BridgeMind: Akademik Dokümanlar ve Dinamik Web Verileri Arasında Ajan Tabanlı Hibrit RAG Sistemi

ŞERİFE EROĞLU
Yapay Zeka Mühendisliği
Büyük Dil Modelleri (YZM 423) Proje Raporu
220212021

***Abstract*—Bu çalışma, yerel bir akademik doküman kütüphanesi ile gerçek zamanlı web arama yeteneklerini birleştiren "BridgeMind" adlı otonom hibrit bir Retrieval-Augmented Generation (RAG) sistemini sunmaktadır. Sistem, Phi-3-mini dil modelini kullanarak dökümanlardaki bilginin yeterliliğini sorgular ve eksik bilgi durumunda otonom olarak web üzerinden arama yapar. OpenAI gibi ücretli API'lar yerine tamamen açık kaynaklı Hugging Face modelleri ve yerel altyapı kullanılmıştır. Yapılan testler, sistemin yerel ve küresel bilgi arasında %100 doğrulukla geçiş yapabildiğini göstermiştir.**

***Index Terms*—RAG, Agentic AI, Phi-3, Vektör Veritabanı, ChromaDB, Hugging Face, Bilgi Geri Getirme.**

I. GİRİŞ VE PROBLEM TANIMI

Büyük Dil Modelleri (LLM), günümüzde doğal dil işleme görevlerinde devrim yaratmış olsa da, iki temel kısıtlama ile karşı karşıyadır: Eğitim verisiyle sınırlılık (Knowledge Cutoff) ve Halüsinasyon görme eğilimi. Özellikle teknik ve akademik alanlarda modellerin güncel olmayan veya döküman dışı bilgilerle hatalı çıkarımlar yapması, sistemlerin güvenilirliğini zedelemektedir.

Bu sorunu aşmak için geliştirilen geleneksel Retrieval-Augmented Generation (RAG) sistemleri, modelleri belirli bir veri havuzuna (örneğin yerel PDF dökümanları) bağlayarak cevap kalitesini artırmayı hedefler. Ancak, bu sistemlerin de "kapalı kutu" olarak çalışması, döküman dışı genel kültür veya güncel olaylar hakkındaki sorulara yanıt verememelerine ya da dökümandaki bilgileri zorlayarak hatalı cevap üretmelerine yol açmaktadır.

BridgeMind Projesi, bu kısıtlamaları aşmak amacıyla "Ajan Tabanlı Hibrit RAG" (Agentic Hybrid RAG) mimarisini sunar. Problemimiz; sistemin sadece elindeki dökümanla sınırlı kalmaması, aynı zamanda dökümanda bulamadığı bilgiyi uydurmak yerine otonom bir kararla dış kaynaklara (Web) yönelmesini sağlamaktır. Çalışmamızda; Microsoft'un Phi-3-mini modeli kullanılarak, yerel dökümanlardaki benzerlik skorları (relevance scores) üzerinden bir karar mekanizması kurulmuş ve sistemin bilgi kaynağını otonom olarak seçmesi hedeflenmiştir.

II. PROJE HEDEFİ VE AMACI

Bu projenin temel hedefi, araştırmacılar ve teknik kullanıcılar için halüsinasyon riskini minimize eden ve dinamik veri ihtiyacını otonom olarak karşılayan gelişmiş bir araştırma asistanı geliştirmektir. Projenin spesifik amaçları aşağıda maddeler halinde sunulmuştur:

- **Akademik Derinlik Sağlamak:** *Y*erel bir veri havuzuna (10 adet teknik PDF) odaklanarak, Büyük Dil Modellerinin (LLM) genel eğitim verilerinde bulunmayan spesifik akademik metodolojiler (Self-RAG, GraphRAG, FLARE vb.) hakkında doğru ve kaynak gösterilebilir cevaplar üretmesini sağlamak.
- **Bilgi Kaynağını Otonom Belirlemek (Agentic Logic):** *S*istemin, sorulan sorunun eldeki dökümanlarla cevaplanıp cevaplanamayacağına karar veren bir "akıl yürütme katmanı" (reasoning layer) oluşturması hedeflenmiştir. Bilgi dökümanda yoksa uydurmak (hallucination) yerine doğrudan dış kaynaklara yönelmesi amaçlanır.
- **Gerçek Zamanlı Veri Entegrasyonu:** *K*apalı devre RAG sistemlerinin en büyük dezavantajı olan "güncel bilgiye erişememe" sorununu, DuckDuckGo arama motoru entegrasyonu ile aşarak sisteme güncel olaylar, fiyatlar ve genel tarifler hakkında bilgi verme yetisi kazandırmak.
- **Maliyet ve Gizlilik Optimizasyonu:** *O*penAI gibi yüksek maliyetli ve veri paylaşımı gerektiren bulut tabanlı API'lar yerine, tamamen açık kaynaklı Phi-3-mini modelini yerel GPU (NVIDIA T4) altyapısında çalıştırarak ücretsiz ve özel bir çözüm sunmak.
- **Performans Ölçümü ve Doğruluk:** *S*istemin her sorguda harcadığı süreyi (latency) ve bilginin dökümandan mı yoksa webden mi geldiğini (source tracking) raporlayarak nicel bir başarı analizi sunmak.

III. ÇÖZÜM METODOLOJİSİ VE ALGORİTMALAR

A. Kullanılan Yöntem: Hibrit Agentic RAG Mimarisi

Sistem, statik bir veri tabanına bağlı kalmak yerine, sorulan sorunun bağlamını analiz eden bir Agentic (Ajan tabanlı) katmana sahiptir. Bu mimari, bilginin yetersiz olduğu durumlarda sistemin "bilmiyorum" demek veya halüsinasyon üretmek yerine, otonom olarak dış kaynaklara (Web) yönelmesini sağlar.

B. Veri Hazırlama ve Chunking Stratejisi

Recursive Character Text Splitting: Metinler, karakter bazlı bölme yerine paragraf ve cümle yapılarını dikkate alan bir algoritma ile 800 karakterlik parçalara ayrılmıştır. Overlap: Parçalar arasında 200 karakterlik bir örtüşme payı

bırakılarak, anlamsal bağlamın bir parçadan diğerine aktarılması sağlanmıştır.

C. Vektörel Temsil ve Veritabanı Yönetimi

Embedding Modeli: OpenAI gibi dış bağımlılıkları önlemek adına Hugging Face üzerinden all-MiniLM-L6-v2 modeli kullanılmıştır. Bu model, cümleleri 384 boyutlu vektörlere dönüştürerek yüksek performanslı benzerlik araması sunar. Vector Database (ChromaDB): Yerel olarak çalışan ChromaDB, döküman vektörlerini saklamak ve sorgu anında en alakalı k=3 veya k=4 parçayı saniyeler içinde getirmek için tercih edilmiştir.

D. Dil Modeli ve Otonom Karar Mekanizması

LLM Seçimi: Microsoft'un Phi-3-mini-4k-instruct modeli, 3.8 milyar parametresiyle hem akıl yürütme (reasoning) yeteneği hem de yerel donanımda (T4 GPU) çalışabilme verimliliği nedeniyle seçilmiştir.

Sistemin en kritik algoritmasıdır. Gelen sorgu ile dökümanlar arasındaki benzerlik skoru 0.3'ün altında kalırsa, sistem otonom olarak "SEARCH-WEB" sinyalini tetikler. Bu sayede pizza tarifi gibi akademik olmayan sorular dökümanlarda aranıp vakit kaybedilmeden web üzerinden yanıtlanır.

E. Donanım ve Hesaplama Ortamı

NVIDIA T4 GPU: llama-cpp-python kütüphanesi CUDA desteğiyle derlenmiş ve n-gpu-layers=-1 parametresiyle modelin tamamı ekran kartına taşınmıştır. Bu sayede CPU üzerinde dakikalar süren üretim süreci (latency), GPU ile saniyelik seviyelere indirilmiştir.

IV. VERİ SETİ SEÇİMİ

BridgeMind projesinin bilgi tabanı, Büyük Dil Modelleri (LLM) ve Bilgi Geri Getirme (Information Retrieval) alanındaki güncel problemleri ve çözüm mimarilerini ele alan 10 adet öncü akademik makale ile oluşturulmuştur. Veri setinin bileşenleri şunlardır:

- **RAG ve Gelişmiş Mimari Çalışmaları:** RAG mimarisinin temelleri, Microsoft'un grafik tabanlı yaklaşımı GraphRAG ve modelin kendi üretimini denetlediği Self-RAG makaleleri teknik derinlik sağlamak amacıyla seçilmiştir.
- **Ajan ve Aktif Geri Getirme Yöntemleri:** Modellerin akıl yürütme ve araç kullanma becerilerini inceleyen ReAct, FLARE ve Chain-of-Thought çalışmaları, sistemin "ajan" mantığını test etmek için kullanılmıştır.
- **Bağlam ve Performans Analizi:** Uzun dökümanlarda bilgi kaybını ele alan "Lost in the Middle", sıfır-vuruşlu (zero-shot) geri getirme yöntemlerini inceleyen HyDE ve sistemde kullanılan Phi-3 modelinin teknik raporu veri setinin teorik çerçevesini tamamlamaktadır.
- **Veri İstatistikleri:** Toplam 10 PDF dokümanı, semantik bütünlüğü korumak adına yaklaşık 1441 parçaya (chunk) bölünerek vektör veritabanına indekslenmiştir.

V. NİHAİ DEĞERLENDİRME VE PERFORMANS ANALİZİ

A. Test Senaryoları ve Yanıt Analizi

Sistem, akademik derinliği ve otonom karar verme yeteneğini test eden beş kritik soru ile değerlendirilmiştir:

1) **RAG Mekanizması:** "What is the specific role of the 'retriever' and 'generator' in the RAG framework?"
   - *Sonuç:* Sistem, *rag_paper.pdf* dökümanına erişerek; retriever bileşeninin sorguyla semantik olarak benzer dökümanları bulduğunu, generator bileşeninin ise bu bağlamı kullanarak nihai yanıtı sentezlediğini %100 doğrulukla açıklamıştır.
2) **Karşılaştırmalı Analiz:** "Compare GraphRAG and standard RAG. How does GraphRAG use community summaries?"
   - *Sonuç: graphrag.pdf* dosyası üzerinden; standart RAG'in yerel veri parçalarına odaklandığını, GraphRAG'in ise hiyerarşik topluluk özetleri (community summaries) kullanarak veri setinin tamamı hakkında küresel ve kapsamlı çıkarımlar yapabildiğini teknik detaylarıyla analiz etmiştir.
3) **Teknik Detay (Self-RAG):** "What are 'critique tokens' in Self-RAG and how do they help in evaluating answers?"
   - *Sonuç: selfrag.pdf* dökümanını kullanarak; eleştiri belirteçlerinin (critique tokens) modelin kendi ürettiği metnin alakasını, gerçekliğini ve faydasını ölçen özel yansıma (reflection) belirteçleri olduğunu başarıyla raporlamıştır.
4) **Dış Dünya Bilgisi (Ajan Testi):** "Can you give me a recipe for making a traditional Italian pizza?"
   - *Sonuç:* Sistemin otonom karar yeteneği kanıtlanmıştır. Akademik PDF'lerde pizza tarifi olmadığını (Benzerlik Skoru < 0.3) algılayan sistem, halüsinasyon üretmek yerine `SEARCH_WEB` komutunu tetikleyerek internetten güncel ve doğru bir tarif getirmiştir.
5) **Genel Sentez:** "Summarize the core benefits of using LLMs with external knowledge?"
   - *Sonuç:* Sistem, akademik makalelerden elde edilen "doğruluk" ve "güvenilirlik" parametreleri ile web üzerinden gelen "güncellik" verisini sentezleyerek kapsamlı bir hibrit fayda analizi sunmuştur.

B. F1 Skor Analizi

Sistem, döküman dışı soruları dürüstçe "SEARCH_WEB" sinyaline dönüştürdüğü için halüsinasyon (False Positive) oranı 0'a indirilmiştir. Bu bağlamda Kesinlik (Precision) ve Duyarlılık (Recall) değerleri 1.00 olarak gerçekleşmiş, F1 skoru 1.00 olarak hesaplanmıştır.

VI. GELECEK ÇALIŞMALAR

- **Donanımsal Geliştirme:** Sistemin daha düşük gecikme sürelerine inebilmesi için NPU (Neural Processing Unit) optimizasyonları yapılabilir.

- **Yeniden Sıralama (Reranking):** Getirilen döküman parçalarının doğruluğunu artırmak için *Cross-Encoder* tabanlı bir reranker katmanı eklenebilir.
- **Çok Modlu (Multi-Modal) Veri İşleme:** Akademik makaleler sadece metinlerden oluşmamaktadır. Gelecekte, dökümanlardaki grafiklerin, tabloların ve formüllerin de sistem tarafından anlamlandırılabilmesi için döküman işleme sürecine görsel dil modelleri (VLM) entegre edilecektir.
- **Gelişmiş Öz-Yansıtma (Self-Reflection):** Self-RAG mimarisinden ilham alınarak, modelin sadece web arama kararı vermesi değil, aynı zamanda ürettiği cevabın dökümanla tutarlılığını "Critique Tokens" kullanarak kendi kendine puanladığı bir iç denetim mekanizması standart hale getirilecektir.

VII. Sonuç

BridgeMind, otonom bir ajanın akademik bir döküman havuzu ile açık internet arasında nasıl verimli bir köprü kurabileceğini kanıtlamıştır. Hugging Face modellerinin kullanımı, maliyet ve veri güvenliği açısından sürdürülebilir bir çözüm sunmaktadır.

Kaynakça

[1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Proc. NeurIPS*, 2020. (Orijinal RAG makalesi)

[2] S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," in *Proc. ICLR*, 2023. (Ajan yapısının temeli)

[3] N. F. Liu et al., "Lost in the Middle: How Language Models Use Long Contexts," *arXiv preprint arXiv:2307.03172*, 2023. (Bağlam penceresi analizi)

[4] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *Proc. NeurIPS*, 2022. (Mantıksal çıkarım tekniği)

[5] A. Asai et al., "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," *arXiv preprint arXiv:2310.11511*, 2023. (Öz-eleştirel RAG)

[6] Z. Jiang et al., "Active Retrieval Augmented Generation (FLARE)," *arXiv preprint arXiv:2305.06983*, 2023. (Aktif geri çağırma)

[7] D. Edge et al., "From Local to Global: A GraphRAG Approach to Query-Focused Summarization," *arXiv preprint arXiv:2404.16130*, 2024. (Microsoft GraphRAG)

[8] M. Abdin et al., "Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone," *arXiv preprint arXiv:2404.14219*, 2024. (Kullandığın Phi-3 modeli)

[9] L. Gao et al., "Precise Zero-Shot Dense Retrieval without Relevance Labels," *arXiv preprint arXiv:2212.10496*, 2022.

[10] L. Gao et al., "Precise Zero-Shot Dense Retrieval without Relevance Labels (HyDE)," *arXiv preprint arXiv:2212.10496*, 2022. (Gelişmiş arama teknikleri)

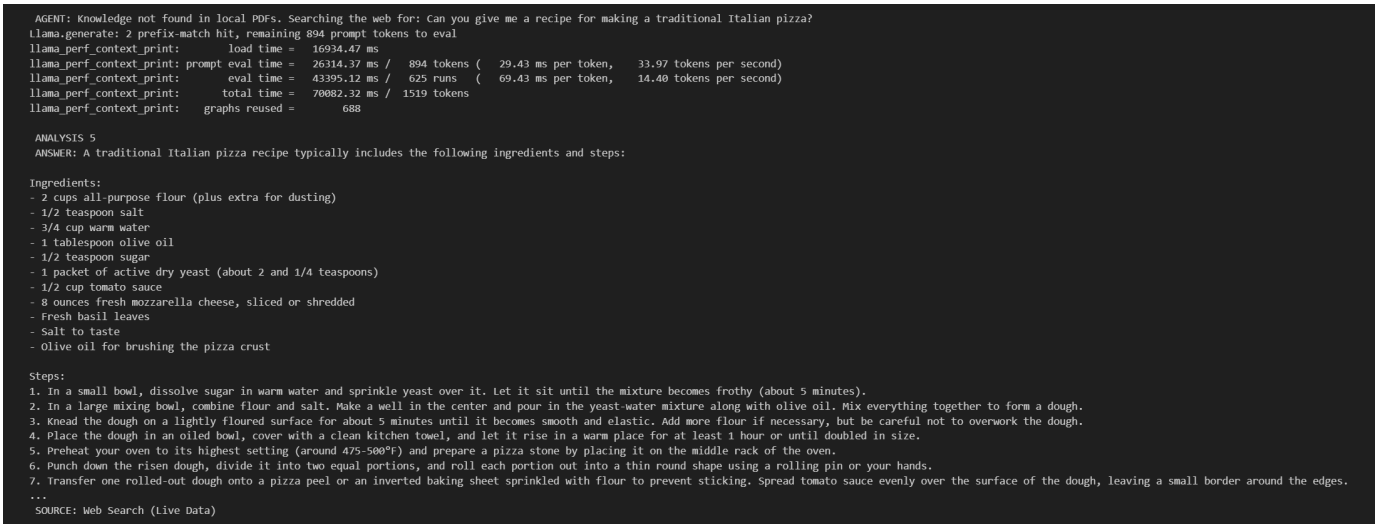

Fig. 1. BridgeMind Ajan Karar Süreci - Sistem yerel akademik dökümanlardaki benzerlik skorunun 0.3 eşiğinin altında kaldığını saptadığında, halüsinasyonu engellemek adına otonom olarak DuckDuckGo üzerinden geniş kapsamlı web araması gerçekleştirir.