

# Retrieval Yöntemlerinin Performans Karşılaştırması

Raif AKYOL  
Bilgisayar Mühendisliği  
Yıldız Teknik Üniversitesi  
Esenler, İstanbul  
raif.akyol@std.yildiz.edu.tr

**Özetçe:** Bu çalışma, bilgi erişimi (retrieval) problemlerini çözmek için çeşitli yöntemlerin kombinasyonlarını değerlendirmektedir. BM25, dense vector representation modelleri (all-MiniLM-L12-v2, multilingual-e5-large-instruct, turkish-colbert, gte-large ve bge-m3) ve Word Matching yöntemleri kullanarak retrieval performansı ölçülmüştür. Hibrit yöntemlerle doğruluk oranlarını artırmak için ağırlıklı ortalama yöntemi uygulanmıştır. Deneyler, WikiRAG-TR veri seti üzerinde gerçekleştirilmiş ve yöntemlerin Top-1 ve Top-5 doğruluk oranları karşılaştırılmıştır. Sonuçlar, Bazı modellerin daha önce çıktığını ancak genel olarak hibrit yöntemlerin retrieval performansında belirgin bir iyileşme sağladığını göstermektedir.

**Anahtar Kelimeler**—Retrieval, BM25, Dense Vectors, Hybrid Methods, Jaccard Similarity, WikiRAG-TR.

## I. GİRİŞ

Doğal Dil İşleme (NLP) alanındaki hızlı ilerlemeler, bilgi erişimi ve metin tabanlı sorgu işlemleri için yenilikçi yöntemlerin geliştirilmesine olanak sağlamıştır. Günümüzde metin tabanlı bilgi erişimi, çeşitli alanlarda kritik bir role sahiptir. Eğitim, sağlık, e-ticaret, yargı ve daha pek çok sektörde, doğru ve hızlı bilgiye erişim, verimliliği artırmakta ve kullanıcı deneyimini iyileştirmektedir. Bu bağlamda, bilgi erişiminde kullanılan yöntemlerin doğruluk ve performans açısından geliştirilmesi önemli bir araştırma konusu olmuştur.

Bu çalışmada, bilgi erişimi için birden fazla yöntemin Türkçe metinler üzerinde karşılaştırılması ve hibrit yaklaşımlarla performans iyileştirmesi hedeflenmiştir. Kullanılan yöntemler arasında klasik BM25 algoritması, Jaccard benzerlik metriğine dayalı kelime eşleştirme ve Transformer tabanlı modeller (ör. Sentence-Transformers, E5, ColBERT) bulunmaktadır.

### A. Çalışmanın Amacı

Çalışmanın temel amacı, farklı bilgi erişim yöntemlerini değerlendirmek ve bunların ağırlıklı kombinasyonlarla nasıl optimize edilebileceğini incelemektir.

### B. Araştırmanın Hedefleri

- BM25, Jaccard benzerlik metriği ve Transformer tabanlı modellerin performanslarını Türkçe veri kümesi üzerinde değerlendirmek.
- Farklı yöntemlerin güçlü yönlerini birleştirerek, bilgi erişim performansını artırmak.
- Türkçe metinler üzerinde çalışan bilgi erişim sistemlerinin, dilin morfolojik ve sentaktik özellikleri karşısında nasıl bir performans gösterdiğini analiz etmek.

- Elde edilen sonuçları hem sayısal hem de görsel olarak sunarak, yöntemler arasındaki farkları anlamlı bir şekilde değerlendirmek.

## II. LİRETATÜR TARAMASI

Bilgi erişimi ve doğal dil işleme (NLP) alanında yapılan araştırmalar, özellikle son yıllarda yoğun bir şekilde gelişmiştir. Bilgi erişimi sistemleri, kullanıcıların doğal dilde sordukları sorulara doğru cevaplar verebilmek amacıyla, hem klasik yöntemlere hem de derin öğrenme tabanlı modern yöntemlere dayanmaktadır. Bu bölümde, çalışmamızla ilgili önemli literatür çalışmaları incelenmiş ve mevcut yöntemlerin avantajları ile sınırlılıkları ele alınmıştır.

### A. Klasik Bilgi Erişim Yöntemleri

Klasik bilgi erişim yöntemleri, sorgu ve dokümanlar arasındaki kelime düzeyinde benzerlikleri ölçmeye odaklanmıştır. BM25 (Best Matching 25) algoritması, bu alandaki en yaygın kullanılan yöntemlerden biridir ve ters frekans ağırlığı (TF-IDF) ile sorgu-doküman benzerliğini hesaplar. BM25 doküman sıralama için etkili bir modeldir ve çeşitli metin tabanlı uygulamalarda yaygın olarak kullanılır [2].

### B. Kelime Temelli Eşleştirme Yöntemleri

Kelime temelli eşleştirme yöntemleri, sorgu ve dokümanlar arasında benzerlik ölçümleri yapmak için Jaccard benzerlik metriği gibi basit metriklerden faydalanır. Bu yöntemler, sorgu ve dokümanları kelime kümelerine dönüştürerek, ortak kelimelerin oranını hesaplar. Basitliği nedeniyle hızlıdır; ancak bağlam anlayışında yetersiz kalır.

### C. Transformer Tabanlı Yöntemler

Derin öğrenme ve Transformer modellerinin yükselişi, bilgi erişim sistemlerini önemli ölçüde dönüştürmüştür. BERT ve türevleri, sorgu ve dokümanların bağlamını dikkate alarak anlam temelli eşleştirme yapabilmektedir.

### D. Hibrit Yöntemler

Hibrit yöntemler, farklı algoritmaların güçlü yönlerini birleştirerek performansı artırmayı amaçlar. BM25 gibi klasik yöntemler, Transformer tabanlı modellerin yoğun vektör temsilleriyle birleştirilerek doğruluk artırılabilir.

## III. METODOLOJİ

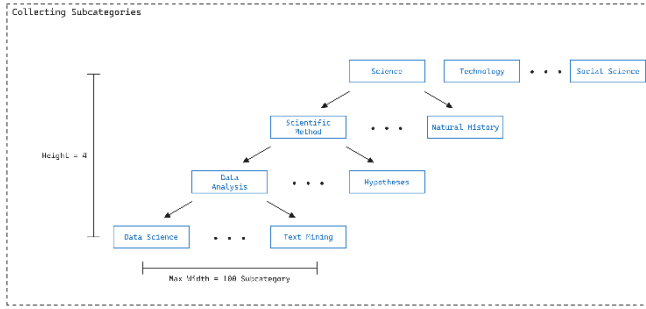
Metodoloji, veri setinin hazırlanması, bilgi alma yöntemlerinin uygulanması ve bu yöntemlerin doğruluk ölçütlerine göre değerlendirilmesi olmak üzere üç temel aşamadan oluşmaktadır.

## A. Veri Seti

Çalışmada, Hugging Face platformundan temin edilen WikiRAG-TR veri seti kullanılmıştır. Bu veri seti, Wikipedia Türkçe Maddelerinin giriş kısmından sentetik olarak oluşturulan 5999 soru ve cevap çiftinden oluşan bir veri kümesidir. Veri kümesi, Türkçe Arama-Artırılmış Üretim (RAG) görevleri için kullanılmak üzere oluşturulmuştur [1]. Veri seti, Türkçe dilindeki bilgi alma süreçlerini test etmek ve analiz etmek için uygun bir yapı sunmaktadır.

Veri setinin oluşturulması iki ana aşamada gerçekleştirildi ve her aşama ayrı bir diyagramla gösterilmiştir [1].

### Aşama 1: Alt Kategori Koleksiyonu



Şekil. 1 Alt Kategori Koleksiyonu [1].

Bu ilk aşamada:

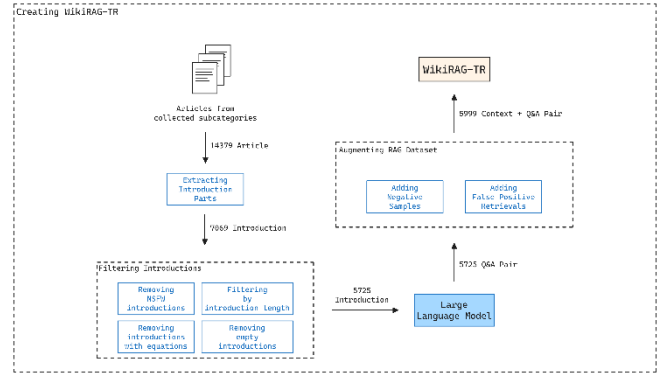
- Bilim, teknoloji, mühendislik, matematik, fizik, kimya, biyoloji, jeoloji, meteoroloji, tarih, sosyal bilimler ve daha fazlasını içeren özel bir tohum kategorisi listesi belirlendi.
- Bu tohum kategorileri kullanılarak, alt kategoriler Wikipedia'dan tekrarlı olarak toplandı. Tekrarlama derinliği 3 olarak ayarlandı ve toplanacak alt kategori sayısı her derinlik katmanı için 100 ile sınırlandırıldı.
- Her adım için aşağıdaki alt kategori türleri filtreledi:  
NSFW kelimeleri içeren alt kategoriler .  
Yalnızca öge listeleri içeren alt kategoriler  
Şablon olarak kullanılan alt kategoriler
- Ortaya çıkan alt kategori listesinden makaleler edinildi[1].

Veri Kümesi Sütunları,

- id: Her satır için benzersiz tanımlayıcı.
- question: Modelin ürettiği soru.
- answer: Modelin ürettiği cevap.
- context: Hem ilgili hem de ilgisiz bilgileri içeren artırılmış bağlam.
- is\_negative\_response: Cevabın olumsuz bir yanıt olup olmadığını belirtir (0: Hayır, 1: Evet).
- number\_of\_articles: Bağlamı oluşturmak için kullanılan makale girişlerinin sayısı.
- ctx\_split\_points: Bağlamdaki her girişin bitiş karakter dizinleri. Bunlar, sütunu contextkendi bireysel makale girişlerine bölmek için kullanılabilir.

- correct\_intro\_idx: Bağlamdaki ilgili girişin dizini. İlgili girişi bulmak için birlikte kullanılabilir ctx\_split\_points. Bu, eğitim sonrası analiz için de yararlı olabilir [1].

### Aşama 2: Alt Kategori Koleksiyonu



Şekil. 2 Veri Seti Oluşturma [1].

İkinci aşama:

- Giriş bölümleri 1. Aşamada toplanan makalelerden çıkarılmıştır.
- Giriş bölümü çok kısa ya da çok uzun (50 karakterden az, 2500 karakterden fazla) ise makale silindi.
- Girişte NSFW sözcükleri yer alıyorsa makale atıldı.
- Girişte denklemler yer alıyorsa makale atılıyordu.
- Giriş kısmı boş bırakılmışsa makale atılmıştır.
- (Gemma-2-27B-it)Filtrelenen girişler , sentetik soru ve cevap çiftleri oluşturmak için büyük bir dil modeline beslendi .
- Veri setinde ortaya çıkan her satır için (giriş, soru ve cevap içeren) aşağıdaki işlemler gerçekleştirildi:

Bağlama yanlış pozitif geri dönüşler eklemek için diğer satırlardan ilgisiz bağlamlar (girişler) toplandı.

Birbirleriyle ilgisi olmayan bu bağlamlar bir listeye eklendi.

İlgili bağlam bu listeye eklendi. (Bazı durumlarda, cevabın modelin yetersiz bilgi nedeniyle soruyu cevaplayamayacağını gösterdiği olumsuz örnekler oluşturmak için ilgili bağlam atlandı . Bu olumsuz örnekler ayrı ayrı oluşturuldu ve tüm orijinal soruların karşılık gelen cevaplara sahip olması sağlandı.)

İlgili bağlamın konumunu rastgele belirlemek için liste karıştırıldı.

Liste elemanları '\n' karakteri kullanılarak birleştirildi.

Veri setinden rastgele 1000 soru örneklenmiş ve her bir soru için bağlam oluşturulmuştur. Bağlamlar, sorularla ilişkilendirilmiş farklı metin parçalarını (chunks) içermektedir. Her bir soru için 5 metin parçası (chunk) oluşturulmuş ve doğru chunk'lar veri setinde belirtilmiştir. Chunk'lar, bağlamı temsil eden küçük metin bölümleridir ve bu bölümler üzerinde bilgi alma yöntemleri test edilmiştir.

### B. Model Yöntemleri

Bu çalışmada, hem geleneksel hem de derin öğrenme tabanlı bilgi alma yöntemleri değerlendirilmiştir.

#### BM25 (Bag-of-Words Tabanlı)

BM25, bir "bag-of-words" tabanlı yöntem olup, sorgu ve doküman arasındaki terim frekansına dayalı bir skor hesaplar. Bu yöntem, sorgunun doküman içindeki terimlerle eşleşme durumuna ve terimlerin ağırlıklarına göre dokümanların sıralanmasını sağlar. BM25, kelime sırasını dikkate almamasına rağmen bilgi alma sistemlerinde geniş çapta kullanılan temel bir yöntemdir.

#### Kelime Benzerliği (Word Matching)

Kelime benzerliği, sorgu ve dokümanlar arasında kelime tabanlı benzerlik ölçümleri yapar. Bu çalışmada Jaccard Benzerliği metriği kullanılmıştır. Jaccard benzerliği, iki metin arasındaki ortak kelimelerin oranını hesaplayarak dokümanlar arasındaki benzerliği değerlendirir.

#### Dense Vektör Temsilleri

Dense vektör temsilleri, metinlerin bağlamlarını daha iyi anlamak için derin öğrenme modelleri tarafından oluşturulan vektörlerdir. Bu çalışmada aşağıdaki modeller kullanılmıştır:

- sentence-transformers/all-MiniLM-L12-v2: Transformer tabanlı bir model olup, metinler arasında bağlam tabanlı benzerlik ölçümü yapar.
- intfloat/multilingual-e5-large-instruct: Çok dilli bir model olup, farklı dillerde bağlamları eşleştirmede başarılıdır.
- ytu-ce-cosmos/turkish-colbert: Türkçe için optimize edilmiş bir modeldir ve bağlamları daha yüksek doğrulukla eşleştirir.
- thenlper/gte-large ve BAAI/bge-m3: Daha büyük dil modelleri olup, metin temsillerinde daha derin bir anlam çıkarımı sağlar.

#### Birleşik Yöntem (Weighted Combination)

Bilgi alma yöntemlerinden elde edilen skorlar, doğrusal kombinasyon (weighted sum) yöntemi ile birleştirilmiştir. Her yöntemin çıktı skorlarına farklı ağırlıklar atanarak, bu yöntemlerin ortak performansları analiz edilmiştir. Kombinasyonda kullanılan ağırlıklar şu şekilde belirlenmiştir:

BM25: 0.1

Jaccard Word Matching: 0.1

Dense Vektör Modelleri: 0.1, 0.3, 0.1, 0.2, 0.1

Birleşik yöntem, tüm yöntemlerden elde edilen skorların normalize edilerek ağırlıklı toplamalarının alınmasıyla oluşturulmuştur.

### C. Performans Ölçütleri

Bilgi alma yöntemlerinin başarısı, aşağıdaki iki ölçüte göre değerlendirilmiştir:

- Top-1 Doğruluk: Doğru chunk'ın, sorgu sonuçlarında ilk sırada yer alma oranıdır.
- Top-5 Doğruluk: Doğru chunk'ın, sorgu sonuçlarında ilk 5 sırada yer alma oranıdır.

Her yöntem için doğruluk oranları hesaplanmış ve karşılaştırmalı analiz yapılmıştır.

### D. Uygulama Süreci

Dense vektör temsilleri, ilgili modeller kullanılarak üretilmiş ve FAISS kütüphanesi kullanılarak bir vektör veritabanına kaydedilmiştir.

Sorgular, hem geleneksel hem de derin öğrenme tabanlı yöntemler ile işlenmiştir.

Elde edilen skorlar birleştirilmiş ve performans ölçütlerine göre değerlendirilmiştir.

## IV. DENEYLER VE SONUÇLAR

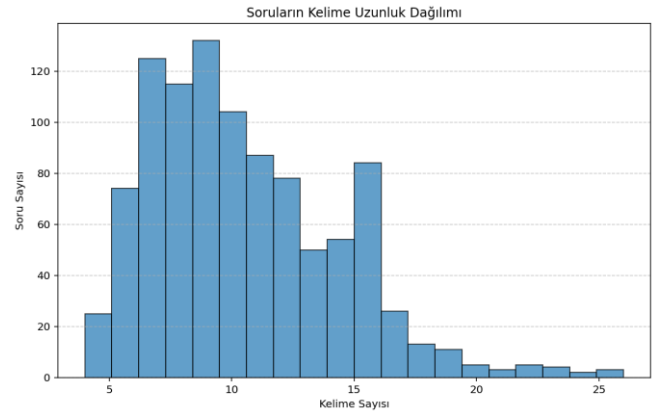
Çeşitli bilgi alma yöntemlerinin performansı Türkçe dilindeki WikiRAG-TR veri kümesi üzerinde değerlendirilmiştir. Deneylerde kullanılan yöntemler şunlardır:

- BM25.
- Word Matching.
- Dense Representations:
  - sentence-transformers/all-MiniLM-L12-v2
  - intfloat/multilingual-e5-large-instruct
  - ytu-ce-cosmos/turkish-colbert
  - thenlper/gte-large
  - BAAI/bge-m3
- Weighted Combination:

Deneylerde:

- Veri kümesinden rastgele 1000 soru seçildi.
- Her soru için 5 bağlam (chunk) oluşturuldu.
- Modeller, hem Top-1 hem de Top-5 doğruluk metriklerine göre değerlendirildi.
- FAISS tabanlı indeksler dense vektörler için kullanıldı.

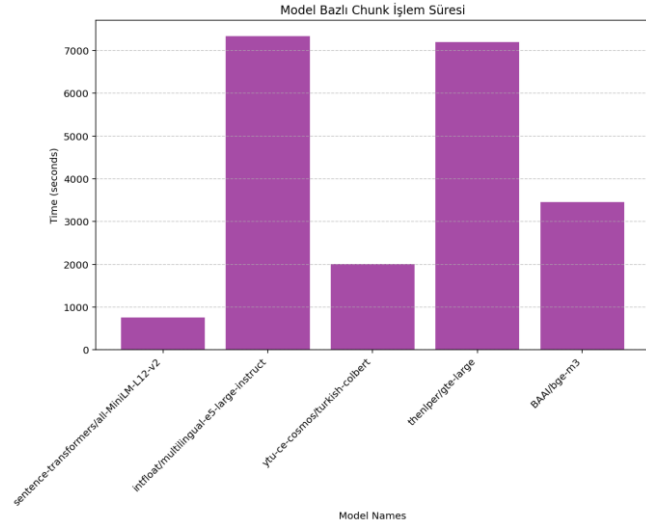
### A. Rastgele seçilen 1000 sorunun Kelime Uzunluk Dağılımı:



Şekil. 3 Soruların kelime uzunluk dağılımı.

### B. Model bazlı olarak Chunkların İşlem Süresi

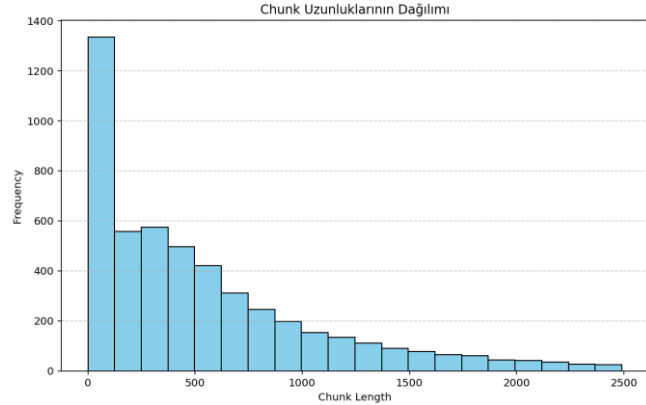
Modellerin bağlamlar üzerinde işlem süreleri görselleştirilmiştir. Dense modellerin işlem süreleri arasında belirgin farklılıklar gözlenmiştir.



Şekil. 4 Model bazlı chunk işlem süresi.

### C. Chunk Uzunluklarının Dağılımı

Soru bağlamlarının uzunlukları histogram ile görselleştirilmiştir. Verinin geneli kısa bağlamlardan oluşmaktadır.



Şekil. 5 Chunk uzunluklarının dağılımı.

### D. Performans Sonuçları

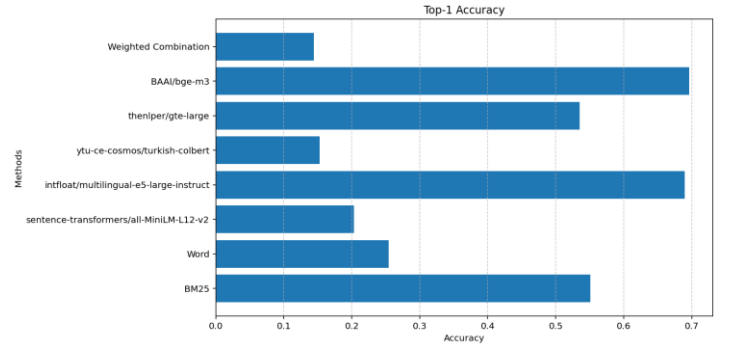
Sonuçlar aşağıdaki tablo ve grafiklerde özetlenmiştir.

TABLO 1. PERFORMANS SONUÇLARI

Yöntem	Top-1 Accuracy	Top-5 Accuracy
BM25	0.551	0.785
Word Matching	0.254	0.440
sentence-transformers/all-MiniLM-L12-v2	0.203	0.321
intfloat/multilingual-e5-large-instruct	0.690	0.912
ytu-ce-cosmos/turkish-colbert	0.153	0.300
thenlper/gte-large	0.535	0.776
BAAI/bge-m3	0.696	0.915
Weighted Combination	0.144	0.198

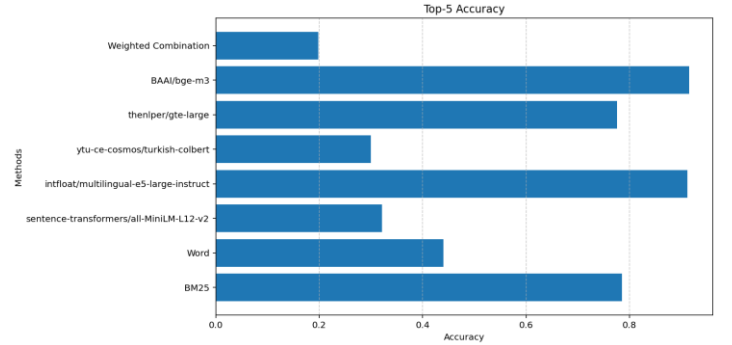
- intfloat/multilingual-e5-large-instruct ve BAAI/bge-m3 modelleri en yüksek Top-1 ve Top-5 doğruluk değerlerine ulaşmıştır.
- BM25, Word Matching yöntemine kıyasla daha yüksek bir performans göstermiştir.
- Dense vektör tabanlı yöntemlerden bazıları beklenenden düşük performans sergilemiştir, bu durum veri yapısı ve dil modeli optimizasyonlarıyla ilişkilendirilebilir.
- Weighted Combination yöntemi, beklenenin aksine düşük performans göstermiştir, bu da ağırlıklandırma optimizasyonunun yeniden gözden geçirilmesi gerektiğini göstermektedir.

### Top-1 Accuracy Karşılaştırması



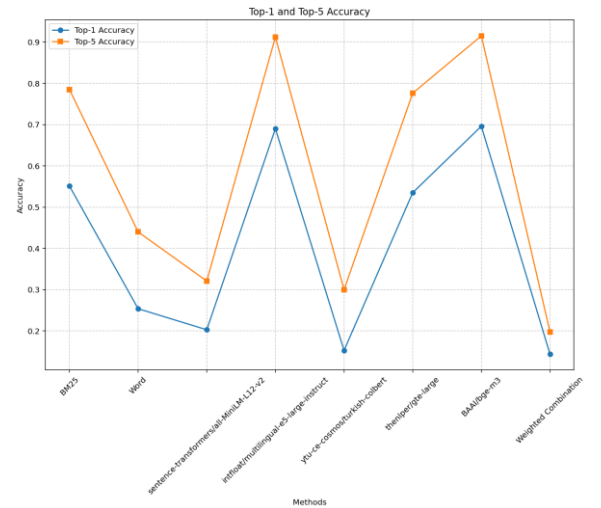
Şekil. 6 Top-1 accuracy.

### Top-5 Accuracy Karşılaştırması



Şekil. 7 Top-5 accuracy.

### Top-1 ve Top-5 Accuracy Karşılaştırması

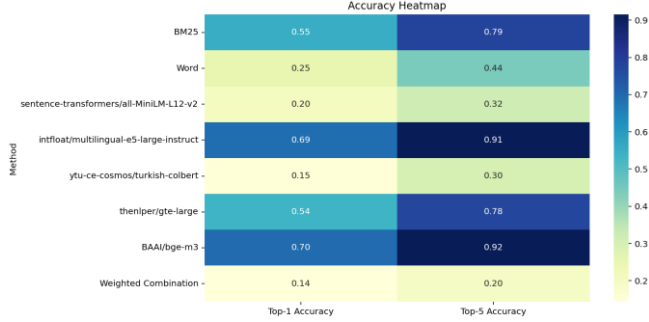


Şekil. 8 Top-1 vs Top-5 accuracy.

En dikkat çekici fark, dense modellerin Top-5 doğruluk oranlarının daha belirgin bir şekilde yüksek olmasıdır.

#### Accuracy Heatmap

Performans sonuçlarının görselleştirilmesi amacıyla oluşturulmuştur. En yüksek doğruluk değerleri intfloat/multilingual-e5-large-instruct ve BAAI/bge-m3 yöntemlerinde gözlemlenmiştir.



Şekil. 9 Model bazlı chunk işlem süresi

Deneyler, farklı bilgi alma yöntemlerinin performanslarını karşılaştırmak için etkili bir temel oluşturmuştur:

BM25, hızlı ve etkili bir temel yöntem olarak ön plana çıkmıştır.

Dense modeller, özellikle intfloat/multilingual-e5-large-instruct ve BAAI/bge-m3, yüksek doğruluk sonuçlarıyla dikkat çekmiştir.

Weighted Combination yönteminin ağırlıklarının iyileştirilmesi gerektiği açıkça görülmüştür.

Bu sonuçlar, Türkçe dilinde bilgi alma görevleri için yöntemlerin avantaj ve dezavantajlarını değerlendirmek adına önemli bir katkı sağlamaktadır.

#### V. SONUÇLAR

Bu çalışmada, Türkçe dilinde bilgi alma (retrieval) yöntemlerinin performansı, WikiRAG-TR veri kümesi kullanılarak değerlendirilmiştir. Çalışma, kelime tabanlı yöntemler (BM25, Word Matching), dense vektör tabanlı yöntemler ve bu yöntemlerin ağırlıklı bir kombinasyonu ile gerçekleştirilmiştir. Elde edilen sonuçlar, çeşitli yöntemlerin güçlü ve zayıf yönlerini ortaya koymuştur.

- **BM25:** Kelime tabanlı bir yöntem olarak hızlı ve etkili bir performans göstermiştir, ancak dense yöntemler kadar ayrıntılı anlam yakalayamamıştır.
- **Word Matching:** Jaccard Similarity temelli bu yöntem, BM25'e kıyasla daha düşük bir doğruluk oranına sahiptir.
- **Dense Modeller:**  
intfloat/multilingual-e5-large-instruct ve BAAI/bge-m3, yüksek Top-1 ve Top-5 doğruluk değerleriyle ön plana çıkmıştır.  
Diğer dense modellerin performansları değişiklik göstermiştir ve bazı modeller beklenenden düşük sonuçlar vermiştir.
- **Weighted Combination:** Farklı yöntemlerin ağırlıklı bir kombinasyonu düşük performans göstermiş, bu

da ağırlıklandırmanın iyileştirilmesi gerektiğini ortaya koymuştur.

#### Çalışmanın Katkıları

Çalışma, Türkçe dilinde bilgi alma yöntemlerinin karşılaştırılması için önemli bir katkı sağlamış ve literatürdeki boşluğu doldurmuştur. Farklı bilgi alma yöntemlerinin güçlü ve zayıf yönleri detaylı olarak analiz edilmiştir. Soru uzunlukları, chunk uzunlukları ve model işlem süreleri gibi detaylı analizlerle veri daha anlaşılır hale getirilmiştir.

#### Çalışmanın Sınırlamaları

- Kullanılan veri kümesi belirli bir alanla sınırlıdır (Wikipedia tabanlı).
- Dense modellerin işlem süreleri, gerçek zamanlı uygulamalar için optimize edilmemiştir.
- Weighted Combination yöntemi beklenen sonuçları verememiştir.

#### VI. GELECEK ÇALIŞMALAR

Bu çalışmada, Türkçe dilinde bilgi alma yöntemlerinin performansları detaylı bir şekilde incelenmiştir. Çalışma, mevcut yöntemlerin güçlü ve zayıf yönlerini anlamamıza olanak sağlamış olsa da, gelecekte yapılabilecek birçok geliştirme ve genişletme imkanı bulunmaktadır:

- **Veri Seti ve Çeşitlilik**

Kullanılan WikiRAG-TR veri kümesi, Türkçe dilinde bilgi alma yöntemlerini değerlendirmek için uygun bir başlangıçtır. Gelecekte, daha büyük ve daha çeşitli veri setleriyle çalışarak modellerin genel performansı test edilebilir.

Medikal, finansal veya hukuki gibi alanlarda özelleşmiş veri setleriyle deneyler yapılarak, yöntemlerin belirli alanlarda ne kadar etkili olduğu incelenebilir.

- **Model Optimizasyonu**

Weighted Combination yöntemindeki düşük performans, ağırlıkların daha etkili bir şekilde optimize edilmesi gerektiğini göstermektedir. Genetik algoritmalar veya grid search gibi hiperparametre optimizasyon teknikleri kullanılabilir.

Daha gelişmiş ve modern dil modellerinin (örneğin, GPT-4 tabanlı dense modeller) eklenmesiyle performans karşılaştırmaları genişletilebilir.

- **Zaman ve Kaynak Optimizasyonu**

**Verimli Model İşleme:** Dense modellerin uzun işlem süreleri, gerçek zamanlı bilgi alma sistemleri için bir sınırlamadır. Daha hızlı sonuçlar elde edebilmek için model sıkıştırma (pruning) veya daha hafif modellerin kullanımı araştırılabilir.

**Parallelleştirme ve Donanım Kullanımı:** GPU ve TPU gibi donanımlardan daha iyi yararlanarak işlem süreleri azaltılabilir.

- **Hibrid Yöntemlerin Geliştirilmesi**

Ağırlıklı Hibrid Yöntemler: BM25, Word Matching ve Dense yöntemlerini daha etkili bir şekilde birleştirmek için makine öğrenimi tabanlı modeller (örneğin, bir Ensemble öğrenme yöntemi) kullanılabilir.

Derin Öğrenme Tabanlı Hibrid Modeller: Hibrid yöntemlerde derin öğrenme tabanlı modellerin karar mekanizması olarak kullanılması incelenebilir.

- Performans Değerlendirme

Detaylı Hata Analizi: Modellerin düşük performans gösterdiği örneklerin incelenmesi, sistemin geliştirilmesi için yeni fikirler sağlayabilir.

Kullanıcı Çalışmaları: Bilgi alma yöntemlerinin kullanıcı memnuniyetine etkisi araştırılabilir.

- Çok Dilli Performans Karşılaştırması

Türkçe dilindeki yöntemlerin performansı başarılı bir şekilde ölçülmüştür. Gelecekte, aynı deneyler farklı dillerde tekrarlanarak yöntemlerin çok dilli ortamlarda nasıl performans gösterdiği değerlendirilebilir.

Çalışmanın kod linkine [3]. ulaşabilirsiniz.

#### KAYNAKÇA

- [1] WikiRAG-TR, <https://huggingface.co/datasets/Metin/WikiRAG-TR>, E.T. Ocak 2025.
- [2] S. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, 2009, pp 333-389.
- [3] Çalışmanın Kod linki, [https://github.com/raifakyol/LLM\\_WikiRAG-TR/blob/main/kollektif.py](https://github.com/raifakyol/LLM_WikiRAG-TR/blob/main/kollektif.py), E.T. Ocak 2025.
- [4] Çalışmanın deposu, [https://github.com/raifakyol/LLM\\_WikiRAG-TR](https://github.com/raifakyol/LLM_WikiRAG-TR), E.T. Ocak 2025.