodoloji

A. *Araçlar*

Proje, hızlı geliştirme avantajı ve kolaylığı sebebiyle bulut tabanlı bir ortam olan Google Colab üzerinde kurulmuştur. Eğitim aşamasında NVIDIA T4 GPU kullanılmıştır fakat modelin küçük boyutu nedeniyle daha zayıf donanım ile de hızlı sonuç alınabilir. Yazılım dili olarak Python tercih edilmiş olup, temel kütüphaneler şunlardır:

- Sentence-Transformers: Metinleri vektör uzayına dönüştürmek için kullanılmıştır.
- FAISS: Vektörler arasındaki benzerliği hızla hesaplamak için kullanılmıştır.
- Gradio: Son kullanıcının sistemle etkileşime girebilmesi için web tabanlı bir arayüz oluşturulmuştur.
- Stopwords_tr: Türkçe dolgu kelimelerin (ve, ile, ama vb.), skor hesaplamasını etkilememeleri amacıyla temizlenmeleri için kullanılmıştır.

B. *Veri Seti*

Sistemin eğitimi ve test edilmesi için Hugging Face platformunda yer alan ”alibayram/hukuk_soru_cevap” veri seti kullanılmıştır. Veri seti, hukuki alanlara dair 2080 adet soru-cevap çiftinden oluşmaktadır. Bu veri seti, modelin hukuki terminolojiye aşinalığını göstermek için saf haliyle kullanılmış fakat vektörleştirme öncesinde basit bir ön işleme yapılmıştır.

C. *Kullanılan Embedding Modeli*

Vektörleştirme işlemi için ”sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2” modeli tercih edilmiştir. Bu modelin seçilme sebebi: küçük olması, açık kaynak olması, 50’den fazla dil desteği ve 12 katmanlı yapısı sayesindeki verimliliğidir.

D. *Hibrit Arama Algoritması ve Parametreler*

Bu çalışmanın özgün yönü, çeşitli yöntemlerin güçlü yönlerini birleştiren, kural tabanlı algoritmasıdır. Algoritma çalışma adımları:

*1) **Vektör Araması ile Aday Havuzu belirleme**:* Kullanıcıdan gelen soru, noktalama işaretleri ve etkisiz kelimeler (stopwords) temizlenerek vektörleştirilir ve FAISS ile Kosinüs Benzerliği kullanılarak en yakın 50 aday soru getirilir.

*2) **Re-Ranking**:* Yalnızca vektör benzerliği kullanmak, özellikle bir LLM’den destek alınmadığı durumlarda, vektörel olarak birbirine yakın fakat kullanıcının aradığı bağlamdan uzak sonuçlar getirebilmektedir. Bunu önlemek amacıyla, kelime eşleşmesine dayalı, kurallı bir skorlama yöntemi geliştirilmiştir.

Final skor (S_{final}), aşağıdaki parçalı fonksiyon ile hesaplanır:

$$S_{final} = \begin{cases} S_{vec} + \left(\frac{m}{n} \times 0.30\right) & \text{eğer } m > 0 \\ S_{vec} \times 0.80 & \text{eğer } m = 0 \text{ ve } S_{vec} < 0.40 \\ S_{vec} & \text{diğer durumlarda} \end{cases} \quad (1)$$

Parametre Açıklamaları:

- m: Sorgu ve hedef metin arasındaki ortak anahtar kelime sayısı.
- n: Sorgudaki toplam temizlenmiş kelime sayısı.
- S_{vec}: Başlangıçtaki vektör benzerlik skoru.
- 0.30: Eşleşme ödül katsayısı.
- 0.80: Düşük güvenilirlik ceza katsayısı.

Bu formül sayesinde; ortak kelime içeren sonuçlar ödüllendirilirken, ortak kelime içermeyen ancak yüksek vektör benzerliğine ($S_{vec} \geq 0.40$) sahip sonuçlar anlamsal eşleşme(semantic match) olarak kabul edilip korunmaktadır. Bu kural sayesinde farklı dillerde yazılmış fakat yakın anlamlı sonuçların, kelime uyumsuzluğundan dolayı daha düşük puan almamaları hedeflenmiştir. Sadece hem kelime eşleşmesi olmayan hem de benzerliği düşük olan sonuçlar cezalandırılmaktadır.

3) Hesaplanan S_{final} değeri, 0.45 eşik değerinin altındaysa sonuç kullanıcıya gösterilmez.

V. Sonuç

Yapılan testlerde, sistemin ”taşınmazda vergi borcu” veya ”reddi miras” gibi hukuki terim içeren sorularda, doğrudan kelime eşleşmesi olmasa dahi anlamsal bütünlüğü yakaladığı ve istenen sonuçları getirebildiği görülmüştür. Puanlar hesaplandıktan sonra kullanıcıya en yüksek puanlı 5 sonuç gösterilmiştir. Ortalama yanıt süresi kullanılan veri setinde Colab CPU kullanıldığında dahi 1 saniyedir.

VI. Gelecek Çalışmalar ve Gereksinimler

Bu çalışma, ”Retriever” modülünü kullanan bir üründür. Sistem geliştirilip tam bir Yapay Zeka Asistanına dönüştürülmek istenirse şu geliştirmeler yapılabilir:

- Mevcut sistemde Nvidia T4 kullanılarak ufak bir veri seti gömülmekte ve tek bir kullanıcı tarafından sorgu gönderilmektedir. Büyük miktarda verinin düzenli aralıklarla sisteme eklenmesi ve aynı anda yüzlerce kullanıcıya kesintisiz sorgu hizmeti verilebilmesi için Nvidia A100 gibi güçlü bir GPU’ya ihtiyaç duyulacaktır.
- Şu an Google Colab üzerinde çalışan sistem, oturum sonlandığında kapanmakta ve bilgileri unutmaktadır. Sistemin 7/24 hizmet verebilmesi, veritabanının kalıcı olması ve mobil uygulamalarla entegre edilebilmesi için bir bulut sunucuya taşınması gerekmektedir.
- Elde edilen yüksek doğruluklu dökümanlar, bir LLM’e ”bağlam” (context) olarak verilerek, kullanıcının sorusuna İşin uzmanına soruyormuşçasına cevap veren tam otonom bir sistem kurulabilir. Yapılan testlerde Colab’da çalışabilecek kadar küçük ve açık kaynaklı modellerin bu konuda başarısız olduğu görülmüştür, daha iyi donanıma erişim sağlandığında model tekrar eğitilerek ”özelleşmiş bir asistan” sistem ortaya çıkarılabilir.
- Bu çalışmada çoklu dil desteği göz önünde bulundurulmuş fakat tam destek sağlanmamıştır. Mevcut sistemde farklı dillerdeki sonuçlar, ortak anahtar kelime olmamasından dolayı %20 oranında kırılabilmektedir. Daha

büyük ve çok dilli bir veri seti sağlandığında puanlama algoritması optimize edilerek kullanıcının sorgusuna, farklı dillerdeki sonuçları getirmesi sağlanabilir.