Llamaindex ile Belge Tabanlı Sorgulama ve Değerlendirme

Hazırlayan

İrem KUMLU

İçindekiler

1. Giriş

Bu rapor, LLamaIndex kütüphanesi kullanılarak geliştirilen Retrieval-Augmented Generation (RAG) modelinin oluşturulması, test edilmesi ve değerlendirilmesi sürecini detaylandırmaktadır. Model, Groq API üzerinden LLama 3.3-70B Versatile modelini kullanarak sorgulara metin tabanlı yanıtlar üretmekte ve bu yanıtların doğruluğunu Cosine Similarity metriğiyle ve kullanıcı değerlendirmesi ile değerlendirmektedir.

2. Kullanılan Teknolojiler

- **LLM:** Groq API üzerinden LLama 3.3-70B Versatile
- **Embedding Modeli:** Data-Lab/multilingual-e5-large-instruct-embedder-tgd
- **Veritabanı:** ChromaDB
- **Chunking:** SentenceSplitter (chunk_size=1024, chunk_overlap=128)
- **Retrieval Parametreleri:** Top_k ve Similarity Cutoff
- **Değerlendirme:** Cosine Similarity metriği
- **Kullanıcı Değerlendirmesi:** Kullanıcı, gerçek cevap ile modelin ürettiği yanıtı karşılaştırarak değerlendirme yapmıştır.

3.Kullanılan Veri:

Bu çalışmada, modelin eğitimi ve test edilmesi için "yapayz.pdf" isimli Türkçe bir belge kullanılmıştır. Bu belge, modelin verilerini işlemek ve testleri gerçekleştirmek için gerekli olan ana kaynağı oluşturmuştur. PDF belgesi, Colab ortamında bir "data" klasörü oluşturularak bu klasöre yerleştirilmiş ve veriler burada işlenmiştir.

Belge İçeriği: "yapayz.pdf" adlı belge, yapay zekânın günümüzdeki gelişimini ve özellikle üretken yapay zekâ ile ayırt edici yapay zekâ arasındaki farkları ele alır. Bu belge, yapay zekânın edebiyat, müzik, sinema, tasarım, mimari, eğitim gibi pek çok alanda etkileyici bir şekilde nasıl şekillendiğini ve insanlık kapasitesini yeniden keşfetmesini vurgular.

4. Model Geliştirme Süreci

4.1 Veri İşleme ve Chunking

Veriler, **SentenceSplitter** kullanılarak belirlenen **chunk_size** ve **chunk_overlap** parametreleri ile bölünmüştür. Farklı boyutlardaki chunk'ların modelin yanıt kalitesi üzerindeki etkileri test edilmiştir.

4.2 Vektör Veritabanı ve Indexleme

Veriler, **Data-Lab/multilingual-e5-large-instruct-embedder-tgd** embedding modeli ile vektörlere dönüştürülmüş ve **ChromaDB** üzerinde saklanmıştır.

4.3 Sorgu ve Yanıt Üretme

Groq API üzerinden **LLama 3.3-70B Versatile** modeli kullanılarak sorgulara yanıt üretilmiş ve retrieval sürecinde **top_k** ve **similarity_cutoff** parametreleri ile en uygun yanıtlar belirlenmiştir.

5. Parametre Testleri ve Sonuçlar

Farklı parametreler ile testler gerçekleştirilmiş ve model yanıtları Cosine Similarity metriği ile değerlendirilmiştir. Aşağıdaki tabloda, seçili parametreler ve elde edilen benzerlik skorları özetlenmiştir:

Chunk Size	Chunk Overlap	Top_k	Similarity Cutoff	Temperature	Cosine Similarity
1024	128	5	0.95	0.7	0.7190
2048	256	10	0.75	0.8	0.7200
512	64	15	0.85	1.0	0.6702

Yanıtların detaylı incelemesi sonucunda, chunk size ve similarity cutoff değerlerinin model performansını doğrudan etkilediği gözlemlenmiştir.

6.PDF İçeriğinden Soruya Cevap Üretilecek Kısım: Ayırt Edici ve Üretken Modeller

İstatistiksel modeller ayırt edici (discriminative) ve üretken (generative) olarak iki başlık altında incelenebilir. Ayırt edici modeller, sağladıkları avantajlarla uzun yıllardır ön planda iken son zamanlarda üretken modellerin popülerliği hızla yükseliyor. Peki, bu ayırt edici ve üretken modeller nedir ve ne işe yarar? Basit bir örnek üzerinden gidelim. Elimizde milyonlarca kedi ve köpek fotoğrafı olduğunu farz edelim. Ama tüm fotoğraflar birbirine karışmış ve açıp bakmadan hangi fotoğrafın kediye hangisinin köpeğe ait olduğunu bilemiyoruz. Görevimiz ise kedi ve köpek fotoğraflarını ayırarak düzenlemek. İnsan gücüyle bunu tek tek yapmanın ne kadar zaman alacağını tahmin edebilirsiniz. İşte bu noktada, yapay zekâ devreye girebilir ve size büyük bir kolaylık sağlayabilir. Siz sadece bazı fotoğrafları kedi veya köpek olarak etiketlersiniz. Ardından, bu etiketlenmiş fotoğrafları yapay zekâya verirsiniz. Yapay zekâ, bu fotoğraflar ve etiketler üzerinden kedi ve köpek ayırımını öğrenmeye başlar. Eğitim tamamlandığında ise kalan fotoğrafları ona sunarak sınıflandırmasını isteyebilirsiniz. İşte bu süreçte eğittiğimiz ve faydalandığımız model, ayırt edici modeldir. Ayırt edici model, fotoğraflardan kedi ve köpeği ayırt eden özellikleri öğrenmeye çalışır (örneğin kulaklar, burunlar vb.). Bu modeller örnekte olduğu gibi etiketli veri üzerinden gözetimli (supervised) öğrenme yoluyla eğitilir. Ancak amacımız hiç var olmamış bir köpek fotoğrafı oluşturmak gibi biraz sıra dışıysa süreç daha farklı işler. Bu durumda elimizdeki milyonlarca köpek fotoğrafını herhangi bir etiketleme yapmadan üretken modele veririz. Model, köpek fotoğraflarındaki verinin dağılımlarını öğrenir. Eğitim tamamlandıktan sonra ondan bir köpek fotoğrafı oluşturmasını istediğimizde bir sanatçı edasıyla bize hiç var olmamış bir köpek fotoğrafı sunabilir (tabii her zaman mükemmel sonuçlar alamayabiliriz). Ayırt edici modeller daha eski ve bilinen türler olduğu için bu yazımızda yeni geliştirilen üretken modellere, özellikle de ChatGPT'ye odaklanacağız.(syf:3)

7. Karşılaşılan Sorunlar ve Çözümler

7.1 Groq API Rate Limit Problemi

Groq API kullanımı sırasında belirli bir süre sonra **rate limit hatası** alınmıştır. Bu durum, modelin sorgulara yanıt üretme sürecinde **bazı verilerin eksik işlenmesine** neden olmuş ve Cosine Similarity hesaplamasının objektifliğini etkileyebilecek bir durum oluşturmuştur.

8. Sonuç ve Değerlendirme

Bu çalışmada, RAG modelinin farklı parametre kombinasyonlarıyla performansı değerlendirilmiş ve en yüksek **Cosine Similarity skoru 0.7200** olarak elde edilmiştir. Testler sonucunda, **chunk size, top_k, similarity cutoff ve temperature** gibi parametrelerin model yanıtlarının doğruluğu ve kapsamı üzerinde doğrudan etkili olduğu görülmüştür.

Chunk size artırıldığında modelin yanıtları daha geniş bağlam içerecek şekilde zenginleşirken, küçük chunk size kullanımı yanıtların daha öz olmasını sağlamış ancak bazı bağlamsal bilgilerin kaybolmasına neden olmuştur. **Top_k değeri arttıkça model daha fazla doküman parçasını dikkate almış, bu bazen yanıtların gereksiz detaylarla dolmasına yol açmıştır.** Öte yandan, düşük top_k değeri daha kısa ama bazen eksik yanıtlarla sonuçlanmıştır. **Similarity cutoff parametresi** düşük tutulduğunda modelin alakasız bilgileri de yanıtlarına dahil ettiği, yüksek tutulduğunda ise bazı durumlarda eksik bilgi sunduğu tespit edilmiştir. **Temperature parametresi yükseltildiğinde modelin daha yaratıcı ve çeşitli yanıtlar ürettiği, ancak bu durumun bazen bağlam dışı veya gereksiz detaylara yol açabildiği gözlemlenmiştir.**

Deneyler sırasında **Groq API rate limit problemi** nedeniyle bazı sorgular eksik işlenmiş ve modelin yanıtlarının değerlendirilmesini etkileyen durumlar oluşmuştur. Bu problem nedeniyle bazı testler tamamlanamamış veya eksik veri işlenmiştir. **Bu sorunu çözmek adına API çağrıları arasına bekleme süresi eklemek, sorguları batch olarak işlemek veya daha düşük top_k ve chunk size değerleri kullanarak API limitini daha verimli kullanmak gibi önlemler alabilirdik.**

Genel olarak, bu çalışma **RAG modellerinin Türkçe belgeler üzerinde nasıl optimize edilebileceğini ve farklı parametrelerin model performansı üzerindeki etkilerini detaylı bir şekilde ortaya koymuştur.** Elde edilen bulgular, benzer çalışmalar için bir rehber niteliğinde olup, gelecekteki optimizasyon süreçleri için faydalı olabilecek önemli çıkarımlar sunmaktadır.

9.Başarı Sonucu: Rate Limit Öncesi En Yüksek Cosine Similarity Skoru

Groq API üzerinden yapılan testler sırasında, rate limit hatası almadan önce elde edilen **en yüksek Cosine Similarity skoru 0.86 olarak kaydedilmiştir.** Bu sonuç, modelin chunk size 1024, chunk overlap 128, top_k 5, temperature 0.7 ve similarity cutoff 0.90 parametreleri ile elde edilmiştir. Bu skor, modelin parametreler üzerindeki ince ayarların yanı sıra, belgenin içeriğine ne kadar uygun yanıtlar verdiğini gösteren önemli bir başarıyı temsil etmektedir.

Bu sonuç, yüksek similarity cutoff değeri ve uygun chunking parametreleriyle modelin doğru yanıt üretme kapasitesinin arttığını ortaya koymuştur. Ancak, rate limit hatası sonrası yapılan testlerde bazı verilerin eksik işlenmesi ve modelin yanıtlarında azalma gözlemlenmiştir. Yine de, bu yüksek performanslı skor, modelin parametrelerin optimize edilmesiyle elde edilebilecek maksimum doğruluk seviyesinin önemli bir göstergesidir.